Ontology Semantic Disambiguation by LLM

Anastasiia Riabova[1], Rémy Kessler[1][a] and Nicolas Béchet[2][b]
[1]*Univ. d'Avignon, LIA, 339 chemin des Meinajariès, 84911 Avignon, France*
[2]*Univ. Bretagne Sud, CNRS, IRISA, Rue Yves Mainguy, 56000 Vannes, France*

Keywords: LLM, CamemBERT, Ontology, Zero/Few-Shot, CoT, Self-Consistency, Prompt Engineering.

Abstract: Within the BPP project, a combination of statistics and word n-gram extraction enabled the creation of a bilingual (French/English) ontology in the field of e-recruitment. The produced dataset was of good quality, but it still contained errors. In this paper, we present an approach that explores the use of large language models (LLMs) to automate the validation and enrichment of ontologies and knowledge graphs. Starting with a naive prompt and using small language models (SLMs), we tested various approaches, including zero-shot, few-shot, chain-of-thought (CoT) reasoning, and self-consistency (SC) decoding. The preliminary results are encouraging, demonstrating the ability of LLMs to make complex distinctions and to evaluate the relationships derived from our ontology finely.

1 INTRODUCTION

Integrating knowledge from various sources into ontologies and knowledge graphs remains a complex problem. The Web is now a major resource, with vast and diverse information, digital encyclopedias, forums, blogs, public websites, "social tagging," and networks, enabling the generation of ontologies and knowledge graphs. Yet, the exponential growth of these bases makes manual verification and validation increasingly time-consuming.

Within the BPP project (Butterfly Predictive Project), statistics and n-gram extraction supported the creation of an ontology for e-recruitment in English and French, covering 440 job titles across 27 sectors, accessible here [1]. Part of it was manually evaluated with good results (0.8 precision), but the large number of candidate terms prevented full validation.

The rise of Large Language Models (LLMs) opens new perspectives for ontology enrichment. They can detect errors, inconsistencies, ill-defined concepts, and missing relations Petroni et al. (2019). Leveraging their analysis and generation abilities makes creating consistent and user-friendly knowledge bases more feasible.

[a] https://orcid.org/0000-0002-9947-3048
[b] https://orcid.org/0000-0001-9425-5570

[1]https://www-labs.iro.umontreal.ca/~lapalme/LBJ/BPPontologie/

This study presents initial results to aid semantic disambiguation of concepts and relations within this noisy ontology. After related work (Section 2), we present the data (Section 3), then detail our approach (Section 4) and first results (Section 5), before concluding (Section 6).

2 RELATED WORKS

Recent advances in natural language processing (NLP) have improved the modeling of semantic relationships in ontologies. Resources like ESCO or ROME provide structured information, but keeping them up to date requires significant manual work. To address this, deep learning approaches based on transformers have emerged Vaswani et al. (2017), either encoder-only (e.g. BERT Devlin et al. (2018)) or decoder-only (e.g. GPT Brown and al. (2020)). Encoders project entities into a vector space to estimate semantic proximity, while LLMs enable explicit reasoning methods such as zero-shot, few-shot, or chain-of-thought prompting (CoT). In this article, we compare these methods for semantic matching on noisy recruitment ontology data.

LLM performance has grown with model scaling, but larger models face limits in energy consumption and deployment complexity. Alternative methods—few-shot prompting Brown and al. (2020), CoT Wei et al. (2022); Kojima et al. (2022),

Riabova, A., Kessler, R. and Béchet, N.
Ontology Semantic Disambiguation by LLM.
DOI: 10.5220/0013834800004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 2: KEOD and KMIS*, pages 175-183
ISBN: 978-989-758-769-6; ISSN: 2184-3228

instruction-tuned CoT Ranaldi and Freitas (2024), or self-consistency Wang et al. (2022); Chen et al. (2023) improve results without increasing size. Given the time-intensive nature of ontology construction, adopting LLMs appears both expected and justified.

Several works explore this direction. Meyer et al. Meyer et al. (2024) tested ChatGPT for query generation and knowledge extraction. Kommineni et al. Kommineni et al. (2024) combined ChatGPT for competency questions with Mixtral 8x7B for entity extraction, building a knowledge graph via a RAG-based workflow. Abolhasani and Pan Abolhasani and Pan (2024) developed OntoKGen, which uses an iterative CoT algorithm with user validation for automated graph generation in Neo4j.

Unlike these works, which focus on building ontologies from scratch, we address an already populated ontology, emphasizing LLM-based validation of entities and relationships.

3 DATA AND STATISTICS

In this section, we present the data from the BPP project. A bilingual (English/French) ontology of 440 occupations from 27 activity domains was developed for the e-recruitment sector.

Each occupation is linked to the necessary skills for its practice, totaling approximately 6,000 different skills. This data is organized according to the ESCO modeling le Vrang et al. (2014)[2], a multilingual European classification project for skills, occupations, and qualifications, aiming to create European harmonization in recruitment. Table 1 presents some descriptive statistics for this ontology.

Table 1: Descriptive statistics for the ontology.

	Ontology	
	in French	in English
Unigrams	9,335	3,810
Bigrams	5,995	3,785
Trigrams	305	2,421
Unique skills	2,962	4,015
No. of occupations	312	127

Table 2 presents an example of evaluated n-grams for the occupation of 'Analyste financier' (Financial analyst), categorized by transversal skills[3] and technical skills[4]. Each occupation is thus linked to a set of word n-grams (from 1 to 3) ranked by TF-IDF.

4 METHODOLOGY

This section is divided into three subsections, corresponding to the three phases of our experiments. First, we present the methodology used for fine-tuning a BERT-based model. Second, we will describe the methodology behind our experiment involving prompt engineering. And thirdly, we present our final pipeline, which yielded the best results.

Fine-Tuning an Encoder-Only Model. This subsection presents experiments conducted with encoder-only models, whose objective was to measure the semantic similarity between occupation and skill descriptions. These models represent each element (occupation or skill) as a vector in a latent space and estimate their proximity using a measure such as the cosine similarity. To train an encoder model to classify occupation-skill pairs as relevant or not, we formulated the problem as a binary classification task. Positive examples were extracted directly from the ESCO ontology. To generate negative examples, we compared three techniques: negative random sampling, easy negative mining and hard negative mining. Figure 1 illustrates the three methods.

Random Negative Sampling. For each occupation, we randomly selected unrelated skills in ESCO, considering them as irrelevant. This method assumes that relationships absent in the ontology correspond to absences of semantic link, which can introduce latent noise if some skills, relevant, are simply not listed.

Easy Negative Mining. A more controlled variant of random negative sampling consists of selecting, among the skills unrelated to a given occupation, those that are most semantically distant in the vector space. This method, known as easy negative mining, enables the creation of high-quality negative examples while minimizing the risk of false negatives.

To this end, we used vector representations derived from the [CLS] token of CamemBERT Martin et al. (2020), which serves as a global summary of the

[2]European Skills Competences and Occupations https://ec.europa.eu/esco/

[3]Transversal skills, also known as soft skills, are personal and social skills, oriented towards human interactions, which can be considered relevant regardless of the occupation.

[4]Technical skills, also known as hard skills, are formally demonstrable skills resulting from technical learning, often academic, and evidenced by grades, diplomas, or certificates.

Table 2: List of skills, classified by unigrams, bigrams, and trigrams, for the occupation 'financial analyst' obtained from 497 job offers. N-grams considered irrelevant have been struck through. Skills are sorted in descending order of score.

Financial analyst	
Soft skills	Hard skills
financial, business, support, management, process, reports, data, project, ~~including~~, projects ...	accounting, analysis, finance, reporting, cpa, cma, budget, cga, ~~end~~, forecast ...
analytical skills, communication skills, problem solving, ability work, internal external, ~~real estate~~, decision making, interpersonal skills, financial services, verbal written ...	~~financial analyst~~, financial analysis, financial reporting, financial statements, variance analysis, finance accounting, accounting finance, balance sheet, journal entries, financial modelling ...
analytical problem solving, problem solving skills, verbal written communication, ability work independently, key performance indicators, fast paced environment, oral written communication, time management skills, communication interpersonal skills, interpersonal communication skills ...	financial planning analysis, ad hoc reporting, financial reporting analysis, financial analysis reporting, financial statement preparation, year end close, consolidated financial statements, planning budgeting forecasting, business case analysis, possess strong analytical ...

sequence. Each occupation and skill was individually encoded and represented by the [CLS] token vector extracted from the model's final layer. We then computed the cosine distance between each occupation's [CLS] vector and those of all unrelated skills in the ESCO ontology. For each occupation, the most distant skills were selected as easy negatives and labeled as 0 in our classification dataset.

Hard Negative Mining. To complement the previous methods, we explored a hard negative mining strategy, inspired by contrastive learning Robinson et al. (2021), to generate more difficult negative examples, i.e. skills that are semantically close to the occupations but not actually related. The objective is to encourage the model to learn finer distinctions between truly relevant and ambiguous cases.

We used the positive-aware hard negative mining proposed in de Souza P. Moreira et al. (2025), in particular Top-k with percentage to positive threshold (TopKPercPos). This method helps to reduce the number of potential false negatives, which represent a fairly common problem when performing hard negative mining, taking advantage of information from the positive relevance score (percentage in this case).

Unlike the previous method, which relied on CamemBERT for embeddings, here we employed the intfloat/multilingual-e5-large model, a multilingual encoder trained with contrastive learning and compatible with the Sentence Transformers library Reimers and Gurevych (2019). This choice ensured consistency with the family of models used in our reference article.

For each occupation, we proceeded as follows:

- We computed the embeddings of all skills and occupations using the E5 model.
- We identified the highest cosine similarity score among the positive skills for the given profession.
- We then selected, among the unrelated skills, those whose score was less than 95% of this positive score, as hard negatives.

However, with a threshold set at 95% of the positive score, we observed that many relevant, yet not explicitly related, skills were falsely considered negative. We therefore lowered the threshold to 90% of the positive score, which preserved the difficulty of the negative examples while reducing the occurrence of false negatives.

4.1 Prompt Engineering

In this subsection, we present the LLMs used for evaluating occupation–skill relationships. Unlike encoder models, which produce vector representations for static pairs, LLMs are used here in a generative setting, responding to carefully designed prompts. This setup enables us to leverage their ability to follow instructions, reason over input, generalize, and generate explanations.

We used several models, belonging to different families and sizes: open-source models that can be deployed locally (Mistral, Gemma, DeepSeek, Qwen, Phi) via the Ollama tool, as well as proprietary models accessible via a web interface and API (GPT, Le Chat (Mistral)).

They were tested with different prompt configurations: zero-shot, few-shot, chain-of-thought, and self-consistency. It will be discussed in more detail in the following subsections.

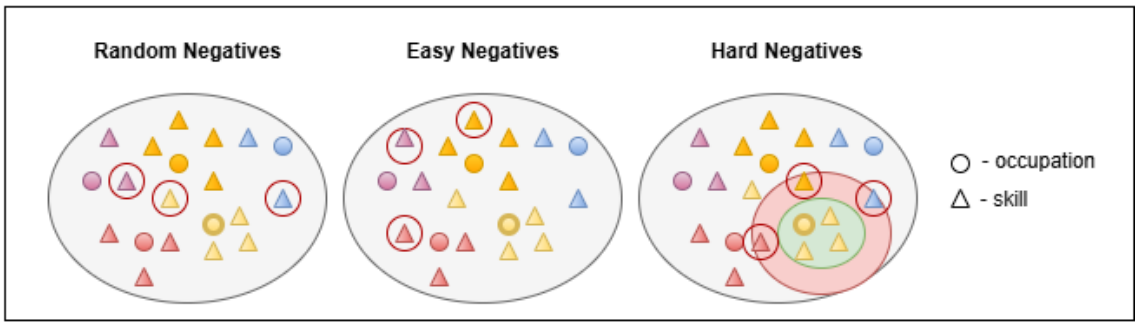

Figure 1: Schematic representation of the negative sampling methods in the embedding space for a given occupation (yellow circle).

4.1.1 Zero-Shot and Few-Shot Prompting

We began in a zero-shot setting, i.e., without providing examples or detailed instructions. This approach is not only computationally lightweight but also allows us to directly evaluate the model's implicit knowledge and potential biases.

Few-shot prompting involves presenting the model with a small number of annotated examples directly within the prompt, in order to guide its behavior without additional training. As demonstrated by Brown et al. Brown and al. (2020) with GPT-3, this approach often achieves results comparable to fine-tuning, while avoiding the costs associated with creating annotated datasets.

In our study, few-shot prompting was used as an intermediate step between zero-shot prompting and more advanced techniques such as chain-of-thought prompting. The objective was twofold: to assess the potential improvement over the zero-shot setting; to observe the effects of example formatting and distribution on model responses.

Few-shot prompting was subsequently reused in a more elaborate form in the context of chain-of-thought prompting.

4.1.2 Chain-of-Thought Prompting

Conventional zero-shot and few-shot prompting approaches show their limitations when a task requires explicit reasoning. To address this shortcoming, Wei et al. Wei et al. (2022) introduced Chain-of-Thought (CoT) prompting, which involves enriching the prompt with a natural language reasoning chain that explicitly exposes the logical steps leading to the answer. This method has achieved significant improvements on several complex reasoning tasks, but primarily with very large models (over 100 billion parameters).

By contrast, small models (Small Language Models, SLMs) tend to produce superficially coherent but logically incorrect reasoning Gudibande et al. (2023), often leading to worse performance than conventional prompting. To address this limitation without resorting to resource-intensive techniques such as fine-tuning or knowledge distillation, we adopted a strategy of directly injecting pre-constructed reasoning chains into the few-shot prompt.

Three variants of CoT prompting were tested:

1. Zero-shot Chain-of-Thought (zero-CoT), based on the approach of Kojima et al. Kojima et al. (2022), simply adds the phrase "Let's think step by step" to the original prompt.
2. CoT generated by Mistral Le Chat (in-family): We asked the model to generate complete justifications for occupation–skill pairs, which were then inserted into a few-shot prompt. The goal was to test the hypothesis proposed by Ranaldi and Freitas (2024), namely that a student model benefits more from exposure to reasoning generated by a model from the same family (in-family alignment).
3. ChatGPT-generated CoT (out-of-family): This variant followed the same approach as above, but used ChatGPT-4 as the generator. GPT-4 acted here as an out-of-family teacher model, following a logic inspired by knowledge distillation without training, but by injecting curated reasoning examples. All generated outputs were manually reviewed by an expert annotator.

4.1.3 Self-Consistency Decoding

The self-consistency (SC) method, introduced by Wang et al. Wang et al. (2022), aims to improve the robustness of CoT prompting. Instead of relying on a single greedy response, this approach samples multiple reasoning chains via stochastic decoding and determines the final answer by majority voting. The underlying intuition is that for complex tasks, correct reasoning—though diverse—tends to converge on the same conclusion more often than incorrect reasoning.

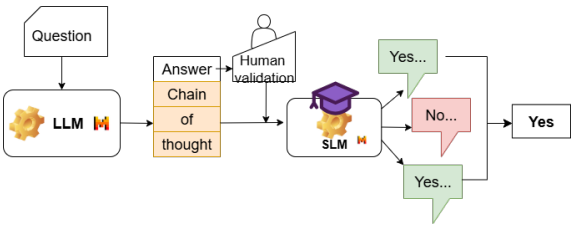

Figure 2: Self-consistency using a CoT prompt generated by the teacher LLM and validated by a human.

In this study, we did not use the classic probabilistic decoding mechanism, but instead adapted the self-consistency concept for our context. More specifically, we implemented two variants:

Majority Voting. For each occupation–skill pair, we generated nine independent responses from the same model using an identical few-shot CoT prompt. The final prediction (yes or no) was determined by a majority vote across the nine responses.

Universal Self-Consistency (USC). Inspired by Chen et al. Chen et al. (2023), this variant consists of submitting the nine generated responses to a follow-up prompt in which the same model is asked to select the most consistent answer, according to its judgment.

Figure 2 illustrates the overall process of implementing SC using a CoT prompt generated by the teacher LLM.

4.1.4 Reformulating the Instructions

Although the techniques explored in the previous subsections improved model performance, the results remained insufficient to achieve satisfactory filtering quality. In particular, the Mistral 7B model et al. (2023) — despite its efficiency and speed — frequently produced contradictory responses, sometimes accompanied by incorrect or internally inconsistent explanations.

As also observed by Gudibande et al. Gudibande et al. (2023), SLMs tend to mimic the reasoning structure of large teacher models without really understanding the underlying logic. This limitation reduces their ability to accurately detect errors in noisy data.

We then formulated two hypotheses to explain the persistence of false positives:

Prompt Wording: Our initial prompt asked whether a “skill” was required for a given job, while some of the inputs to be evaluated were not skills at all (e.g., “job search,” “10,” “thread”). This lexical bias may have led the model to validate such terms by default.

Model Size: Despite recent advances, a model’s overall knowledge and reasoning capabilities remain strongly correlated with its size. Smaller models struggle to generalize or to effectively exploit limited contextual clues.

To test the first hypothesis, we designed a revised prompt that emphasizes the automatic and potentially noisy nature of the candidate terms to be assessed. The aim was to free the model from the implicit assumption that “this term is a skill” and encourage it to more readily reject vague or irrelevant inputs:

> You are a job market expert. You are given a job and a candidate skill. This skill was extracted automatically and may be incorrect, irrelevant, or too vague. Your task is to determine whether this skill is:
>
> yes: a technical skill that is truly necessary for this job; no: a behavioral or transversal skill, or a skill that does not correspond to this job (e.g., vague, redundant, irrelevant, etc.).
>
> Answer only ”yes” or ”no”, followed by a short explanation.

4.2 Ensembling LLMs

To test the second hypothesis (the effect of model size), we compared the performance of Mistral 7B with five larger models, each evaluated using both the initial and the revised prompt.

Then, inspired by the performance gains observed with self-consistency, we explored the potential of model ensembling using these larger models, which already demonstrated satisfactory results individually. For each occupation–skill pair, we performed a majority vote across the predictions of the five models. Figure 2 illustrates the overall principle of this voting process.

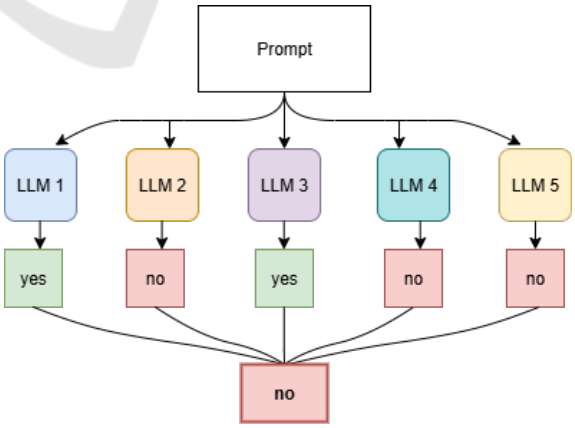

Figure 3: Schematic representation of the ensembling.

5 EXPERIMENTS

In this section, we present our experiments and results, organized into three parts corresponding to the main stages of our methodology.

5.1 Experimental and Evaluation Protocol

We focused on a subset of the complete ontology by selecting only the occupations and skills from the ”Telecoms, Hosting, Internet” activity domain. This subset includes 294 skills linked to 5 occupations. After a thorough manual analysis, we determined that some repeated skills could be grouped by occupation, resulting in a final set of 289 distinct skills. A manual evaluation by three domain experts was performed on the obtained results. The inter-annotator agreement, measured using Fleiss' Kappa Fleiss (1971) reached 0.75, indicating a substantial level of agreement, while also highlighting the inherent difficulty of the task. We conducted a second review to reassess the points of disagreement. Most of these centered on skills that also tended to confuse the language models, falling primarily into two categories: soft skills and vague or ambiguous terms. Given that the ontology already provides a separate list of soft skills for each occupation, we chose to retain only hard (technical) skills. Consequently, vague terms such as “integration” or “infrastructure,” as well as soft skills like “team leader,” were annotated as “no.”

Given that the subset of data used is imbalanced, with 173 instances labeled ”yes” and 116 ”no,” we report Precision, Recall, F-score metrics for both classes and overall accuracy to ensure a more reliable evaluation.

5.2 Results

5.2.1 Fine-Tuning an Encoder-Only Model

Since the data subset is in French, we selected CamemBERT-base Martin et al. (2020), a RoBERTa-based model, for fine-tuning. The dataset (occupation–skill pairs) was framed as a binary classification problem: 1 = positive, 0 = negative. Positive pairs came from ESCO. As ESCO lacks negatives, we tested three negative mining strategies (cf. methodology). Two dataset versions were used: (i) a preprocessed one with lexical noise; (ii) a cleaned version where skills were rewritten automatically (e.g., *respect echeanciers* → *respect des échéanciers*).

Training used identical hyperparameters (3 epochs, batch size: 32, learning rate: $1e$-5). Results are shown in Table 3.

5.2.2 Prompt Engineering

Zero-Shot. With Mistral 7B, the basic prompt *“Does occupation X require skill Y?”* revealed: (i) a strong bias toward “yes”; (ii) systematic inclusion of soft skills; (iii) lexical variability in outputs; (iv) sensitivity to wording.

Few-Shot. Using four annotated examples, Mistral reproduced the demo format but also inherited the class imbalance, yielding extreme “yes” bias (only 15 “no” predictions).

CoT and SC. Chain-of-Thought methods gave modest gains but increased costs. Comparing Universal Self-Consistency (USC) and majority-vote SC showed SC was usually superior with zero-CoT prompts. Table 4 shows results.

New Prompt. To improve results, we designed a revised prompt (see Methodology). Tested zero-shot with several LLMs, it significantly boosted accuracy (Table 5).

5.3 Discussion

5.3.1 Negative Mining Strategies

For encoder-only models, we observed that: **Easy negatives** caused overfitting: near-perfect accuracy on ESCO but poor generalization. **Random negatives** offered the best trade-off, with balanced precision/recall and stable training. **Hard negatives** slowed convergence but improved robustness, especially on our manual dataset.

Fine-tuned models consistently performed better on Mistral-corrected inputs, confirming the role of lexical clarity. BERT models underperform LLMs, partly because ESCO skills are long and generic, while our ontology emphasizes concise, technical terms. This limits BERT's vocabulary alignment and cross-dataset generalization.

5.3.2 LLM-Based Evaluations

LLMs showed both strengths and limitations. Small models (e.g., Mistral 7B) lacked reasoning depth and were prompt-sensitive. Larger ones (12–14B) showed biases and often accepted vague terms. Intra-model variability remained an issue across runs.

These findings motivated an ensemble strategy using majority voting, which reduced inconsistencies. Compared with GPT-4o/4.1, large open-source models were already competitive, yet our ensemble consistently outperformed both them and individual GPT baselines (Table 6).

Table 3: Results with fine-tuned CamemBERT.

Model	Data	Method	Precision		Recall		F1		Accuracy
			yes	no	yes	no	yes	no	
CamemBERT	Preproc.	random neg.	0.63	0.54	0.87	0.23	0.73	0.33	0.61
		easy neg.	0.59	0.09	0.94	0.01	0.72	0.02	0.57
		hard 95%	0.66	0.53	0.75	0.42	0.70	**0.47**	0.62
		hard 90%	**0.81**	0.35	0.65	**0.55**	0.72	0.43	0.63
	Cleaned	random neg.	0.63	0.77	0.97	0.17	0.77	0.28	0.65
		easy neg.	0.60	–	**1.00**	0.00	0.75	–	0.60
		hard 95%	0.65	0.64	0.89	0.29	0.75	0.40	0.65
		hard 90%	0.66	**0.78**	0.95	0.27	**0.78**	0.39	**0.68**

Table 4: Performance of Mistral 7B with prompting methods.

Model	Method	Precision		Recall		F1		Accuracy
		yes	no	yes	no	yes	no	
Mistral 7B	zero-shot	0.62	0.53	0.87	0.22	0.72	0.31	0.61
	few-shot	0.65	0.74	0.94	0.25	0.77	0.37	0.66
	zero-shot CoT	**0.68**	0.56	0.75	**0.47**	0.71	**0.51**	0.64
	CoT Mistral	0.65	0.65	0.90	0.28	0.76	0.39	0.65
	CoT ChatGPT	0.65	**0.80**	**0.96**	0.24	**0.78**	0.37	0.67
	CoT SC	0.67	0.70	0.90	0.34	0.77	0.46	0.67
	CoT USC	0.64	0.60	0.89	0.25	0.74	0.35	0.63

Table 5: Performance with old vs. new zero-shot prompt.

Model	Prompt	Precision		Recall		F1		Accuracy
		yes	no	yes	no	yes	no	
Mistral 7B	old	0.62	0.53	0.87	0.22	0.72	0.31	0.61
	new	0.74	0.66	0.80	0.58	0.77	0.61	0.71
Gemma-3 12B	old	0.68	0.69	0.89	0.36	0.77	0.48	0.68
	new	0.88	**0.91**	**0.95**	0.81	**0.91**	**0.86**	**0.89**
DeepSeek-R1 14B	old	0.68	0.53	0.68	0.53	0.68	0.53	0.62
	new	0.85	0.85	0.91	0.77	0.88	0.81	0.85
Phi-4 14B	old	0.61	0.42	0.55	0.48	0.58	0.45	0.52
	new	0.88	0.90	0.94	0.82	**0.91**	**0.86**	**0.89**
Qwen-3 14B	old	0.72	0.51	0.58	0.65	0.64	0.58	0.61
	new	**0.91**	0.81	0.86	**0.87**	0.88	0.84	0.86

Table 6: Ensemble vs GPT models with new zero-shot prompt.

Model	Prompt	Precision		Recall		F1		Accuracy
		yes	no	yes	no	yes	no	
Ensemble	new	0.93	0.91	0.94	0.90	0.94	0.90	**0.92**
GPT-4o	new	0.77	0.92	0.97	0.57	0.86	0.70	0.81
GPT-4.1	new	0.86	0.88	0.93	0.77	0.89	0.82	0.86

6 CONCLUSION

This study explored several strategies to improve the automatic validation of occupation–skill relations in an ontology, combining fine-tuned encoder-based models and prompt-based LLM evaluations. We demonstrated that hard negative mining yields more robust classification for encoder models, especially when coupled with input correction. In parallel, prompt engineering and reasoning-based prompting (CoT, self-consistency) improved LLM performance, though limitations persisted—particularly in smaller

models. To address these, we proposed an ensemble approach that outperformed all individual models, including proprietary LLMs like GPT-4, highlighting its potential as a lightweight yet effective alternative for ontology curation tasks.

Despite remaining challenges, our work opens promising directions for automating knowledge base validation and enrichment. In future work, we aim to investigate fine-tuning strategies for LLMs to improve their reasoning on domain-specific tasks. Another perspective involves adapting our methods to different domains and ontological structures. We also see potential in integrating external knowledge sources, such as curated databases of occupations and skills, to enhance LLM interpretability and decision-making. Finally, assessing the impact of these methods on real-world applications, like recommendation systems or career guidance platforms, would be an essential step toward validating their practical value.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge InterMEDIUS for the partial funding of this work.

REFERENCES

Abolhasani, M. S. and Pan, R. (2024). Leveraging llm for automated ontology extraction and knowledge graph generation.

Brown, T. B. and al. (2020). Language Models are Few-Shot Learners. arXiv:2005.14165 [cs] version: 1.

Chen, X., Aksitov, R., Alon, U., Ren, J., Xiao, K., Yin, P., Prakash, S., Sutton, C., Wang, X., and Zhou, D. (2023). Universal Self-Consistency for Large Language Model Generation. arXiv e-prints, page arXiv:2311.17311.

de Souza P. Moreira, G., Osmulski, R., Xu, M., Ak, R., Schifferer, B., and Oldridge, E. (2025). Nv-retriever: Improving text embedding models with effective hard-negative mining.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] version: 1.

et al., A. Q. J. (2023). Mistral 7b.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5):378–382. Place: US Publisher: American Psychological Association.

Gudibande, A., Wallace, E., Snell, C., Geng, X., Liu, H., Abbeel, P., Levine, S., and Song, D. (2023). The False Promise of Imitating Proprietary LLMs. arXiv e-prints, page arXiv:2305.15717.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, Advances in Neural Information Processing Systems, volume 35, pages 22199–22213. Curran Associates, Inc.

Kommineni, V. K., König-Ries, B., and Samuel, S. (2024). From human experts to machines: An llm supported approach to ontology and knowledge graph construction.

le Vrang, M., Papantoniou, A., Pauwels, E., Fannes, P., Vandensteen, D., and De Smedt, J. (2014). Esco: Boosting job matching in europe with semantic interoperability. Computer, 47(10):57–64.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a tasty French language model. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7203–7219, Online. Association for Computational Linguistics.

Meyer, L.-P., Stadler, C., Frey, J., Radtke, N., Junghanns, K., Meissner, R., Dziwis, G., Bulert, K., and Martin, M. (2024). Llm-assisted knowledge graph engineering: Experiments with chatgpt. In Zinke-Wehlmann, C. and Friedrich, J., editors, First Working Conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow, pages 103–115, Wiesbaden. Springer Fachmedien Wiesbaden.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. (2019). Language models as knowledge bases? arXiv preprint arXiv:1909.01066.

Ranaldi, L. and Freitas, A. (2024). Aligning Large and Small Language Models via Chain-of-Thought Reasoning. In Graham, Y. and Purver, M., editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1812–1827, St. Julian's, Malta. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. (2021). Contrastive learning with hard negative samples.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., and Zhou, D. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs] version: 1.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs] version: 1.