Automating Resume Analysis: Knowledge Graphs via Prompt Engineering

Giorgio Lazzarinetti[1,2][0000−0003−0326−8742] Sara Manzoni[1][0000−0002−6406−536X] and Italo Zoppis[1][0000−0001−7312−7123]

[1] Università degli Studi Milano-Bicocca, Milano, Italy,
giorgio.lazzarinetti@unimib.it, sara.manzoni@unimib.it,
italo.zoppis@unimib.it
[2] BID Company S.r.l., Milano, Italy, https://www.bidcompany.it/

Abstract. The rapid digitization of recruitment processes and the growing complexity of resume data have posed significant challenges in managing and extracting information from such sources. Traditional methods necessitate innovative approaches that can adapt and scale effectively. This paper introduces a methodology employing Large Language Models (LLMs) facilitated by advanced prompt engineering techniques, to construct Knowledge Graphs (KGs) directly from resumes. Our approach bypasses the extensive customization typically required for domain-specific tasks, leveraging the intrinsic capabilities of LLMs to interpret and organize complex data. We evaluate our methodology, focusing particularly on Named Entity Recognition (NER) as a measure of effectiveness. The results demonstrate superior performance of our system against baseline models. Additionally, we explore the practical applicability of our system through a novel self-consistency metric, which further attests to the method's ability to accurately capture and reproduce essential resume information in KG format. This study not only underscores the potential of LLMs in automated information extraction but also opens up new avenues for research and application in the HR technology domain and beyond.

Keywords: Knowledge Graph · Resume Analysis · Deep Learning · Large Language Model · Prompt Engineering · Text to Graph.

1 Overview

Online recruitment platforms such as LinkedIn have revolutionized job advertising, offering significant time efficiencies for employers and job seekers alike. However, the increasing volume of data on these platforms complicates the effective analysis of each resume, a challenge that has attracted considerable research interest [1]. Resumes, which are primarily text-based and lack a uniform format, contain diverse information types, creating structural uncertainties.

Traditionally, resume review was a manual task where HR professionals extracted information and matched skills against job descriptions to identify suitable candidates [2]. To overcome the limitations of manual processes, automated techniques like keyword retrieval and Document Object Model (DOM) tree-based methods have been developed [1]. While keyword retrieval is straightforward, it often lacks accuracy due to textual noise [2–4]. DOM tree approaches, on the other hand, face scalability challenges because they depend on manual input and are template-specific [5].

Recent advances in Natural Language Processing (NLP) have introduced machine learning-based methods, predominantly using Named Entity Recognition (NER) and pattern matching, to extract and structure resume information [6–8]. Despite their efficiency in handling large volumes of data, these methods struggle with complex, time-sensitive information and scaling to new, unseen entities in a dynamic job market [9].

To address these issues, the use of Knowledge Graphs (KGs) has gained traction in resume parsing and skill matching, providing a more expressive framework than traditional tables for organizing and analyzing information [10–12]. However, constructing KGs from unstructured texts remains a challenging task, often tackled using various Deep Learning (DL) methods [13], such as Large Language Models (LLMs) pre-trained on extensive corpora and fine-tuned on real data [14–16]. However, only recently, the increasing size of these models has enabled LLMs to perform complex, practical tasks effectively even without fine-tuning, benefiting significantly from advancements in prompt engineering — a burgeoning field aimed at optimizing prompt design to enhance LLMs performance across different applications and research initiatives [17].

Despite the proliferation of tools and applications for managing resume data, there has been limited research focused on assessing the effectiveness of prompt-based techniques for constructing KGs from text, particularly resumes. In this study, we investigate whether modern LLMs can successfully build KGs from resumes in a way similar to a professional human analyst. The main goal of this research is to explore the feasibility and effectiveness of prompt engineering in automating the resume parsing process. For this reason we propose a prompt-based methodology to create KGs from resumes using a pre-trained LLM, aiming to provide both theoretical insights into specific prompting techniques and a comprehensive evaluation of the proposed method.

The paper is structured as follows: Section 2 introduces LLMs and prompt engineering techniques; Section 3 outlines our approach to convert texts to KGs, the ontology for resume parsing and the evaluation schema; Section 4 discusses the datasets used and experimental results; Section 5 offers final thoughts.

2 Background

2.1 Large Language Models

LLMs are advanced artificial intelligence systems that utilize DL techniques to understand, interpret, and generate human language. They are pre-trained on expansive corpora and have demonstrated remarkable capabilities across a diverse array of NLP tasks [18]. A fundamental component underpinning the success of most LLMs is their reliance on the Transformer architecture [19], which is distinguished by its incorporation of encoder and decoder modules, both of which are enhanced through the self-attention mechanism. Predicated on their architectural framework, LLMs can be systematically classified into one of three distinct categories: (i) encoder-only models like BERT [14] and RoBERTa [15], which are adept at understanding word relationships and are used in tasks such as text classification and NER; (ii) encoder-decoder models such as T5 [16], which excels in context-based sentence generation tasks like summarization, translation, and question answering; and (iii) decoder-only models, including GPT-4 [20], which focus on generating text from minimal prompts without additional modifications [21]. Decoder-only models excel in free-form text generation, creative writing, and dialog systems where the generation of coherent and contextually relevant responses is crucial. The flexibility of these models is particularly advantageous in scenarios requiring adaptive responses or generating content based on sparse inputs.

2.2 Prompt Engineering

Prompt engineering is an area of inquiry that concentrates on the formulation and refinement of prompts to enhance the performance of LLMs across a myriad of applications and research domains [17]. Within this framework, a prompt is conceptualized as a sequence of natural language inputs tailored to a specific task, such as sentiment classification, and is composed of distinct elements: (i) *instruction*, i.e., a concise directive that guides the model in executing a particular task; (ii) *context*, which furnishes the relevant backdrop for the input text or supplies few-shot examples; and (iii) *input text*, denoting the textual content subject to the model's processing. This discipline endeavors to augment the efficacy of LLMs in executing an array of complex tasks, encompassing question answering, sentiment analysis, and the elucidation of common sense reasoning.

Several primary techniques have been identified, each contributing distinctively to the functionality of LLMs. Among these, Zero-shot Learning (ZsL), Few-shot Learning (FsL), and Chain-of-Thought (CoT) are particularly prevalent. ZsL allows models to execute tasks based solely on prompt instructions, showcasing their ability to generalize across various tasks [22]. FsL enhances this by incorporating a limited number of task examples within the prompt, enabling LLMs to leverage pattern recognition to understand and perform tasks [21]. CoT, in contrast, prompts the model to articulate intermediate steps or reasoning processes before producing a final answer, facilitating deeper engagement with complex reasoning tasks by encouraging the model to "think aloud" [23]. Additionally, as prompt engineering is an active area of research, several innovative techniques have emerged. Notable among these are Retrieval Augmented Generation [24], Tree-of-Thought Prompting [25], and Graph-of-Thought Prompting [26], each offering sophisticated methods to enhance the effectiveness and applicability of LLMs in diverse computational tasks.

3 Prompt-based Solution

3.1 Text to Knowledge Graph Pipeline

To design our framework we begin with an analysis of the prevailing methods in the literature for constructing KGs from textual data [13]. The development of a KG involves multiple phases, each tailored to overcome specific obstacles encountered during the conversion of unstructured text into a structured form of knowledge representation.

1. **Entity Recognition and Classification**: detects and classifies entities within the text, such as people, places, and organizations, which are essential for constructing nodes in the KG.
2. **Relationship Extraction**: determines the relationships between entities, which form the edges of the KG, crucial for mapping the connections within the graph.
3. **Entity Disambiguation and Linking**: addresses the challenge of distinguishing between entities with similar names and correctly linking them to existing entries in a knowledge base, ensuring accurate entity representation.
4. **Knowledge Integration**: entities and relationships are integrated into an existing KG or used to create a new one. It involves resolving inconsistencies and integrating new knowledge using technologies like RDF and SPARQL.
5. **Knowledge Refinement and Enrichment**: the KG is continuously refined and updated with new information, corrections, and enhancements to improve its accuracy and quality.

3.2 Resume Data Schema

The outlined stages present a generalized framework for transforming textual data into a KG, emphasizing the need for adaptation to specific business domains for accurate representation. In the context of our research, we focus on candidate resumes, necessitating a tailored approach to precisely identify domain-specific concepts. Various ontologies for modeling resumes and job offers have been suggested [27–29]. Referencing them, we seek to define the key entities, attributes, and relationships that should be extracted from resumes to accurately and flexibly represent real-world scenarios. Our schema, detailed in Table 1, provides a comprehensive view of an individual's professional and academic achievements. We have specified attributes for each entity to reflect essential and commonly encountered data. In the proposed schema, we identify seven principal entities which focus on distinct yet interconnected aspects of a professional profile:

- **Person**: acts as the central node, linking all other entities, with attributes including name, contact details, and demographics.
- **Education**: details academic qualifications and training, with attributes like institution name, degree, field of study, and graduation dates.
- **Professional Experience**: outlines employment history, including employer names, job titles, employment dates, and responsibilities.

– **Skill**: covers specific competencies, both technical and soft, with attributes such as skill name, proficiency level, and years of experience.
– **Certification**: documents certifications and licenses, with attributes including certification name, issuing organization, and validity.
– **Achievement**: highlights significant accomplishments, including awards or recognitions, with relevant details such as the award name, issuing body, and date.
– **Publication**: represents scholarly work, with attributes including title, description, publication date, and outlet.

Table 1. KG's schema for individuals derived from resume data, detailing seven primary entities along with their respective attributes and relationships.

Entities	Attributes	Relationships
Person	Name, Contact Information, (email, phone number), Location, LinkedIn Profile	HasEducation, HasExperience, HasSkill, HasCertification, HasAchievement, HasPublication
Education	Degree, Field of Study, Educational Institution, Start Date, End Date	ObtainedBy (inverse of HasEducation), RelatedToField
Professional Experience	Job Title, Company Name, Industry, Start Date, End Date, Responsibilities, Achievements	UndertakenBy (inverse of HasExperience), InIndustry, UsesSkill
Skill	Skill Name, Proficiency Level, Years of Experience	UsedInJob (inverse of UsesSkill)
Certification	Certification Name, Issuing Organization, Issue Date, Expiry Date	ObtainedBy (inverse of HasCertification), RelatedToSkill
Achievement	Achievement Title, Description, Date,	AchievedBy (inverse of HasAchievement), RelatedToSkill, RelatedToExperience
Publications	Title, Description, Date, Journal, Conference	PublishedBy (inverse of HasPublication), RelatedToSkill, RelatedToExperience

3.3 CoT Prompt for Resume to Knowledge Graph

Analyzing the general pipeline for converting text into a graph detailed in Section 3.1 reveals a highly sequential process that incorporates logical reasoning at various stages to enhance the outcome. CoT prompting, as used with language models for complex problem-solving or reasoning tasks, exemplifies this approach. It involves guiding the model through intermediate steps or reasoning paths towards a conclusion, similar to human problem-solving strategies. This method is particularly useful where direct answers require information synthesis, logical reasoning, or nuanced context understanding, leveraging the model's capability to generate relevant text sequences from structured prompts.

Consequently, we crafted a specialized CoT prompt that integrates the standard procedures for text-to-graph conversion, with tailored modifications to align with HR-specific requirements. The structure of the prompt is outlined in Algorithm 1. The prompt sets out clear instructions for the LLM on handling entities (nodes), their identifiers, and the treatment of numerical data and dates. We recommend using simple, textual identifiers, particularly for the *Person* entity, to keep the node identification straightforward and interpretable. Numerical data and dates are treated as attributes, not nodes, to minimize complexity.

Algorithm 1: Resume2KnowledgeGraph Prompt Main Structure

You are an algorithm designed for extracting information in structured formats from resumes to build a knowledge graph. In order to do it consider the following statements.

Preliminary Considerations

- **Nodes**: Nodes represent entities and concepts. Ensure you use basic or elementary types for node labels. For example, when you identify an entity representing a person, always label it as "person".
- **Nodes ID**: Never utilize integers as node IDs. Node IDs should be names or human-readable identifiers found in the text.
- **Numerical Data and Dates**: Numerical data, like age or other related information, should be incorporated as attributes or properties of the respective nodes. Always attach them as attributes or properties of nodes. Properties must be in a key-value format.

To build the knowledge graph then follow the steps and the instructions described in Algorithm *Resume2KnowledgeGraph Steps*.

Additional Considerations

- Incorporate nodes for Languages with attributes for proficiency and usage context, linked to Person entities, to capture linguistic capabilities.
- Add nodes for Volunteer Experience similar to Professional Experience, including attributes like Role, Organization, and Date, to capture non-work-related skills and achievements.
- Consider temporal relationships between experiences to infer career progression paths and potential skill development over time.

The extraction methodology is further detailed in Algorithm 2, customized per our schema in Table 1. It includes specific examples and focal points to guide the model in aligning with the predefined schema and includes steps for

Algorithm 2: Resume2KnowledgeGraph Steps Prompt

Step 1: Entity Recognition and Classification

- Action: Scan the resume text to identify and classify entities according to predefined categories: Person, Education, Professional Experience, Skill, Certification, Achievement, and Publications.
- Customization: Pay special attention to keywords and phrases that signify the beginning and end of sections (e.g., "Education", "Experience", "Skills"), and use formatting cues like bullet points and headings to distinguish between different entities and their attributes.

Step 2: Relationship Extraction

- Action: Analyze the context around identified entities to extract relationships between them. This involves understanding how different entities like Education and Person or Professional Experience and Skill are connected.
- Customization: Focus on verbs and prepositions that indicate relationships, such as "earned" for Education and "worked on" for Professional Experience, to map the defined relationships accurately (e.g., HasEducation, UsesSkill).

Step 3: Entity Disambiguation and Linking

- Action: Resolve ambiguities among entities (e.g., differentiating between Java the programming language and Java the island) and link entities to unique identifiers where possible (e.g., using LinkedIn profiles for disambiguation).
- Customization: For the HR domain, prioritize disambiguation of educational institutions, company names, and certification bodies by cross-referencing known databases or lists to ensure accuracy in entity identification.

Step 4: Knowledge Integration

- Integrate the extracted entities and relationships into a cohesive knowledge graph structure, ensuring that each entity is represented as a node with its attributes and that the relationships between entities are accurately depicted as edges.
- Customization: Ensure that nodes for Person entities serve as central hubs, linking to various aspects of their professional profile (Education, Experience, Skills, etc.) to reflect the comprehensive nature of a resume.

Step 5: Knowledge Refinement and Enrichment

- Action: Refine the knowledge graph by checking for consistency, removing duplicates, and filling in missing information. Enrich the graph by adding inferred relationships or attributes (e.g., inferring skill proficiency levels from years of experience or job responsibilities).
- Customization: Consider adding nodes for Industries to connect Professional Experience and Skills, enhancing the graph's utility for HR purposes by facilitating industry-specific analysis and talent mapping.

information validation through cross-referencing and directives on managing the *Person* entity as a central link to other relevant entities.

Additionally, the main prompt incorporates guidance for handling contextual information not covered in the schema, such as language skills or volunteer activities, and advises on assessing temporal relationships to deduce career paths and skills. These additional guidelines aim to ensure the model captures both structured and contextual dimensions of resume data, thus enhancing the utility and accuracy of HR analytics.

3.4 Proposed Evaluation Approach

Evaluating the quality of a KG presents significant challenges, particularly when approached as an unsupervised task. Typically, the evaluation targets individual sub-tasks within the KG construction pipeline such as NER, Triple Extraction (TE) or Entity Linking. In our unified framework, where the output comprises entities, attributes and relations forming the complete KG, it is not feasible to evaluate each sub-task individually. Notably, while TE generally poses challenges in converting texts to KGs, it is somewhat simplified in our context since each entity directly connects to the main entity *Person*, easing the identification process.

To assess the effectiveness of our approach in constructing KGs from resumes, we focus on measuring the NER capability of our solution, given its critical role, using standard classification metrics: Precision, Recall, F1-score and Accuracy. These metrics are computed on a publicly available dataset from Kaggle [30] that includes resume texts alongside identified entities. We benchmark these results against two state-of-the-art methods: a RoBERTa-based model and a spaCy-based model. Finetuning for RoBERTa is performed on the *roberta-base* model [31] with token classification head for NER. All parameters are fine-tuned with Adam [32] optimizer at a learning rate of $E \times 10^{-4}$. Weight decay of 0.01 is applied to all parameters except biases and normalization parameters, which were exempt to stabilize training. Training is conducted over 10 epochs, employing gradient clipping with a max norm of 1.0 to prevent gradient explosion. For the spaCy model, we adopt the *en-core-web-sm* model [33] adding the NER component and training for 20 epochs with a dropout rate of 0.3. Default stochastic gradient descent optimizer is adopted, with its built-in decay mechanism, L2 regularization and gradient clipping strategy [34]. To fine-tune the baseline models we apply a 70-30 ratio split of the dataset.

Additionally, to assess the quality of KGs built from resumes we developed a novel self-consistency metric. Typically, to evaluate correctness, completeness, and consistency of the extracted information used to construct the KG, a ground truth is necessary. However creating a scalable ground truth in real-world scenarios is impracticable. For this reason, we have opted to introduce a self-consistency metric to measure the quality of the KGs generated. This metric involves reverting the generated KG back into a textual resume format using GPT-4 (with a simple yet effective ZsL approach), then quantitatively comparing the regenerated resume to the original one. For this comparison, we use a pre-trained sentence transformer [35] to compute the cosine similarity between embeddings of the original and regenerated texts of the resume. To interpret these similarity scores, we create a dataset comprising triplets of resumes: a reference resume, a second resume deemed similar by human evaluators, and a third one deemed different. We calculate average cosine similarities for both similar and different resume sets. The effectiveness of our method is gauged by comparing these similarity scores, aiming for our results to approach the upper similarity bounds established by the comparable resumes. Although this task involves a translation model, we have chosen not to use standard translation model evaluation metrics (e.g., BLEU, ROUGE, or METEOR), since transforming the KG into a resume is not done in a supervised manner, as is typical for translation models. Consequently, because the ultimate goal is not to produce a resume that closely mirrors the original one but rather one that includes all essential information from the original resume, we use this self-consistency metric that correctly quantifies the fidelity of the information retained in the KG.

4 Experimental Results

4.1 Datasets Description

As mentioned in Paragraph 3.4, to test our methodologies we used three datasets.

The *NER Dataset*, is a dataset publicly available on Kaggle [30] that contains resumes and tagged entities for NER. It is composed of 220 resumes and a total of 3556 entities divided in the following 10 types: Name, Email Address, Location, College Name, Degree, Graduation Year, Companies Worked at, Designation, Skills, Years of Experience. Each of these maps to the set of entities or attributes designed in Table 1. This dataset is used to evaluate the performance of the LLM in extracting entities after steps 1 to 3 of Algorithm 2 have been performed.

The *Resumes Dataset*, used to directly measure the information extraction and KG construction capabilities of the algorithms together, is composed of 188 resumes in PDF format. These are resumes from professionals of different sectors (e.g., IT, Banking, Finance, Fashion, Food, Industry) and with different seniority level (e.g., junior, mid and senior). These resumes have been synthetically generated starting from real-world data collected from LinkedIn.

Finally, the *Triplets Dataset*, is composed of 50 triplets of resumes. We randomly extract 50 resumes from the Resumes Dataset, and for each of these reference resumes we manually build a similar resume so that it contains all the information of the reference one expressed in a different way. Finally, we add a third resume to the triplet, by getting one from the same sector and possibly similar professional experience, but with different information. As an example, Table 2 shows the similar sentence and the different sentence built as part of the triplet with respect to a reference sentence. This dataset is used as ground truth to benchmark the performance of the LLM in building the KG from the resume, based on its ability to build the resume back from the constructed KG.

Table 2. Example of reference sentence, similar sentence and different sentence from the Triplet Dataset used to evaluate the approach in a qualitative way.

Reference Sentence	Computer Science engineer with one year of experience in DevOps. Competent in using various automation tools and able to work in environments with agile methodologies. Motivated to constantly improve technical skills and to bring innovation within projects.
Similar Sentence	Passionate Computer engineer with a year's experience in DevOps, skilled in employing diverse automation tools and adept at operating within agile methodologies. Eager to continuously enhance technical abilities and introduce new techniques and tools in projects.
Different Sentence	DevOps engineer with 6 years of experience in building and managing scalable and reliable infrastructures. Expert in using CI/CD tools to optimize development and deployment operations. Skilled in team collaboration and solving complex problems.

4.2 Named Entity Recognition

As described, the effectiveness of NER directly influences the accuracy and integrity of the resultant KG. To assess the quality of NER in our study, we employed classical NER performance metrics: Precision, Recall, F1-Score, and Accuracy. These metrics provide a comprehensive view of our model's capability to correctly identify and classify entities within resumes. They are presented in Table 3, which shows a comparative analysis between the baseline models — a fine-tuned RoBERTa model and a fine-tuned spaCy model — and our proposed CoT prompt-based methodology applied on the GPT-4 model.

Table 3. Results of the NER task in terms of Precision, Recall, F1-score and Accuracy, that show how the CoT prompt-based model outperforms the baseline models.

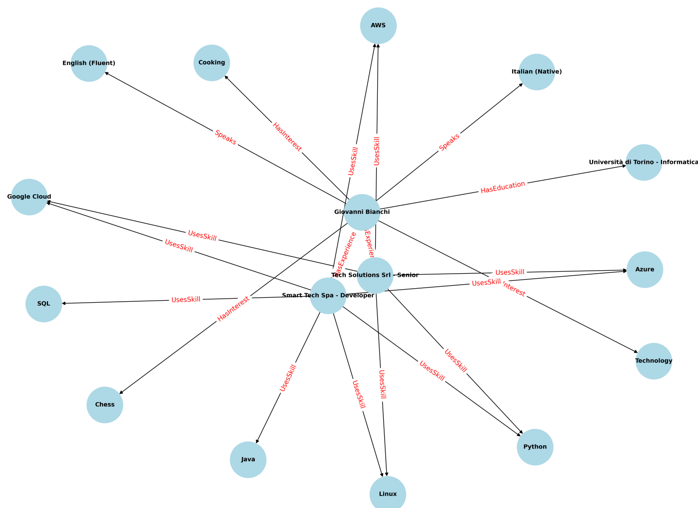
Model	Precision	Recall	F1-score	Accuracy
spaCy	0,86	0,85	0,86	0,86
RoBERTa	0,91	0,89	0,90	0,89
GPT-4	0,94	0,90	0,92	0,92

Results are computed on the 30% of test data (73 resumes with a total of 1175 entities). Here, our methodology demonstrates superior performance across all metrics. These metrics signify not only an enhancement in identifying correct entities but also in reducing false positives and negatives, crucial for building reliable KG. To visualize the GPT-4 model's capability to extract entities based on our CoT prompt, Figure 1 shows an example of extracted KG from a resume.

4.3 Self Consistency Metric

As mentioned, to evaluate the quality of the KGs generated, we compute a self-consistency metric by serving the KGs generated from the original resume texts

Fig. 1. Example of a generated KG with Person as central entity connected to information about Education, Professional Experience, Languages, Interests and Skills.



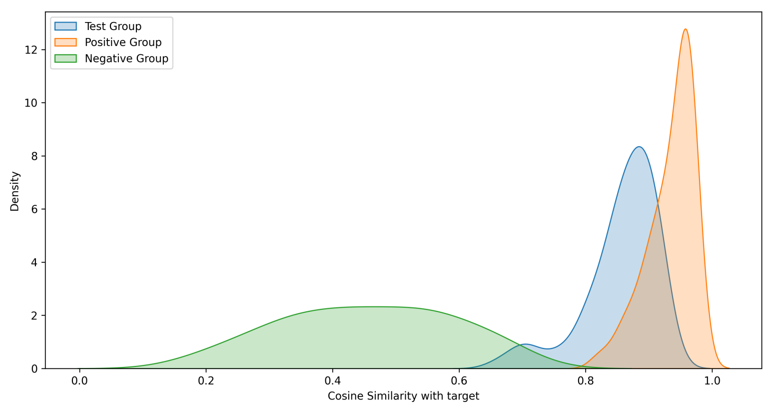
as input to GPT-4 to regenerate, with a ZsL prompt-based approach, the resume texts and subsequently measuring the cosine similarity between these texts (properly embedded with a pre-trained sentence transformer) to assess fidelity and accuracy. The effectiveness of our KG-based reconstruction is quantified by comparing the cosine similarity scores of the regenerated resumes against those of two control groups: a set of similar and a set of different resumes. These control groups, composed of 50 resumes serve to establish benchmark similarity ranges for evaluation. The results of these comparisons are summarized in Table 4, which shows that the average cosine similarity score between the original resumes and their KG-based reconstructions (Test Group) is 0.876. This closely approaches the 0.942 average for manually identified similar resumes (Positive Control Group), indicating that the KG effectively captures and reproduces key resume information. Only a few reconstructions fall below the threshold set as

Table 4. Comparison of cosine similarity scores on the Triplets Dataset (with a support of 50 resumes) and on test instances (with a support of 220 resumes).

Control Group	Min Sim	Avg Sim	Max Sim	Support
Positive	0,823	0,942	0,977	50
Negative	0,182	0,46	0,698	50
Test	0,655	0,876	0,932	220

the maximum similarity score of the Negative Control Group (0.698), with the lowest of the Test Group being 0.655. Moreover, the average similarity with the Negative Control Group at 0.46 is significantly lower with respect to the lowest of the Test Group, clearly differentiating between relevant and non-relevant content in the KG reconstructions, as depicted in Figure 2. These results highlight the precision of the information captured by our KG-based method in replicating resume content, demonstrating performance nearly equivalent to human-judged similar texts and significantly surpassing the threshold for non-similar texts.

Fig. 2. Distribution of the cosine similarity of each control group with respect to target.



5 Conclusion and Future Works

In this paper, we investigated whether modern LLMs can successfully build KGs from resumes akin to professional human analyst. We designed a CoT prompt-based methodology that leverages advanced LLMs to convert texts into KGs that encapsulate professional profiles. The primary objective of our research was to explore the feasibility and effectiveness of prompt engineering in automating the resume parsing process and subsequently creating detailed KGs. We focused on NER to benchmark the effectiveness of our system against traditional models, showing that our approach achieved superior performance compared to state-of-the-art models without extensive fine-tuning. We also propose a self-consistent approach to measure the ability of our method in generating KG from resumes. The obtained results suggests that our method can accurately capture and reproduce critical information from resumes.

While our results are promising, several avenues remain open for future exploration, such as testing different LLMs or prompting strategies, ethical considerations, the integration with real-time systems and cross-domain adaptability.

References

1. Chen, J., Zhang, C., & Niu, Z.: A Two-Step Resume Information Extraction Algorithm. Mathematical Problems in Engineering **2018**(1), 1–8 (2018)
2. Jagwani, V., Meghani, S., Pai, K., & Dhage, S.: Resume Evaluation through Latent Dirichlet Allocation and Natural Language Processing for Effective Candidate Selection. arXiv preprint, arXiv:2307.15752 (2023)
3. Maheshwari, S., Sainani, A., Reddy, P.K.: An approach to extract special skills to improve the performance of resume selection. In: Databases in Networked Information Systems, LNCS, vol. 5999, pp. 256–273. Springer, Berlin, Germany (2010)
4. Kopparapu, S.K.: Automatic extraction of usable information from unstructured resumes to aid search. In: 1st IEEE International Conference on Progress in Informatics and Computing (PIC '10), vol. 1, pp. 99–103. IEEE, China (2010)
5. Ji, X., Zeng, J., Zhang, S., Wu, C.: Tag tree template for Web information and schema extraction. Expert Systems with Applications **37**(12), 8492–8498 (2010)
6. Rasal, P., Balwaik, Y.: Resume Parser Analysis Using Machine Learning And Natural Language Processing. International Research Journal of Modernization in Engineering Technology and Science **5**(3), (2023)
7. Bhaliya, N., Gandhi, J., Singh, D.K.: NLP based Extraction of Relevant Resume using Machine Learning. IJITEE (2020)
8. Bhor, S., Gupta, V., Nair, V., Shinde, H., Kulkarni, M.: Resume Parser Using NLP Techniques. www.ijres.org **9**(6), 01–06 (2021)
9. de Groot, M., Schutte, J., Graus, D.: Job Posting-Enriched Knowledge Graph for Skills-based Matching. In: RECSYS in HR 2022, Co-located with the 16th ACM Conference on Recommender Systems, US (2022)
10. Goyal, N., Kalra, J. S., Sharma, C., Mutharaju, R., Sachdeva, N., Kumaraguru, P.: JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks. In: Vlachos, A., Augenstein, I. (eds.) Findings of the Association for Computational Linguistics: EACL 2023, pp. 2181–2191. Association for Computational Linguistics, Dubrovnik, Croatia (2023)
11. Wang, Y., Allouache, Y., Joubert, C.: Analysing CV Corpus for Finding Suitable Candidates using Knowledge Graph and BERT. In: 13th International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2021), pp. 256–273. Valencia, Spain (2021)
12. Goyal, N., Kalra, J.S., Sharma, C., Mutharaju, R., Sachdeva, N., Kumaraguru, P.: JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks. In: Vlachos, A., Augenstein, I. (eds.) Findings of the Association for Computational Linguistics: ACL 2023, pp. 2181–2191. Association for Computational Linguistics, Dubrovnik, Croatia (2023).
13. Zhong, L., Wu, J., Li, Q., Peng, H., Wu, X.: A Comprehensive Survey on Automatic Knowledge Graph Construction. ACM Computing Surveys **56**, (2023). https://doi.org/10.1145/3618295
14. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining

Approach. In: Proceedings of the 23rd Conference on Empirical Methods in Natural Language Processing (EMNLP 2019), pp. 4248–4258. Association for Computational Linguistics, Hong Kong, China (2019)
16. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020)
17. Amatriain, X.: Prompt Design and Engineering: Introduction and Advanced Methods.arXiv preprint, arXiv:2401.14423 (2024)
18. Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., Hu, X.: Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. ACM Trans. Knowl. Discov. Data **18**(6), Article 160, 32 pages (2024)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA (2017)
20. OpenAI: GPT-4 Technical Report. arXiv preprint, arXiv:2303.08774 (2023)
21. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020)
22. Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned Language Models Are Zero-Shot Learners. In: International Conference on Learning Representations (ICLR 2022), pp. 1–2. (2022)
23. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: 36th Conference on Neural Information Processing Systems (NeurIPS 2022), pp. 1882–1893. (2022).
24. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), pp. 9459–9474. Vancouver, Canada (2020).
25. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., Narasimhan, K.: Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), NeurIPS (2023)
26. Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., Hoefler, T.: Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In: 30th International Conference on Neural Information Processing Systems (NeurIPS 2023), pp. 15682–15690. ETH Zurich, Zurich, Switzerland (2023)
27. Zaouga, W., Ben Arfa Rabai, L.: Modeling and Evaluating a Human Resource Management Ontology. In: Computer Science On-line Conference, pp. 1–2. (2019)
28. Ahmed, N., Khan, S., Latif, K.: Job Description Ontology. In: 2016 International Conference on Frontiers of Information Technology (FIT), pp. 217–222. Islamabad, Pakistan (2016).
29. Javed, Z., Qazi, H., Khoja, S.A.: An Ontology-based Knowledge Management Model for e-Recruitment utilizing MOOCs Data. In: 2019 8th International Conference on Information and Communication Technologies (ICICT), pp. 124–128. Karachi, Pakistan (2019).
30. Kaggle - Resume Entities for NER, `https://www.kaggle.com/datasets/dataturks/resume-entities-for-ner/data`, last accessed 2024/06/09

31. Hugging Face - FacebookAI - RoBERTa Base Model, https://huggingface.co/FacebookAI/roberta-base, last accessed 2024/08/26
32. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. in proceeding of the 3rd International Conference on Learning Representations (ICLR) , San Diego, USA (2015).
33. Hugging Face - spaCy - en-core-web-sm Model, https://huggingface.co/spacy/en_core_web_sm, last accessed 2024/08/26
34. SpaCy - Initialize Method, https://spacy.io/api/language#initialize, last accessed 2024/08/26
35. Hugging Face - Sentence Transformers, https://huggingface.co/sentence-transformers, last accessed 2024/06/09