Classification Approach Description

Methodology *(Replicability; Scalability; Interpretability)*

Please provide a detailed description of the methodology used for identifying the classification of the online job advertisement. The description should contain (1) the data processing steps, (2) the methods and models used, (3) references to the scientific papers/sources that present the methods and models used, and (4) the time it took to process the data set and classify the job advertisements.

Bear in mind that the workflow will be also evaluated based on the criteria for the Reusability and Innovativity Awards.

This section will be evaluated for:

> *(1) the Replicability criterion: likeliness that the described approach can successfully reproduce the solution submitted by the team for the Accuracy award*
>
> *(2) the Scalability criterion: amount of modification required for the approach to apply to similar datasets on a potentially larger scale*
>
> *(3) the Interpretability criterion: the extent to which a human could understand and articulate the relationship between the approach's predictors and its outcome; how well the logical reasoning behind the model which is making the prediction is developed (whether it is mathematically and/or technically sound*

A RAG-based approach was employed to achieve optimal performance on the classification of job titles according to the ISCO-08 standard. RAG stands for 'Retrieval-Augmented Generation' and represents a powerful method to combine the strength of Large Language Models (LLMs) with domain-specific knowledge. More specifically, RAG starts by identifying information that may be relevant to the question one has and then feeds that information together with the question to a LLM, prompting it to formulate an answer based on the information that was provided.

Our RAG-based approach for the current competition consists of the following steps:

1. Collect information about each of the ISCO categories: in this step, we combined publicly available titles and descriptions of the official ISCO-08 documentation plus a list of ~34.000 ISCO-categorized example job titles in different languages. The latter file is from Professor Emeritus Peter Elias from the University of Warwick, who developed the automatic coding software CASCOT
2. Generate an embedding for each ISCO category: this was done by combining the ISCO category description with that category's example jobs into one text chunk and then sending that piece of text to OpenAI's `text-embedding-3-large` embedding model.
3. For each entry in the competition dataset, generate an embedding based on the entry's job title: for this step we also used OpenAI's `text-embedding-3-large` embedding model.
4. For each entry in the competition dataset, compare its embedding to the embeddings generated for the ISCO categories and retrieve the 10 'most similar' ISCO categories, i.e.: the 10 categories with the highest embedding-based similarity scores. These 10 categories will be the classification candidates for the given entry.

5. For each entry in the competition dataset, send the entry's job title and the list of 10 candidate ISCO categories to GPT and prompt it to select from those 10 the most appropriate one. For this step, we addressed OpenAI's *gpt-4o* model.

Based on a manual curation of the results, we decided to finetune our approach by implementing additional modifications for 2 specific types of entries:

1. For all entries having the label 'manager' in the job title, we generated a new embedding. This embedding was not based on the job title, but rather on the job information mentioned in the entry's job description. More specifically, we used OpenAI's *gpt-4o* model to extract an occupation from the job description, then used the `text-embedding-3-large` embedding model to generate an embedding based on that occupation. After that, we followed the same steps as described above, using the occupation-based embedding to retrieve 10 candidate ISCO categories, then prompting the *gpt-4o* model to select the best candidate.
2. For all entries for which the similarity scores between the entry's embedding and the embeddings of the 10 most similar candidates were low, i.e. < 0.35, we generated a new embedding in the same way as we did for the 'manager' jobs. The idea behind this is that the presence of such 'disputable' candidates may indicate that the job title is not of good quality. This might be overcome by instead using the occupational information in the job description for generating an embedding. Again, with this occupation-based embedding, we followed the same steps as described before, using it to get a new set of 10 candidate categories (with higher similarity scores) and then prompting *gpt-4o* to select the most appropriate out of these 10.

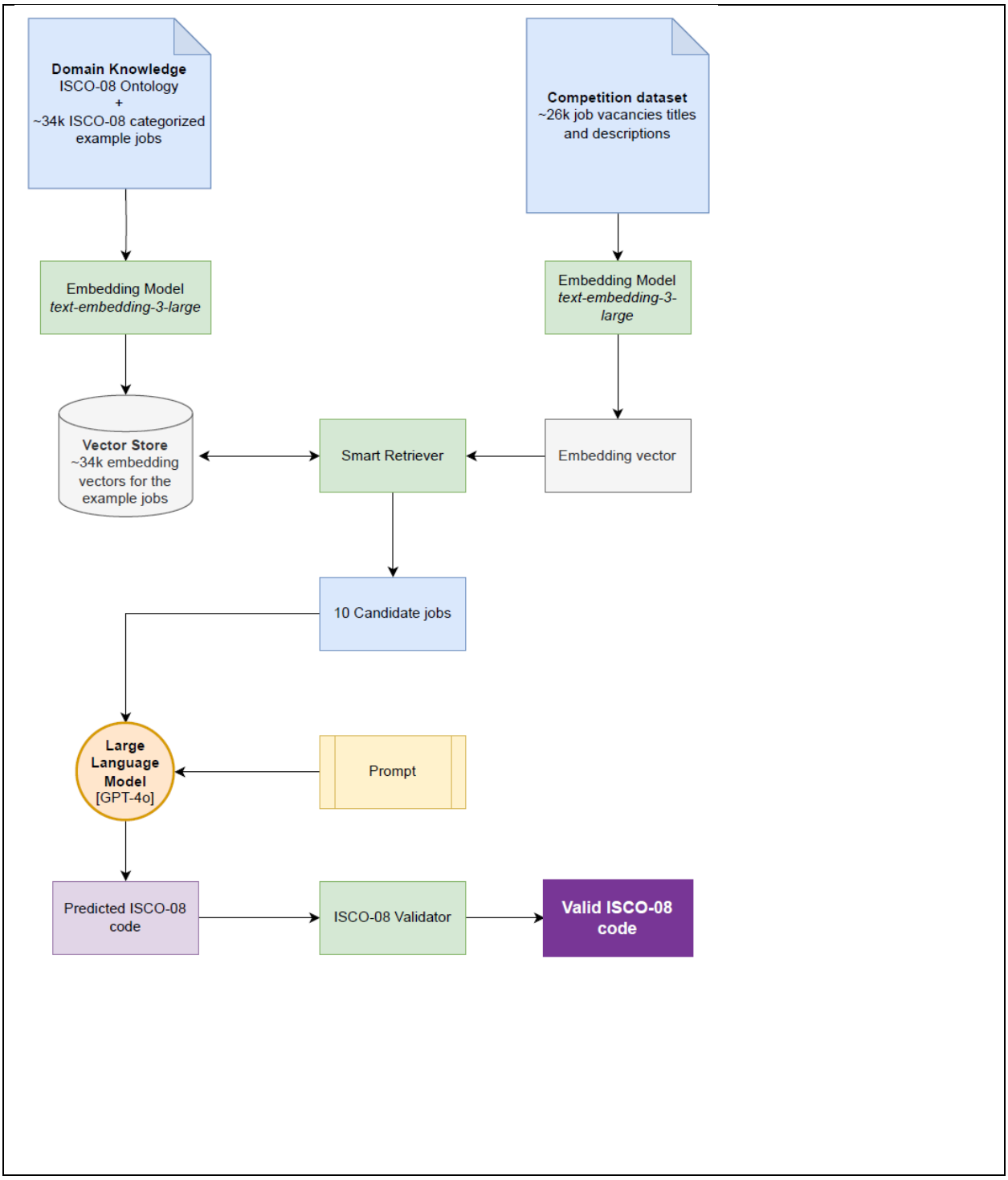
Architecture

Please provide a description of the architecture of your approach. A diagram of the architecture is considered of additional value. Indicate what modifications would be required to apply the approach to similar datasets on a larger scale.

This section will be evaluated for:

(1) the Architecture criterion: evaluated based on its modules, their cohesion and their configurability; an architecture which is modular and includes clear connections between modules or components receives a higher score

The architecture diagram represents the RAG-approach described in the Methodology section.



Hardware Specifications *(Replicability; Scalability; Interpretability)*

Please describe the hardware specifications of the machines that were used to run the methodology.

This section will be evaluated for:

(1) the Replicability criterion

(2) the Scalability criterion

(3) the Interpretability criterion

Machine 1: Azure Service for running Jupyter Notebook with Python scripts

CPUs	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz – 8 CPUs
GPUs	N.A.
TPUs	N.A.
Disk space	6.4 GB

Libraries *(Maintainability)*

Please provide the libraries used for approach, if any, as well as the links to these libraries, if available.

This section will be evaluated for:

(1) the Maintainability and openness criterion: use of libraries which are regularly maintained will yield higher scores. (Examples include pytorch, tensorflow, scikit-learn, pandas, numpy, etc.) The use of libraries which are openly available will yield higher scores.

The following libraries have been used in our approach:

openai: https://pypi.org/project/openai/0.26.5/
openpyxl: https://pypi.org/project/openpyxl/
scikit-learn: https://pypi.org/project/scikit-learn/
pandas: https://pypi.org/project/pandas/
numpy: https://pypi.org/project/numpy/
re: built-in
json: built-in

Open license *(Maintainability)*

Please provide the open license of the provided code, if any.

This section will be evaluated for:

(1) the Maintainability and openness criterion: whether the approach is open and under an open license

Similarities/differences to State-of-the-Art techniques *(Originality)*

Please provide a list of similarities and differences between the used methodology and to the state-of-the-art techniques.

This section will be evaluated for:

> *(1) the Originality of the approach criterion: compare the approach used to the state-of-the-art; the extent to which the submission represents an improvement over these pre-existing approaches*

Our approach is largely distinct from approaches used by existing commercial software and available automatic job classification models. For instance, the CASCOT software developed by the University of Warwick utilises a number of Boolean operators and string similarity matching algorithm in Java to code free-text entries to ISCO and other classifications. In more modern automatic job coding models, text input goes through natural language processing (e.g. tokenization, lemmatization, embedding) before training in machine learning models (e.g. Random forest, XGBoost). More details on existing automatic job coding techniques are described in Wan et al 2023 and Schierholz 2019.

In recent years attempts have been made using LLMs to extract and code jobs from job postings. An example of this is described in Li et al 2023. In July, the Data Science Campus from the UK ONS published an experimental pipeline for text industry and occupation classification. Our team took the pipeline as an example and further developed and refined it for this competition.

Contribution to scientific field *(Future orientation)*

Please describe how your submission contributed to the scientific field, what impact it could have and what could potentially be future work to improve the solution.

This section will be evaluated for:

> *(1) the Future orientation and impact criterion: the potential effect of the approach used will be evaluated; this includes the scale of impact it has on the problem of the classification of job advertisements; the impact will be evaluated based on potential efficiency improvements and cost reductions.*

Our approach with OpenAI embedding and classification with RAG contributes distinct advantages over pre-existing approaches:

1) text pre-processing and embedding are directly handled and created by LLM;
2) OpenAI embedding is not specific to a language, resulting in the ability to label text in any input language;
3) RAG approach requires no training data that was previously coded.

In terms of future research and impact on job classification, there is a need for datasets with high quality job descriptions with high quality job classification assignments. These datasets are important for training of machine learning models and testing/validation of all models. Creation of these datasets, however, presents major technical and cost challenges. Within our submission there are elements that may help improve job description data quality and manual data coding.

Our approach to use LLMs to generate new job titles when the original job title is ambiguous may be used to potentially improve data quality for existing files. This could also be implemented in real-time in, for instance, a job-posting portal to immediately code a job while it is being posted, giving the vacancy poster an opportunity to add or correct job description immediately.

In addition to supporting our generator model, embeddings comparison results from our retriever model may be used to aid manual coding by presenting the top (e.g. 10 or 5) job codes that may be appropriate for a job description entry. This may facilitate the labourious and costly manual coding process and increase the quality of job code assignments.

Lessons Learned *(Future orientation)*

Please state any lessons learned during the competition.

This section will be evaluated for:

> *(1) the Future orientation and impact criterion: what were the lessons learnt during the competition, and what could potentially be future work to improve the solution.*

One of the lessons we learned during this competition is that for a relatively simple classification problem, there are many different possible approaches – each having its own pros and cons. Additionally, each element of the pipeline has different options. For example, there are various choices in embedding models, and the choice of model may determine whether a translation step is needed for the multilingual input text. For our final submission, we chose an embedding model from OpenAI that works with different languages, allowing our workflow to directly accept multilingual input.

Having multiple possible submissions encouraged us to test different approaches and adapt according to the returned performance of each one. In addition to the RAG approach submitted in the final submission, we also tried machine learning (ML) models trained on our own data coded to ISCO-08. The ML models' performance was lower than that of the RAG approach.

Having domain knowledge within the team also played an important role when making decisions about different approaches. For instance, coding of managerial jobs is often challenging, so we have added a step to try to obtain more information from job descriptions if an entry includes "manager" in the job title. Our overall familiarity and experience with job occupation ontologies like ISCO-08 allowed us to use available resources such as the sample job list we obtained from another collaboration for training of our RAG model.

As in many datasets with job information and coding, the type and quality of the data has a huge impact on model performance. Entries with lower quality descriptions (e.g. vague, incomplete) are typically challenging. Therefore, we included a step in our RAG to identify and obtain more detail on entries with low embedding match performance.

Large Language Models are vulnerable to hallucinations, making it essential to have a robust system in place to validate the model outputs and to mitigate invalid outputs.

Short description of the Team – area of expertise

Please provide a description of the team, your area of expertise and contact information.

TNO

- Calvin Ge, PhD, Occupational exposures and epidemiology, Senior Scientist at TNO
- Gino Kalkman, PhD, NLP in the biomedical domain, Team Lead and Scientist at TNO
- Xavier Pinho, MSc Biomedical Engineering, Data Scientist at TNO
- Sadegh Shahmohammadi, PhD, Senior Scientist and knowledge lead of TNO Healthy Living AI-Lab, sadegh.shahmohammadi@tno.nl