Applied Food Research 5 (2025) 100679

Contents lists available at ScienceDirect

Applied Food Research

journal homepage: www.elsevier.com/locate/afres

Extracting chemical food safety hazards from the scientific literature automatically using large language models

Neris Özen, Wenjuan Mu, Esther D. van Asselt, Leonieke M. van den Bulk *

Wageningen Food Safety Research, Wageningen University & Research, Wageningen, The Netherlands

ARTICLE INFO

Keywords:
Chemical contamination
Food safety
Information extraction
Prompt engineering
Natural language processing
Artificial intelligence

ABSTRACT

The number of scientific articles published in the domain of food safety has consistently been increasing over the last few decades. It has therefore become unfeasible for food safety experts to read all relevant literature related to food safety and the occurrence of hazards in the food chain. However, it is important that food safety experts are aware of the newest findings and can access this information in an easy and concise way. In this study, an approach is presented to automate the extraction of chemical hazards from the scientific literature through large language models. The large language model was used out-of-the-box and applied on scientific abstracts; no extra training of the models or a large computing cluster was required. Three different styles of prompting the model were tested to assess which was the most optimal for the task at hand. The prompts were optimized with two validation foods (leafy greens and shellfish) and the final performance of the best prompt was evaluated using three test foods (dairy, maize and salmon). The specific wording of the prompt was found to have a considerable effect on the results. A prompt breaking the task down into smaller steps performed best overall. This prompt reached an average accuracy of 93 % and contained many chemical contaminants already included in food monitoring programs, validating the successful retrieval of relevant hazards for the food safety domain. The results showcase how valuable large language models can be for the task of automatic information extraction from the scientific literature.

1. Introduction

It is crucial that food safety experts are aware of the newest research concerning the occurrence of hazards in the food chain to ensure our food can be as safe as possible. Because of the growing interest in food safety over the last decades, the number of scientific articles related to the domain has consistently been increasing (Luo et al., 2022). However, this has introduced a challenge: Hundreds of publications are currently written on food safety each year, and it has become unfeasible for food safety experts to read everything relevant for their field. This makes it difficult to keep track of all new findings and potential risks related to the food safety domain. To increase awareness of new hazards so that timely action can be taken to safeguard our food, it is important that information reaches the food safety experts in an easy and concise way. Automating the extraction of relevant information from scientific literature can be a promising asset to save researchers' valuable time.

Natural language processing (NLP) can be adopted for this task, which has seen a rise in development in the past few years. NLP is the branch of Artificial Intelligence (AI) that focuses on understanding text and is used for text translation (Luong et al., 2015; Wu et al., 2016; Zhu et al., 2023), summarization (Y. Liu & Lapata, 2019; Zhang et al., 2020), part-of-speech tagging (Chiche & Yitagesu, 2022; Wang et al., 2015), text classification (Kowsari et al., 2019; Minaee et al., 2021), and text generation (Brown et al., 2020; Devlin et al., 2019; Lewis et al., 2019). NLP can also be used for information extraction, where relevant information from unstructured text is collected automatically without the need for a domain expert to manually go through the text (Hong et al., 2021; Zaman et al., 2020). Early research in NLP focused on the application of rule-based methods or n-gram models (K. Min & Wilson, 2006; Nadkarni et al., 2011), but these approaches have long been overtaken by better performing neural network models (Khurana et al., 2023; B. Min et al., 2023). Moreover, in recent years there has been a shift towards using pre-trained neural network models instead of training these models from the ground up on task-specific datasets (Howard & Ruder, 2018; Peters et al., 2018; Vaswani et al., 2017). These pre-trained models have been trained on enormous amounts of texts with the goal of capturing the underlying patterns and structures that are present in human language. In contrast to task-specific datasets, these texts do not

* Corresponding author.
E-mail address: leonieke.vandenbulk@wur.nl (L.M. van den Bulk).

https://doi.org/10.1016/j.afres.2024.100679
Received 21 May 2024; Received in revised form 16 December 2024; Accepted 26 December 2024
Available online 27 December 2024

have to be labelled, and can instead be trained using self-supervised learning (X. Liu et al., 2021). This makes pre-training much less labor and cost intensive, allowing the possibility to train these models on vast amounts of data. When large neural network models are trained on big datasets and use advanced neural network architectures, like the Transformer architecture, they become capable of both understanding and generating realistic and human-like text (Vaswani et al., 2017). These models are referred to as large language models (LLMs) and they have quickly acquired popularity in the AI domain for the past few years, due to their capability of performing various tasks at state-of-the-art levels (Khurana et al., 2023; B. Min et al., 2023). LLMs are powerful models because of their ability to represent segments of text as numerical vectors that reflect their semantic and syntactic meaning (called word embeddings), and their pre-training process (Devlin et al., 2019; Radford et al., 2018). LLMs are pre-trained by learning to predict the next word in a text in a self-supervised setting, forcing the model to master the correct syntax and understanding of the predicted words. Subsequently, LLMs are often trained on an instruction-response dataset to guide the model to produce more relevant responses to a user's input (Wei et al., 2022).

After training, LLMs can either be finetuned on a task-specific dataset to specifically be calibrated to perform a single task or they can be used by giving them instructions in natural language without any additional training, also called prompting. Radford et al. (2019) and Brown et al. (2020) pioneered in this domain by demonstrating that scaling up language models could eliminate the need to further finetune them for specific tasks with labeled datasets. Their LLMs were some of the first models which only required a textual instruction to carry out a particular linguistic task, at a performance comparable to that of state-of-the-art finetuned models. When ChatGPT was released, Qin et al. (2023) explored the performance of ChatGPT across a wide range of tasks to investigate if it was a "general-purpose" model for NLP. They found that although ChatGPT was not yet a true generalist model, it still showed remarkable capabilities and obtained good scores across the full range of tasks, which included arithmetic reasoning, question answering, summarization and information extraction. The specific prompt given to the model, however, is very important since the quality of the prompt can have a major effect on the quality of the answer (P. Liu et al., 2023). The formulation of optimal prompts has evolved into an emerging scientific field, known as prompt engineering. Various strategies can be applied when generating prompts to enhance the performance of the LLM, such as setting clear goals for the task, providing distinct subtasks, giving examples of expected answers and specifically asking for the reasoning behind the answer (P. Liu et al., 2023; Santu & Feng, 2023). These strategies help to establish a better context for the LLM, because the prompt is made less ambiguous and the intent behind the instruction is formulated more precisely. This guides the LLM to predict an output that is contextually close to similar texts it has seen during training and therefore generate a more relevant response.

Automatic information extraction can be a powerful tool to obtain knowledge and insights from any domain, drastically reducing the need for manual work. Since LLMs are trained to understand the context behind a text, they are currently the start-of-the-art when it comes to information extraction (Xu et al., 2023). LLMs have been applied widely across different scientific domains for the purpose of information extraction, and have especially been developed within the biomedical domain. LLMs have already been successfully applied in that domain for the case of medical information extraction from clinical notes (Agrawal et al., 2022), finding adverse effects of drugs in medical reports (Gu et al., 2023), summarizing the effect of food on the absorption of drugs from drug application reviews (Shi et al., 2023), and identifying the number of participants from scientific publications (Paroiu et al., 2023). There has also been a growing interest in applying LLMs in solving the challenges of the food domain. They have been applied for the generation of healthy diet suggestions for people with food allergies (Niszczota & Rybicka, 2023), the extraction of food dish names from restaurant reviews and social media posts (Lin et al., 2023), and the creation of a question-answer system of food testing data (Qi et al., 2023).

In this study, we aimed to assess whether LLMs can automatically extract chemical food safety hazards from the scientific literature through prompting without any additional training of the models, making the approach easy to implement for non-AI experts. We used scientific abstracts as our input for the LLM, as these are readily available in literature databases, contain the most important findings and offer a concise text that can be processed well by language models. This approach can showcase how valuable large language models can be for the task of automatic information extraction from the scientific literature in the field of food safety.

2. Materials and methods

An open-source LLM was applied for the extraction of chemical contaminants from scientific abstracts by prompting the model with textual instructions, and without introducing any extra training for the specific task at hand. The pipeline presented in Fig. 1 shows the performed steps. To prompt the LLM with relevant scientific abstracts and correctly identify the chemical names in the responses of the LLM, data collection and preprocessing steps were required for both the set of abstracts and a chemical dictionary. After filtering the collected abstracts related to a specific food, the LLM is prompted to extract the mentioned chemical contaminants in the abstracts. The final steps aim to parse the responses of the LLM to generate the final list of chemical hazards for a food item and evaluate their correctness. Evaluation was performed for both a validation and test procedure. A validation procedure was used to identify the prompt that elicited the best performance from the LLM using leafy greens and shellfish as validation foods. This was followed by a test procedure to measure the definitive performance of the best-performing prompt using dairy, maize and salmon as test foods. The detailed procedures related to the steps in the pipeline are described in the subsections below.

2.1. Abstracts collection and preprocessing

The open-access literature repository Europe PMC was used to collect scientific abstracts that were of interest for this study. Relevant abstracts

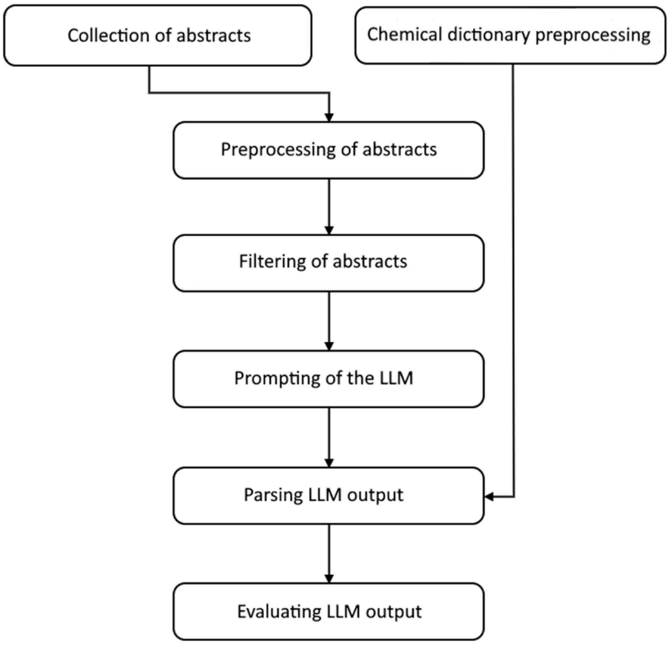


Fig. 1. The pipeline showing the performed steps to extract chemical hazards from scientific abstracts using an LLM.

should include chemical contamination in food with a possible negative effect on human health. A search query was created based on two sets of search terms and each abstract was required to contain at least a word or phrase from each set of terms in its title, abstract or keywords. The first set of terms focused on words/phrases indicating presence of chemical hazards, whereas the second set of terms contained words/phrases focusing on an impact on public health. The search query was based on those used in systematic literature reviews previously performed to identify chemical contaminants in food (Banach et al., 2019; Kluche et al., 2020). The search queries used in these reviews had been meticulously created to capture relevant publications on chemical hazards. They were evaluated by food safety experts for their accuracy and therefore deemed fit for our use case. See Appendix A for the final search query that was used in this study.

To obtain the abstracts and corresponding metadata, a request was made to the REST API service of Europe PMC with the created search query. All abstracts with a publication date up to April 2nd, 2023, were stored. The DOI, title and publication year of the original article were stored along with the abstracts.

The collected abstracts were cleaned by removing HTML tags, excessive spaces and copyright declarations. Empty or duplicate abstracts were excluded, in addition to the abstracts correcting, amending or withdrawing previously published research.

2.2. *Chemical dictionary preprocessing*

Even though the LLM is asked to only retrieve chemical hazards, there is a possibility that it returns a different entity (e.g. a microbiological or physical hazard). Furthermore, chemicals often have many different synonyms that can be used. To link these different synonyms across the abstracts and to make sure that our output only contains chemical hazards, we applied a chemical dictionary to identify correct names of chemicals in the responses of the LLM. The dictionary chosen for this study was ChEBI (Chemical Entities of Biological Interest). ChEBI includes a controlled vocabulary where for each entry, a chemical name is provided alongside its unique ChEBI identifier, IUPAC name and its synonyms (Hastings et al., 2016). The dictionary contains over 170,000 unique chemicals. With chemical names and their synonyms provided alongside a persistent ChEBI identifier, different names or different writing conventions of the same chemical seen in the responses of LLM can be matched to a single identifier (e.g. both ‘Cd’ and ‘cadmium’ match to the same identifier CHEBI:28628).

The dictionary was preprocessed to provide the best applicability to our case. Some entries were removed from consideration because their names are too generic (e.g. "molecule", "solvent" or "vitamins") or don’t represent chemical entities (e.g. "application", "voltage" or "alpha"). In addition, extra synonyms using different writing conventions for existing chemical names were added to the dictionary. This step was executed on chemical names which contain numbers alongside letters. For example, in case a chemical belongs to a specific isotope of an element (e.g. polonium-210) or a specific type of compound (e.g. aflatoxin b1), more synonyms were created by swapping the number and word components or adding or removing spaces or hyphens (e.g. generating 210-polonium or aflatoxin b-1). The dictionary is also extended to contain all plural forms of the chemicals. The final preprocessed dictionary contains approximately 1.4 million chemical names.

2.3. *Abstract filtering and LLM prompting*

Abstracts were filtered to only retain abstracts relevant to a selected food. The collected abstracts from Europe PMC were filtered based on the presence of keywords defining the respective foods (including common synonyms) from the validation and test procedures. For the foods used for validation, abstracts for leafy greens were required to include “leafy green”, “leafy greens”, “leafy vegetable” or “leafy vegetables”, whereas for shellfish, abstracts were required to include “shellfish”. For the foods used to test the final performance, in the case of dairy “dairy” needed to be present in abstracts, for maize “maize” or “corn” was required and finally for salmon “salmon” needed to be present. Note that the keywords representing the foods do not include their subcategories (e.g. “spinach” or “milk”) as the goal was to test the performance for both individual and categories of food (i.e. “leafy greens” and “dairy”).

Once abstracts for the desired food were determined, an open-source LLM was used for information extraction. To ensure the reproducibility of our work, remove the need for a remote cluster with substantial computational power and with privacy concerns of LLM platforms in mind, we decided to employ an open-source LLM that is relatively small in terms of its parameters and can be run locally on a personal computer with a good GPU. The “Nous-Hermes-13B-GPTQ" LLM model was selected based on these criteria. The model rivaled, at the moment of model selection, performance of much bigger open-source models like Vicuna 30B, Llama 30B and Falcon 40B in several LLM benchmark tasks (Chiang et al., 2023; Touvron et al., 2023; Almazrouei et al., 2023). It was also able to achieve performance close to that of GPT-3.5. The model was downloaded from the Hugging Face platform[1], a platform where AI models are openly shared. Nous-Hermes LLM models are made available by Nous Research, a company focused on open-sourcing LLM models that are small enough to be run locally on people’s own computers. The Nous-Hermes-13B-GPTQ model capitalizes on the Llama architecture by taking the pre-trained Llama-13B model as its base and finetuning it on over 300,000 instructions generated with synthetic GPT-4 outputs (Touvron et al., 2023). The Llama-13B model was pre-trained on 4.7 TB of publicly available text data, which includes texts from Wikipedia, books, scientific literature and a large number of web pages.

There is a wide range of hyperparameters in LLMs that can be adjusted to change the way it responds. This study aimed to obtain outputs as reproducible and accurate as possible, while keeping the computational and memory burden low. Therefore, the selection of hyperparameters focused on setting appropriate values for ‘do_sample’, ‘num_beams’ and ‘repetition_penalty’ while keeping other hyperparameters at their default values. The hyperparameter ‘do_sample’ can facilitate creativity but comes at the expense of reproducibility, as it allows the LLM to randomly sample words when generating a response from a probability distribution. Setting ‘do_sample’ to False forces the LLM to always opt for the words that are most likely to follow the existing sequence, generating a consistent response. The hyperparameter ‘num_beams’ specifies the number of most probable words that can be chosen at each time step and controls the number of possible different answers that the LLM needs to evaluate. Values over 1 for ‘num_beans’ introduce a significant computation and memory load on the LLM, therefore a value of 1 was chosen. Lastly, the hyperparameter ‘repetition_penalty’ penalizes the repetition of words in the response. Since chemical hazards should be able to be repeated for multiple food items in an abstract, this hyperparameter was set to the lowest possible value of 1.0.The LLM was prompted by providing each of the filtered abstracts one by one, with the instruction to extract the chemical hazards and the food(s) they are present in from that abstract. Since the performance of an LLM in a task can vary significantly depending on the exact prompt, three different styles of prompts were drafted and assessed for the accuracy of their outputs on the validation foods. Specific wording in the different prompts were incrementally updated according to their performance. The three styles of prompts can be characterized as follows:

1. “Simple prompt”: This is written fully in natural language. The task the LLM is expected to execute and the required format of the LLM

[1] https://huggingface.co/TheBloke/Nous-Hermes-13B-GPTQ

output are expressed in two sentences. In light of earlier experiments with the LLM and the observation of its pitfalls, another paragraph warning the LLM against potential pitfalls was added. At the very end, the abstract of interest is provided in triple backticks.

2. “Step-by-step prompt”: As recommended by Fulford & Ng (2023), the task is broken down into 5 smaller tasks and each smaller task is expressed in enumerated steps to facilitate a reasoning process and to prevent the LLM from giving rash answers. With the same motivation, the model is asked to print the outcome of each step in a specific format. To make sure that this prompt does not have a disadvantage compared to the “simple prompt” concerning the pitfalls the LLM is prone to, the same paragraph of warning is provided in this prompt as well. Like in the “simple prompt”, the abstract is provided in triple backticks for the model to execute the task on.
3. “Pseudocode prompt”: This style of prompting is inspired by the work of Mishra et al. (2023), who defend that pseudocode instructions are less susceptible to ambiguities and find that pseudocode instructions beat their counterparts in natural language over a variety of NLP tasks. The task is introduced in the format of a Python function where the smaller tasks constituting the task are now expressed as a line of code in the function, accompanied by natural language description of the smaller task as comment. Docstrings are used to repeat the natural language descriptions of all smaller tasks and include the warnings against the common pitfalls. After the Python function is defined, it is called with the abstract as input.

Full textual instructions for each prompt are provided in Figs. 2-4.

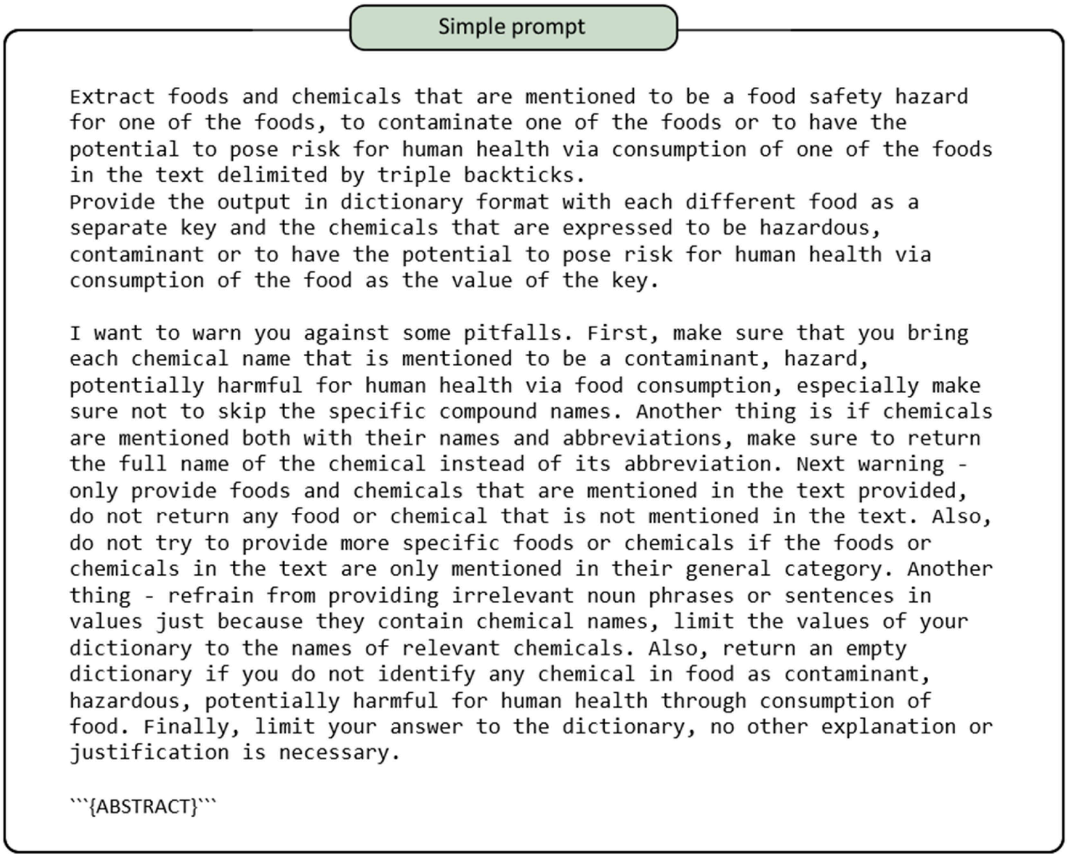
2.4. LLM output parsing and evaluation

The LLM was instructed to extract the chemical hazards and their corresponding foods from each abstract in a dictionary format in each prompt, where a food item is linked to a list of its hazards (e.g. {‘Chinese cabbage’: [‘cadmium’, ‘chromium’], ‘wheat’: [‘deoxynivalenol’, ‘arsenic’], ‘shellfish’: [‘saxitoxin’]}). If there are no chemical hazards present in an abstract, the model was asked to output an empty dictionary. Chemical hazards were only accepted as answers if the corresponding food item extracted by the LLM contained one of the keywords used to filter the abstracts as specified in Section 2.3.

The responses of the LLM were mostly consistent in providing the requested output format as shown above, however, initial evaluation showed that they sometimes deviated from the required format (e.g. the hazards were not provided in a list or the hazards were embedded inside another dictionary). The responses in common deviant formats did, however, tend to contain correct chemical hazards. Therefore, when parsing the responses, the most common consistent deviations from the desired output format were accommodated with a different parsing pattern if there was still a clear link to which food the chemicals belonged to. Other deviant formats were discarded from the results.

The parsed output was automatically compared against the preprocessed chemical names and synonyms from ChEBI. This helped with filtering out the non-chemical names in the list, as the LLM returned some adjectives (e.g. “hazardous”) or some microbiological hazards in a number of responses. In addition, some of the responses can contain an abbreviation of a chemical instead of its full name. In this situation, the abbreviations were replaced by the full chemical name by automatically trying to trace back the abbreviation in the original abstract. The found chemical names were subsequently matched to their ChEBI identifiers. This way, the same chemical extracted in varying writing conventions from different responses were not counted double.

A list of ChEBI identifiers of chemicals identified as hazardous and the corresponding abstract DOI’s was created as the final result to facilitate the evaluation procedure. Evaluation of the results for the validation foods involved manually checking the DOI’s provided for



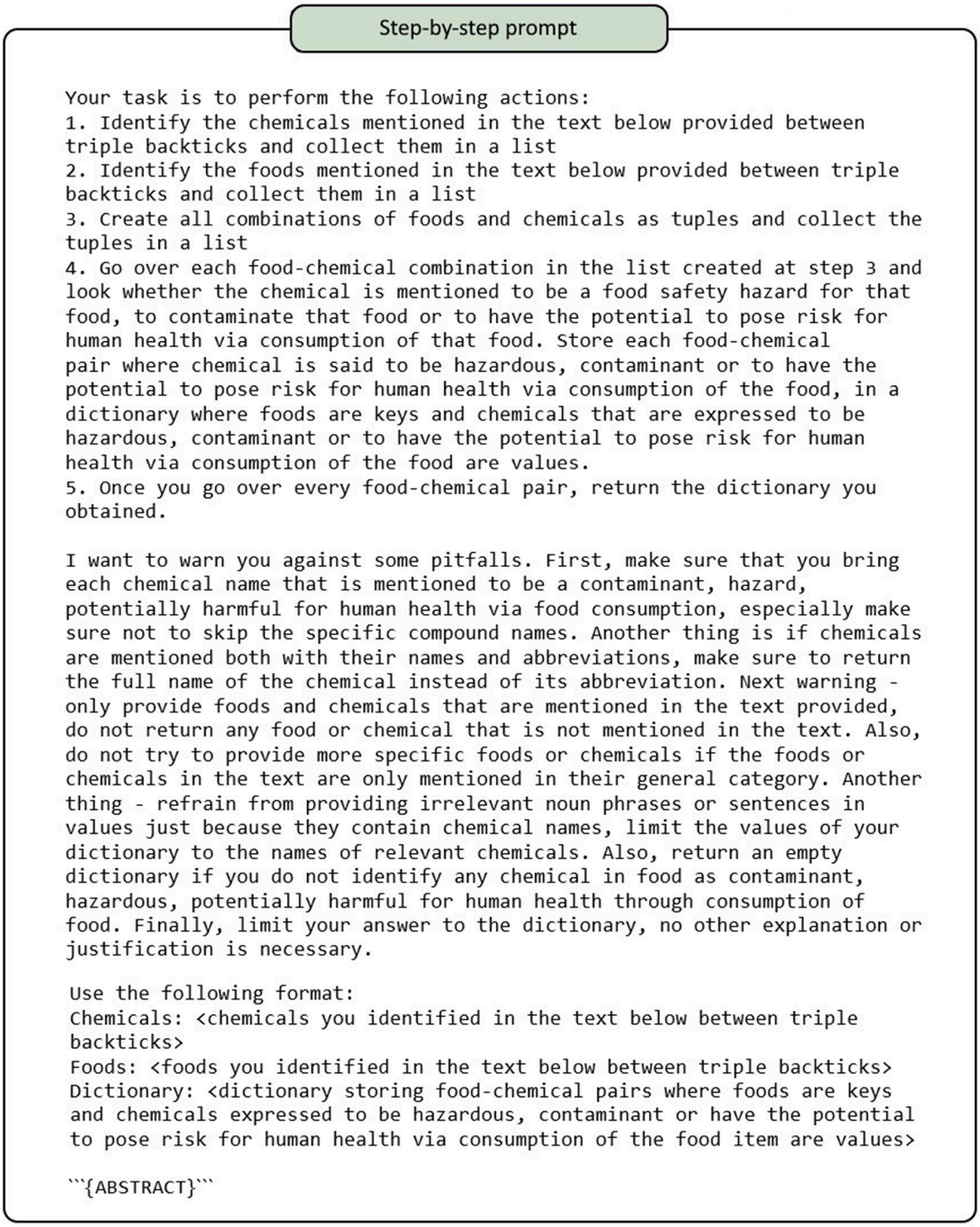
Simple prompt

~~~
Extract foods and chemicals that are mentioned to be a food safety hazard
for one of the foods, to contaminate one of the foods or to have the
potential to pose risk for human health via consumption of one of the foods
in the text delimited by triple backticks.
Provide the output in dictionary format with each different food as a
separate key and the chemicals that are expressed to be hazardous,
contaminant or to have the potential to pose risk for human health via
consumption of the food as the value of the key.

I want to warn you against some pitfalls. First, make sure that you bring
each chemical name that is mentioned to be a contaminant, hazard,
potentially harmful for human health via food consumption, especially make
sure not to skip the specific compound names. Another thing is if chemicals
are mentioned both with their names and abbreviations, make sure to return
the full name of the chemical instead of its abbreviation. Next warning -
only provide foods and chemicals that are mentioned in the text provided,
do not return any food or chemical that is not mentioned in the text. Also,
do not try to provide more specific foods or chemicals if the foods or
chemicals in the text are only mentioned in their general category. Another
thing - refrain from providing irrelevant noun phrases or sentences in
values just because they contain chemical names, limit the values of your
dictionary to the names of relevant chemicals. Also, return an empty
dictionary if you do not identify any chemical in food as contaminant,
hazardous, potentially harmful for human health through consumption of
food. Finally, limit your answer to the dictionary, no other explanation or
justification is necessary.

```{ABSTRACT}```
~~~

Fig. 2. The full prompt text for the simple prompt.



Step-by-step prompt

~~~
Your task is to perform the following actions:
1. Identify the chemicals mentioned in the text below provided between
triple backticks and collect them in a list
2. Identify the foods mentioned in the text below provided between triple
backticks and collect them in a list
3. Create all combinations of foods and chemicals as tuples and collect the
tuples in a list
4. Go over each food-chemical combination in the list created at step 3 and
look whether the chemical is mentioned to be a food safety hazard for that
food, to contaminate that food or to have the potential to pose risk for
human health via consumption of that food. Store each food-chemical
pair where chemical is said to be hazardous, contaminant or to have the
potential to pose risk for human health via consumption of the food, in a
dictionary where foods are keys and chemicals that are expressed to be
hazardous, contaminant or to have the potential to pose risk for human
health via consumption of the food are values.
5. Once you go over every food-chemical pair, return the dictionary you
obtained.

I want to warn you against some pitfalls. First, make sure that you bring
each chemical name that is mentioned to be a contaminant, hazard,
potentially harmful for human health via food consumption, especially make
sure not to skip the specific compound names. Another thing is if chemicals
are mentioned both with their names and abbreviations, make sure to return
the full name of the chemical instead of its abbreviation. Next warning -
only provide foods and chemicals that are mentioned in the text provided,
do not return any food or chemical that is not mentioned in the text. Also,
do not try to provide more specific foods or chemicals if the foods or
chemicals in the text are only mentioned in their general category. Another
thing - refrain from providing irrelevant noun phrases or sentences in
values just because they contain chemical names, limit the values of your
dictionary to the names of relevant chemicals. Also, return an empty
dictionary if you do not identify any chemical in food as contaminant,
hazardous, potentially harmful for human health through consumption of
food. Finally, limit your answer to the dictionary, no other explanation or
justification is necessary.

Use the following format:
Chemicals: <chemicals you identified in the text below between triple
backticks>
Foods: <foods you identified in the text below between triple backticks>
Dictionary: <dictionary storing food-chemical pairs where foods are keys
and chemicals expressed to be hazardous, contaminant or have the potential
to pose risk for human health via consumption of the food item are values>

```{ABSTRACT}```
~~~

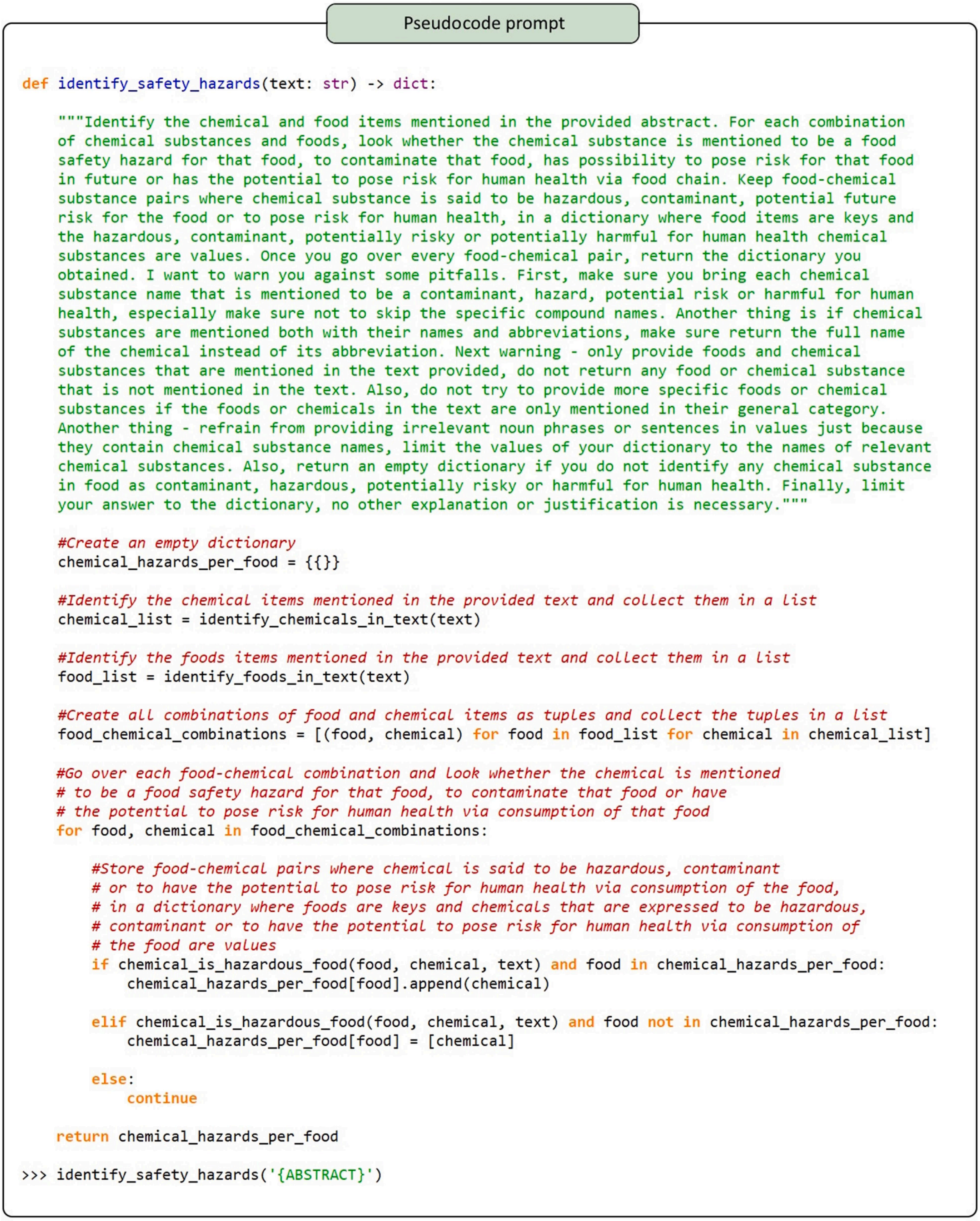
Fig. 3. The full prompt text for the step-by-step prompt.

each chemical entry by two of the authors. It was checked that the abstracts indeed expressed that that chemical, or the wider contaminant group, contaminates that food. At the end of the validation stage, the best performing prompt was chosen for ultimate assessment in the test cases. For each test food, the results returned by the LLM were evaluated by a domain expert following the same approach as in the validation stage. These evaluations were checked by two of the authors for consistency. In both procedures, the performance was measured by the number of correct chemical hazards extracted from scientific abstracts for a specific food.

3. Results

In total 101,727 unique abstracts were retrieved from Europe PMC with the aim to identify chemical contamination in food with a possible negative effect on human health. These abstracts were filtered to only retain abstracts related to the selected validation and test foods. The total number of abstracts relevant for the validation foods were 411 for leafy greens and 1235 for shellfish. For the test foods 1403, 1318 and 353 abstracts were filtered as relevant for dairy, maize and salmon respectively. Fig. 5 shows the number of abstracts per year for each of the foods, clearly showing an upward trend in the number of published abstracts on this topic. Note that the year 2023 was omitted due to abstracts only having been collected until April.

The responses from the three different prompts were analyzed and compared across abstracts. Foremost, as demonstrated in Fig. 6, responses to the prompts differed significantly in their verbosity. While the pseudocode prompt tended to elicit only the desired dictionary as output, the step-by-step prompt provided outputs for the intermediary steps of the task before providing the final answer as requested.



Pseudocode prompt

```
def identify_safety_hazards(text: str) -> dict:

    """Identify the chemical and food items mentioned in the provided abstract. For each combination
    of chemical substances and foods, look whether the chemical substance is mentioned to be a food
    safety hazard for that food, to contaminate that food, has possibility to pose risk for that food
    in future or has the potential to pose risk for human health via food chain. Keep food-chemical
    substance pairs where chemical substance is said to be hazardous, contaminant, potential future
    risk for the food or to pose risk for human health, in a dictionary where food items are keys and
    the hazardous, contaminant, potentially risky or potentially harmful for human health chemical
    substances are values. Once you go over every food-chemical pair, return the dictionary you
    obtained. I want to warn you against some pitfalls. First, make sure you bring each chemical
    substance name that is mentioned to be a contaminant, hazard, potential risk or harmful for human
    health, especially make sure not to skip the specific compound names. Another thing is if chemical
    substances are mentioned both with their names and abbreviations, make sure return the full name
    of the chemical instead of its abbreviation. Next warning - only provide foods and chemical
    substances that are mentioned in the text provided, do not return any food or chemical substance
    that is not mentioned in the text. Also, do not try to provide more specific foods or chemical
    substances if the foods or chemicals in the text are only mentioned in their general category.
    Another thing - refrain from providing irrelevant noun phrases or sentences in values just because
    they contain chemical substance names, limit the values of your dictionary to the names of relevant
    chemical substances. Also, return an empty dictionary if you do not identify any chemical substance
    in food as contaminant, hazardous, potentially risky or harmful for human health. Finally, limit
    your answer to the dictionary, no other explanation or justification is necessary."""

    #Create an empty dictionary
    chemical_hazards_per_food = {{}}

    #Identify the chemical items mentioned in the provided text and collect them in a list
    chemical_list = identify_chemicals_in_text(text)

    #Identify the foods items mentioned in the provided text and collect them in a list
    food_list = identify_foods_in_text(text)

    #Create all combinations of food and chemical items as tuples and collect the tuples in a list
    food_chemical_combinations = [(food, chemical) for food in food_list for chemical in chemical_list]

    #Go over each food-chemical combination and look whether the chemical is mentioned
    # to be a food safety hazard for that food, to contaminate that food or have
    # the potential to pose risk for human health via consumption of that food
    for food, chemical in food_chemical_combinations:

        #Store food-chemical pairs where chemical is said to be hazardous, contaminant
        # or to have the potential to pose risk for human health via consumption of the food,
        # in a dictionary where foods are keys and chemicals that are expressed to be hazardous,
        # contaminant or to have the potential to pose risk for human health via consumption of
        # the food are values
        if chemical_is_hazardous_food(food, chemical, text) and food in chemical_hazards_per_food:
            chemical_hazards_per_food[food].append(chemical)

        elif chemical_is_hazardous_food(food, chemical, text) and food not in chemical_hazards_per_food:
            chemical_hazards_per_food[food] = [chemical]

        else:
            continue

    return chemical_hazards_per_food

>>> identify_safety_hazards('{ABSTRACT}')
```

Fig. 4. The full prompt text for the pseudocode prompt.

Responses to the simple prompt were even more elaborate, repeating a part of the task description provided in the prompt. In addition, responses to the step-by-step and simple prompts were more likely to contain a deviation in the output format from the desired one, particularly in the content of the dictionaries, where output could be plain text or nested dictionaries instead of lists of text. The three prompt styles also returned diverse responses when no chemical hazard was identified in the abstract by the LLM. The simple approach always returned a filled dictionary with foods and (wrong) hazards, however, it would end with a note stating it was returning an empty dictionary because there was no hazard found. The step-by-step approach either responded successfully with an empty dictionary or with the text “None”. The pseudocode

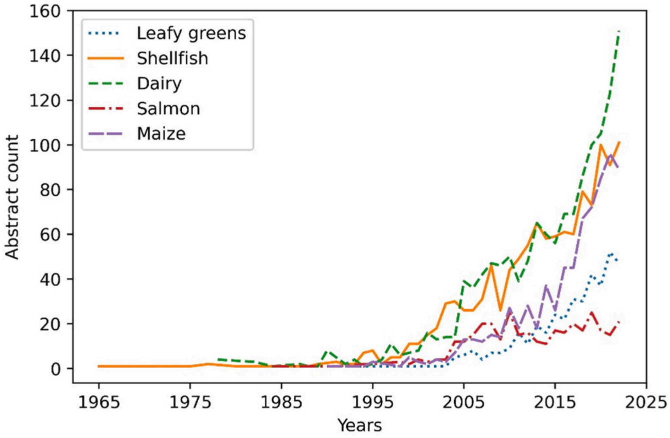


Fig. 5. Counts of abstracts per year retrieved from Europe PMC for each of the validation and test foods.

prompt elicited a response in natural language expressing its inability to identify a chemical hazard. Overall, the step-by-step approach showcased a stronger performance in differentiating between chemical and microbial hazards. Especially in abstracts only discussing microbial hazards the responses to the simple and pseudocode prompts tended to include the microbial entities, whereas the step-by-step prompt usually refrained from providing the microbials in the response. Furthermore, the step-by-step approach had a greater tendency to provide the full names of chemical hazards (e.g. “cadmium” instead of “Cd”) compared to the other two prompts, irrespective of whether the abstract in question contained the full name or not. Importantly, identification of food entities differed across responses to the prompts. Responses to the simple prompt contained broader categories of foods such as “leafy greens” or “shellfish” more often, whereas the other two were more prone to returning specific foods such as “spinach” or “mussel”. Generally, all prompts tended to adopt a more restricted view of food safety hazards and were more likely to return hazards that were explicitly mentioned to be high risk or to exceed maximum residue levels.

Table 1 provides the accuracy of the retrieved chemical hazards returned by the LLM for each of the validation and test foods for the tested prompts. Results are reported in two ways: number of correct responses divided by the total number of responses and the percentage that comes from this division (indicated in parentheses). The simple prompt obtained the worst performance among the three for the validation foods, with a performance of 89.3 % for shellfish and 65.6 % for leafy greens. The step-by-step prompt achieved a higher accuracy than the pseudocode prompt in leafy greens with 100 % compared to 93.8 %. For shellfish the pseudocode prompt obtained a marginally higher accuracy with 93.0 % than the step-by-step prompt with 92.5 %. It should be noted that the step-by-step prompt returned a substantially higher number of correct chemical hazards compared to the pseudocode prompt, resulting in 5 more hazards in leafy greens and 37 more in shellfish. Considering both the accuracy and returned number of chemical hazards, step-by-step prompt was chosen as the best performing prompt to be evaluated on the test foods. The performance of the step-by-step prompt on the test foods was highest for dairy with 98.7 %, followed by 92.0 % for maize and 88.9 % for salmon. The number of correctly retrieved chemical hazards was 76, 69 and 49 for dairy, maize and salmon respectively. A full list of the found chemicals for the test foods can be found in Appendix B.

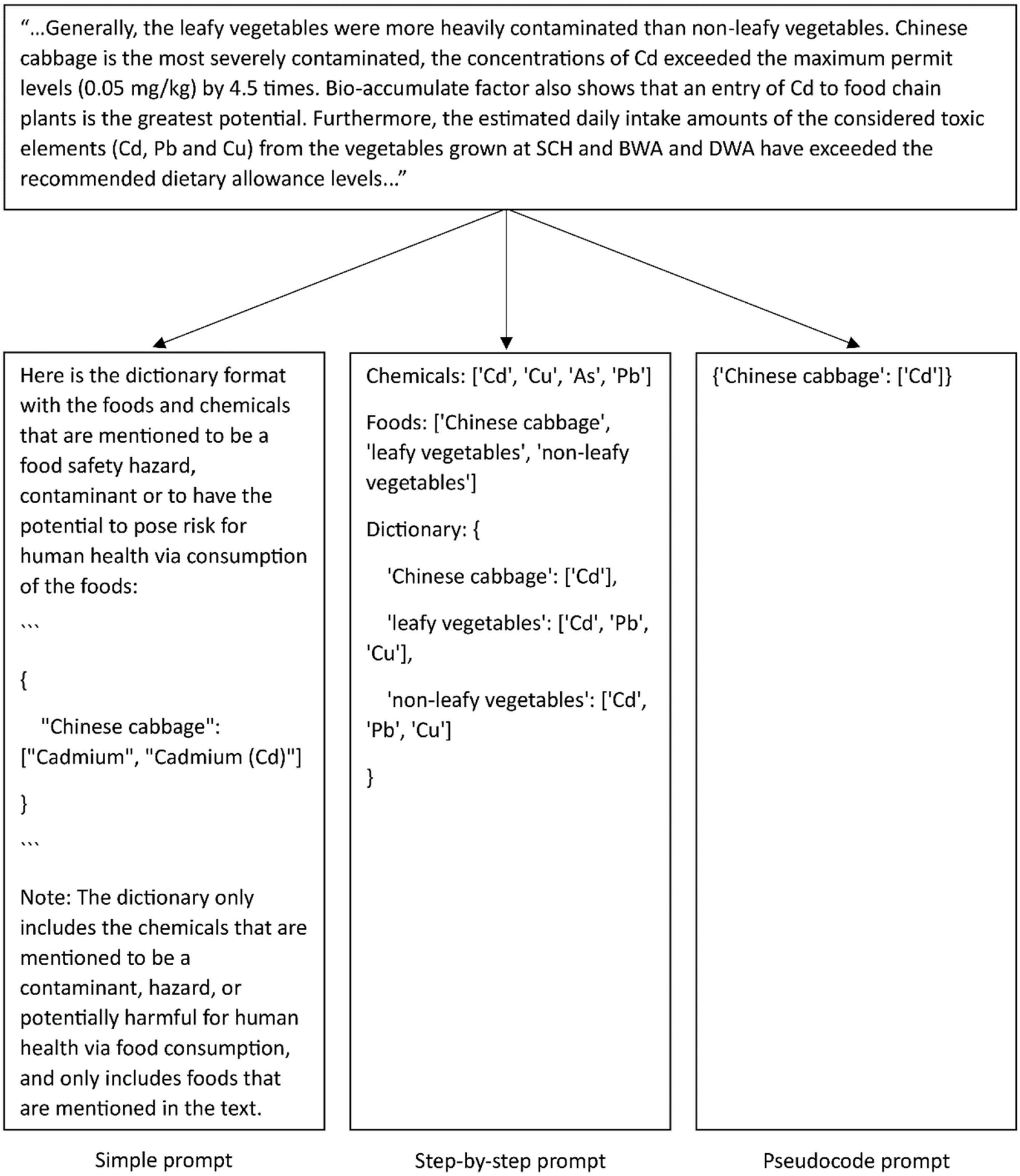
Correctly retrieved chemical hazards for dairy included veterinary drugs (e.g. corticosteroids, chloramphenicol and tetracycline), heavy metals (e.g. arsenic, cadmium and uranium), mycotoxins (e.g. aflatoxin B1+M1, deoxynivalenol and fumonisins), organic contaminants (e.g. bisphenol A+F, perfluorooctanoic acid and hexabromocyclododecane), pesticides (e.g. hexachlorobenzene, pentachlorophenol and deltamethrin) and biogenic amines (e.g. histamine, tyramine and cadaverine). For maize, the chemical hazards included mostly mycotoxins (e.g. ochratoxin A+B, beauvericin and sterigmatocystin), heavy metals (e.g. cadmium, chromium and mercury), pesticides (e.g. carbaryl, thifluzamide and chlorpyrifos) but also polycyclic aromatic hydrocarbons (e.g. pyrene) and pyrrolizidine alkaloids. The chemical hazards classified as correct for salmon included heavy metals (e.g. methylmercury, zinc and selenium), pesticides (e.g. emamectin benzoate, hexachlorobenzene, chlordane), veterinary drugs (e.g. medetomidine, ofloxacin and quinolones), polycyclic aromatic hydrocarbons (e.g. naphtalene, fluorene, phenanthrene), aromatic hydrocarbons (ethylbenzene, toluene and xylene), organotins (dibutyltin) and persistent organic contaminants (e.g. dioxins, furans and polychlorinated biphenyls).

Table 2 provides the chemical hazards retrieved for the test foods that were marked as incorrect by the domain expert, per food. Even though some of the chemicals are seen as food safety hazards, they were classified as incorrect since they were either not mentioned in the abstract or not in connection to the food studied. For dairy, carvacrol was the sole incorrect chemical. Our domain expert explained this to be a beneficial compound in oregano oil, which is used as a feed additive. The expert had marked arsenite, sulfonate, nitrate, aflatoxin M1, ellagic acid and fenpyrazamine as incorrect for maize. Arsenite was not present in the abstract the LLM collected the answer from and was therefore marked as incorrect. Sulfonate was incorrect because the abstract actually mentioned "6:2 fluorotelomer sulfonate" instead, but this was not present as an entity in ChEBI, while sulfonate was. Nitrate and aflatoxin M1 were marked as incorrect as they did not occur in maize in their abstracts, but in soil and milk respectively. Meanwhile, ellagic acid is a beneficial bioactive compound and therefore was not classified as a chemical hazard. Lastly, even though it is a hazardous chemical, fenpyrazamine was mentioned to be dangerous for corn salad, which is a type of leafy green, and is not a hazard for corn/maize. For salmon the expert marked aminobenzoic acid, chitosan, trimethylamine, mirex, polycyclic aromatic hydrocarbons and astaxanthin as incorrect. Aminobenzoic acid, mirex and polycyclic aromatic hydrocarbons were marked incorrect as they were not present in their corresponding abstracts. Chitosan and astaxanthin are described as unharmful chemicals added for prolonging the shelf life and enhancing the color of salmon respectively. Trimethylamine is the main spoilage product in salmon, but is not considered a food safety hazard.

4. Discussion

This study showed that chemical contaminants in foods can be extracted from scientific abstracts using an open-source LLM. This was done by providing textual instructions and without finetuning for the specific task at hand, making it easy to implement and recreate. Three differently styled prompts were evaluated by measuring accuracy of the responses of the LLM for a specific food item (i.e. leafy greens and shellfish) to determine the best performing prompt. The step-by-step prompt was established as the best prompt and was further evaluated for dairy, maize and salmon. Based on the evaluation of the domain expert, the performance for dairy, maize and salmon are very promising with accuracies of 98.7 %, 92.0 % and 88.9 % respectively.

Overall, this research set out to perform an information extraction task. Not all answers were considered correctly extracted chemical hazards, creating a group of false positives. Ideally, false positives should be kept to a minimum to increase the usability and trustworthiness of the model. Interestingly, some of the incorrect answers the LLM provided for the test cases could have been made by anyone without enough specific food safety domain knowledge. One such example is the misclassification of aflatoxin M1 in maize. The LLM extracted this result from an abstract that discussed aflatoxin M1 levels in milk samples, where they also tested maize-based cow feed for aflatoxin contamination (Yunus et al., 2020). The LLM classified aflatoxin



"...Generally, the leafy vegetables were more heavily contaminated than non-leafy vegetables. Chinese cabbage is the most severely contaminated, the concentrations of Cd exceeded the maximum permit levels (0.05 mg/kg) by 4.5 times. Bio-accumulate factor also shows that an entry of Cd to food chain plants is the greatest potential. Furthermore, the estimated daily intake amounts of the considered toxic elements (Cd, Pb and Cu) from the vegetables grown at SCH and BWA and DWA have exceeded the recommended dietary allowance levels..."

~~~
Here is the dictionary format with the foods and chemicals that are mentioned to be a food safety hazard, contaminant or to have the potential to pose risk for human health via consumption of the foods:

```

{

  "Chinese cabbage": ["Cadmium", "Cadmium (Cd)"]

}

```

Note: The dictionary only includes the chemicals that are mentioned to be a contaminant, hazard, or potentially harmful for human health via food consumption, and only includes foods that are mentioned in the text.
~~~

Simple prompt

~~~
Chemicals: ['Cd', 'Cu', 'As', 'Pb']

Foods: ['Chinese cabbage', 'leafy vegetables', 'non-leafy vegetables']

Dictionary: {

  'Chinese cabbage': ['Cd'],

  'leafy vegetables': ['Cd', 'Pb', 'Cu'],

  'non-leafy vegetables': ['Cd', 'Pb', 'Cu']

}
~~~

Step-by-step prompt

~~~
{'Chinese cabbage': ['Cd']}
~~~

Pseudocode prompt

Fig. 6. Examples of the LLM response patterns for each of the three prompts on an abstract concerning chemical hazards in leafy greens.

Table 1
Accuracy of the returned chemical hazards by the LLM for both the validation and test foods. All prompts were applied on the abstracts of the validation foods and the best prompt, which was the step-by-step prompt, was applied on the abstracts of the test foods.

	Validation foods		Test foods		
Prompt style	***Leafy greens***	***Shellfish***	***Dairy***	***Maize***	***Salmon***
Simple prompt	21/32 (65.6 %)	92/103 (89.3 %)	-/- (- %)	-/- (- %)	-/- (- %)
Step-by-step prompt	**22/22 (100 %)**	**74/80 (92.5 %)**	**76/77 (98.7 %)**	**69/75 (92.0 %)**	**49/55 (88.9 %)**
Pseudocode prompt	15/16 (93.8 %)	40/43 (93.0 %)	-/- (-%)	-/- (-%)	-/- (-%)

Table 2
Chemical hazards identified as incorrect in the LLM responses by the domain expert for each test food.

Dairy	Maize	Salmon
Carvacrol	Arsenite	Aminobenzoic acid
	Sulfonate	Chitosan
	Nitrate	Trimethylamine
	Aflatoxin M1	Mirex
	Ellagic acid	Polycyclic aromatic hydrocarbons
	Fenpyrazamine	Astaxanthin

M1 as an aflatoxin also present in the feed, but aflatoxins in maize are only converted into aflatoxin M1 after being consumed by cows. Nonetheless, domain knowledge is required to know that feed can't contain aflatoxin M1 and the abstract does not specify this. Another example is trimethylamine in salmon. The LLM extracts trimethylamine from an abstract discussing it as a spoilage product (Jääskeläinen et al., 2019). The formation in salmon is undesirable, and a reader with no specific knowledge might not have been able to assess whether trimethylamine is only a food quality or also a food safety issue. Similarly, for both carvacrol and astaxanthin it was stated in the abstract that they are an irritant to skin and eyes, but that as an additive they do not raise any safety concerns (Bampidis et al., 2022; Rychen et al., 2017). Since the prompts refer to retrieving contaminants with risks to human health, this is an understandable mistake. In light of these examples, it is obvious that this task cannot only be considered a reading comprehension task. Therefore, it is always important to check the findings of the LLM with a domain expert.

The most interesting behavior observed in the output of the LLM was its provision of specific hazardous chemicals in its responses when only the larger contaminant group was mentioned in the abstract (e.g. retrieving specific veterinary drugs when veterinary drugs are only mentioned as a group), despite it being explicitly instructed to only retrieve contaminants explicitly mentioned in the abstract. Initially, this explicit warning was included to prevent randomly crafted, incorrect answers. However, it was noted that during the validation and test procedures, the specific chemicals returned from abstracts only mentioning their wider group of hazards, were indeed hazardous for the food item of interest. This points to the LLM having a broader implicit knowledge of the food safety domain which it uses to give more specific results than just a group of contaminants. Therefore, these results are not penalized and were also considered correct. The LLM for example returned antimicrobials such as ampicillin and ciprofloxacin for dairy, which are indeed correct hazards, based on an abstract solely referring to the "antimicrobials" in water collected from milking parlors (Veiga-Gómez et al., 2017). Furthermore, in the case of salmon, organic contaminants such as lindane, heptachlor and chlordane, known to be contaminants in salmon, were returned by the LLM from an abstract discussing the levels of organic contaminants in an area known for its salmon farming industry in New Zealand (Niu et al., 2023).

Many chemical hazards currently monitored in food safety programs for the tested foods were present among the parsed output of the LLM, validating the approach by its successful retrieval of relevant hazards for the food safety domain. However, some relevant chemical contaminants present in the abstracts were not among the output and were categorized as false negatives. This was mostly caused by the LLM providing either an incorrect output format, causing the output to not be parsed correctly, or the contaminant being linked to a subcategory of the food instead. For example, a contaminant could be retrieved for milk, which should then also be linked to dairy. Future research can solve this problem by connecting the retrieved food with a food ontology, where the domain of food is described as a set of entities, classes and the relationships between them (Munir & Anjum, 2018). This way the food can automatically be linked to their broader categories, which creates a more complete picture of the contaminants present in food and makes it easier for the researcher to find food safety information across different levels of detail. Similarly, finetuning an LLM for the specific task at hand is a conventional, yet another encouraging avenue for future research. Although it is more time-intensive and computationally costly to finetune an LLM, it will very likely reduce the deviations from the requested output format significantly, as it will have been trained with that format in mind. This makes parsing the output easier and could lead to the identification of more chemical hazards. Furthermore, the probability of a correct chemical contaminant not being extracted from an abstract will presumably become even lower. Both effects will reduce the number of false negatives, therefore creating a more robust system. Developments to address the computational and memory costs associated with finetuning of LLMs are actively being worked on and have led to novel and promising approaches such as Low-Rank Adaptation (LoRA) (Hu et al., 2021).

The prompt styles experimented with in this study included three common techniques of prompting. However, in the ever-expanding literature on prompt engineering and LLMs many more styles and techniques do exist. It would have been too time consuming to try each possible approach presented in the literature to identify the best performing style. In contrast, the study by Yang et al. (2023) proposed an iterative approach in which LLMs are instructed with a "meta-prompt". In this prompt the LLM is asked to create a prompt for a specific task itself, and the newly created prompt is improved step-by-step by the LLM in line with the performance evaluated on a small dataset. They showed the prompts created by the LLM outperform those created by humans. Adopting this approach would be another promising direction for future research.

Although good results were achieved with LLMs in this research, these models are not without their limitations. As can be seen in our results, the specific wording of the used prompt can have a big impact on the quality of the LLM's response. Prompts that are used for LLMs should be carefully evaluated and variations should be tested to steer the output of the model in the right direction. Furthermore, LLMs can suffer from so called hallucinations, where the responses contain factually incorrect information or they do not seem to answer the question that was asked in the right way (Ye et al., 2023). Hallucinations can be caused by biases in the (pre-)training data that can contain misinformation or there might not be enough information on the topic to answer the question at all. Hallucinations can be reduced by carefully designing the prompt and by providing the context that the LLM can use to answer. It is, however, wise to not take an LLM's response at face value and leave room for expert judgement. Lastly, the interpretation of an LLM and it responses is still a challenging problem. Because of their complicated architecture and data-driven decisions they are considered a "black box". It is hard to determine why an LLM gives the response it does, making the models less transparent and trustworthy. Interpretability of LLMs is an area of active research, which can lead to better understanding of their decision-making process in the future (Bills et al., 2023; Conmy et al., 2023; Treviso et al., 2023). Even though LLMs are the current state-of-the-art at information extraction tasks, these limitations should be kept in mind when utilizing them and careful post-processing of the responses should be considered.

This research laid the foundation for a tool that food safety experts can use to get an overview of hazardous chemicals for any food from the literature. It was demonstrated that LLMs are a powerful, accurate instrument for automatic information extraction. They can be applied by giving textual instructions without the need for labelled data or to teach them information on the food safety domain first. By automating the data collection and prompting of the LLM, such a tool can be continuously updated with new scientific literature and therefore be ensured to remain up-to-date. Links to the abstracts from which the LLM extracted the chemical hazards can be provided, so that the findings can be checked in case a result is deemed to be unexpected or unrealistic. Additionally, the frequency with which a chemical has been discussed in the literature for a specific food could be supplied. If a hazard has not been discussed often before, this could be used by food safety experts as a potential indication of an emerging risk and assist them in taking action around such risks more swiftly.

5. Conclusion

In this study, the application of an LLM for the extraction of chemical hazards from the scientific literature for specific foods was demonstrated. It was shown that the LLM can successfully retrieve relevant chemical contaminants and makes few errors. Across a set of three different foods, the average correct response rate was 93 %. Multiple styles of prompts were tested to assess which was most optimal for the task at hand. The specific wording and style of the prompt was found to have a considerable effect on the performance. It was concluded that a prompt breaking the task down into smaller steps performed best overall. Using this approach, the information collection from scientific literature on chemical hazards in food can be automated and save researchers' valuable time. By automatically updating the results with the newest literature and adding frequency on how often a contaminant has been mentioned for a specific food, the approach could provide a way of detecting emerging hazards so that timely action can be taken to improve food safety.

Code and data availability

The code and data used in this study can be found on https://github.com/WFSRDataScience/LLMForChemicalFoodSafetyHazardExtraction.

Author contributions

Neris Özen: Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Wenjuan Mu:** Methodology, Supervision, Writing - review & editing. **Esther D. van Asselt:** Validation, Writing - review & editing. **Leonieke M. van den Bulk:** Conceptualization, Methodology, Project administration, Supervision, Writing - original draft, Writing - review & editing.

Ethical Statement – Studies in humans and animals

This study does not contain any research with human or animal subjects.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research received funding from the Dutch Ministry of Agriculture, Nature and Food Quality (LNV) via the Knowledge Base Program "Healthy and Safe food systems" (KB-37-002-036).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.afres.2024.100679.

References

Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., & Sontag, D. (2022). Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 1998–2022*.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, É., Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., & Penedo, G. (2023). The falcon series of open language models. https://arxiv.org/abs/2311.16867.

Bampidis, V., Azimonti, G., Bastos, M., de, L., Christensen, H., Dusemund, B., Fašmon Durjava, M., Kouba, M., López-Alonso, M., López Puente, S., Marcon, F., Mayo, B., Pechová, A., Petkova, M., Ramos, F., Sanz, Y., Villa, R. E., Woutersen, R., Galobart, J., Holcznecht, O., & Vettori, M. V. (2022). Safety and efficacy of a feed additive consisting of astaxanthin-rich Phaffia rhodozyma for salmon and trout (Igene Biotechnology, Inc.). *EFSA Journal, 20*(2), e07161. https://doi.org/10.2903/j.efsa.2022.7161

Banach, J. L., Hoffmans, Y., & van Asselt, E. D. (2019). Overview of chemical hazards in leafy vegetables. *RIKILT Wageningen University & Research, Report 2019.013*.

Bills, Steven, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders (2023). Language models can explain neurons in language models. *https://openai.com/index/language-models-can-explain-neurons-in-language-models/*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & others. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

Chiang, W., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I. & Xing, E. P. (2023,). Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90 %* ChatGPT Quality. Retrieved from https://lmsys.org/blog/2023-03-30-vicuna/.

Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. In *Journal of big data, 9*. Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1186/s40537-022-00561-y

Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems, 36*, 16318–16352.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT, 1*, 4171–4186.

Fulford, I., & Ng, A. (2023). *ChatGPT prompt engineering for developers [Online course]*. Deeplearning.AI. https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/.

Gu, Y., Zhang, S., Usuyama, N., Woldesenbet, Y., Wong, C., Sanapathi, P., Wei, M., Valluri, N., Strandberg, E., Naumann, T., & others. (2023). Distilling large language models for biomedical knowledge extraction: A case study on adverse drug events. *rXiv Preprint ArXiv:2307.06439*.

Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P., & Steinbeck, C. (2016). ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research, 44*(D1), D1214–D1219.

Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and advances in information extraction from scientific literature: A review. In *JOM, 73* pp. 3383–3400). Springer. https://doi.org/10.1007/s11837-021-04902-9

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In I. Gurevych, & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 328–339). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1031.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *ArXiv Preprint ArXiv:2106.09685*.

Jääskeläinen, E., Jakobsen, L. M., Hultman, J., Eggers, N., Bertram, H. C., & Björkroth, J. (2019). Metabolomics and bacterial diversity of packaged yellowfin tuna (Thunnus albacares) and salmon (Salmo salar) show fish species-specific spoilage development during chilled storage. *International journal of food microbiology, 293*, 44–52.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications, 82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Kluche, M., den Hil, E. F., & van Asselt, E. D. (2020). Overview of chemical hazards in cereals, seeds and nuts. *RIKILT Wageningen University & Research, Report 2020.003*.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information, 10*(4), 150.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv Preprint ArXiv:1910.13461*.

Lin, B., Hibschman, J., & Oshima, K. (2023). Automatic Dish Name Extraction from User-generated Content Using LLM. *Technical Disclosure Commons*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1–35.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering, 35*(1), 857–876.

Liu, Y., & Lapata, M. (2019). Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3730–3740). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1387.

Luo, J., Leng, S., & Bai, Y. (2022). Food supply chain safety research trends from 1997 to 2020: A bibliometric analysis. *Frontiers in Public Health, 9*, Article 742980.

Luong, M.-T., Kayser, M., & Manning, C. D. (2015). Deep neural language models for machine translation. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (pp. 305–309).

Min, B., Ross, H., Sulem, E., Veyseh, A. P. Ben, Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys, 56*(2), 1–40.

Min, K., & Wilson, W. H. (2006). Comparison of numeral strings interpretation: Rule-based and feature- based N-gram methods. *Australasian Joint Conference on Artificial Intelligence*, 1226–1230.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning– based text classification: A comprehensive review. *ACM Computing Surveys (CSUR), 54*(3), 1–40.

Mishra, M., Kumar, P., Bhat, R., Murthy V, R., Contractor, D., & Tamilselvam, S. (2023). Prompting with Pseudo-Code instructions. *ArXiv Preprint ArXiv:2305.11790*.

Munir, K., & Anjum, M. S. (2018). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics, 14*(2), 116–126.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association, 18*(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464

Niszczota, P., & Rybicka, I. (2023). The credibility of dietary advice formulated by ChatGPT: robo-diets for people with food allergies. *Nutrition, 112*, Article 112076.

Niu, S., Chen, R., Hageman, K. J., McMullin, R. M., Wing, S. R., & Ng, C. A. (2023). Understanding impacts of organic contaminants from aquaculture on the marine environment using a chemical fate model. *Journal of Hazardous Materials, 443*, 130090.

Paroiu, R., Ruseti, S., Dascalu, M., Trausan-Matu, S., & McNamara, D. S. (2023). Asking questions about scientific articles—Identifying large N studies with LLMs. *Electronics, 12*(19), 3996.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1202.

Qi, Z., Yu, Y., Tu, M., Tan, J., & Huang, Y. (2023). Foodgpt: A large language model in food testing domain with incremental pre-training and knowledge graph prompt. *ArXiv Preprint ArXiv:2308.10173*.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., & Yang, D. (2023). Is ChatGPT a general-purpose natural language processing task solver? *ArXiv Preprint ArXiv: 2302.06476*.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1*(8), 9.

Rychen, G., Aquilina, G., Azimonti, G., Bampidis, V., Bastos, M., de, L., Bories, G., Cocconcelli, P. S., Flachowsky, G., Gropp, J., Kolar, B., Kouba, M., López-Alonso, M., López Puente, S., Mantovani, A., Mayo, B., Ramos, F., Saarela, M., Villa, R. E., Wallace, R. J., Wester, P., Brantom, P., Dusemund, B., van Beelen, P., Westendorf, J., Gregoretti, L., Manini, P., & Chesson, A. (2017). Safety and efficacy of an essential oil from Origanum vulgare subsp. hirtum (Link) letsw. var. Vulkan when used as a sensory additive in feed for all animal species. *EFSA Journal, 15*(12), e05095. https://doi.org/10.2903/j.efsa.2017.5095

Santu, S. K. K., & Feng, D. (2023). TELeR: A general taxonomy of LLM prompts for benchmarking complex tasks. *ArXiv Preprint ArXiv:2305.11430*.

Shi, Y., Ren, P., Wang, J., Han, B., ValizadehAslani, T., Agbavor, F., Zhang, Y., Hu, M., Zhao, L., & Liang, H. (2023). Leveraging GPT-4 for food effect summarization to enhance product-specific guidance development via iterative prompting. *Journal of Biomedical Informatics, 148*, Article 104533.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., & others. (2023). Llama: Open and efficient foundation language models. *ArXiv Preprint ArXiv:2302.13971*.

Treviso, M., Ross, A., Guerreiro, N. M., & Martins, A. F. (2023). CREST: A joint framework for rationalization and counterfactual text generation. *arXiv preprint arXiv:2305.17075*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. https://doi.org/10.48550/arXiv.1706.03762

Veiga-Gómez, M., Nebot, C., Franco, C. M., Miranda, J. M., Vázquez, B., & Cepeda, A. (2017). Identification and quantification of 12 pharmaceuticals in water collected from milking parlors: Food safety implications. *Journal of dairy science, 100*(5), 3373–3383.

Wang, P., Qian, Y., Soong, F. K., He, L., & Zhao, H. (2015). Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *ArXiv Preprint ArXiv:1510.06168*.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). Finetuned language models are zero-shot learners. *International Conference on Learning Representations*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V, Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., & others. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv Preprint ArXiv: 1609.08144*.

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., & Chen, E. (2023). Large language models for generative information extraction: A survey. *ArXiv Preprint ArXiv: 2312.17617*.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V, Zhou, D., & Chen, X. (2023). Large language models as optimizers. *ArXiv Preprint ArXiv:2309.03409*.

Ye, H., Liu, T., Zhang, A., Hua, W., & Jia, W. (2023). Cognitive mirage: A review of hallucinations in large language models. *ArXiv preprint ArXiv:2309.06794*.

Yunus, A. W., Ullah, A., Lindahl, J. F., Anwar, Z., Ullah, A., Saif, S., … Ibrahim, M. N. M. (2020). Aflatoxin contamination of milk produced in peri-urban farms of Pakistan: Prevalence and contributory factors. *Frontiers in microbiology, 11*, 159.

Zaman, G., Mahdin, H., Hussain, K., & Rahman, A. (2020). Information extraction from semi and unstructured data sources: A systematic literature review. *ICIC Express Letters, 14*(6), 593–603.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *International Conference on Machine Learning*, 11328–11339.

Zhu, W., Liu, H., Dong, Q., Xu, J., Kong, L., Chen, J., Li, L., & Huang, S. (2023). Multilingual machine translation with large language models: Empirical results and analysis. *ArXiv Preprint ArXiv:2304.04675*.