Computer Methods and Programs in Biomedicine 255 (2024) 108326

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

# Information extraction from medical case reports using OpenAI InstructGPT

Veronica Sciannameo [a,1], Daniele Jahier Pagliari [b,1], Sara Urru [c], Piercesare Grimaldi [d], Honoria Ocagli [c], Sara Ahsani-Nasab [c], Rosanna Irene Comoretto [d], Dario Gregori [c], Paola Berchialla [a,*]

[a] Centre for Biostatistics, Epidemiology and Public Health, Department of Clinical and Biological Sciences, University of Turin, Regione Gonzole 10, Orbassano 10043, Italy
[b] Department of Control and Computer Engineering, Politecnico di Torino, Turin 10129, Italy
[c] Unit of Biostatistics, Epidemiology and Public Health, Department of Cardiac, Thoracic, Vascular Sciences and Public Health, University of Padova, Padua, Italy
[d] Department of Public Health and Pediatrics, University of Torino, Via Santena 5 bis, Torino 10126, Italy

ARTICLE INFO

*Keywords:*
Large language model
Natural language processing
Information retrieval
Case reports

ABSTRACT

*Background and objective:* Researchers commonly use automated solutions such as Natural Language Processing (NLP) systems to extract clinical information from large volumes of unstructured data. However, clinical text's poor semantic structure and domain-specific vocabulary can make it challenging to develop a one-size-fits-all solution. Large Language Models (LLMs), such as OpenAI's Generative Pre-Trained Transformer 3 (GPT-3), offer a promising solution for capturing and standardizing unstructured clinical information. This study evaluated the performance of InstructGPT, a family of models derived from LLM GPT-3, to extract relevant patient information from medical case reports and discussed the advantages and disadvantages of LLMs versus dedicated NLP methods.

*Methods:* In this paper, 208 articles related to case reports of foreign body injuries in children were identified by searching PubMed, Scopus, and Web of Science. A reviewer manually extracted information on sex, age, the object that caused the injury, and the injured body part for each patient to build a gold standard to compare the performance of InstructGPT.

*Results:* InstructGPT achieved high accuracy in classifying the sex, age, object and body part involved in the injury, with 94%, 82%, 94% and 89%, respectively. When excluding articles for which InstructGPT could not retrieve any information, the accuracy for determining the child's sex and age improved to 97%, and the accuracy for identifying the injured body part improved to 93%. InstructGPT was also able to extract information from non-English language articles.

*Conclusions:* The study highlights that LLMs have the potential to eliminate the necessity for task-specific training (zero-shot extraction), allowing the retrieval of clinical information from unstructured natural language text, particularly from published scientific literature like case reports, by directly utilizing the PDF file of the article without any pre-processing and without requiring any technical expertise in NLP or Machine Learning. The diverse nature of the corpus, which includes articles written in languages other than English, some of which contain a wide range of clinical details while others lack information, adds to the strength of the study.

## 1. Introduction

Natural Language Processing (NLP) has become a key resource in clinical research, particularly for extracting clinical information from unstructured natural language texts contained in data sources such as Electronic Health Records (EHRs) and clinical notes. Researchers increasingly turn to automated solutions to extract crucial information quickly and efficiently from large volumes of unstructured data and convert it into a structured format to feed machine learning and statistical models.

A review by Kreimeyer et al. [1] identified various NLP systems (i.e. rule-based NLP approaches, hybrid systems, purely machine learning

* Corresponding author.
*E-mail address:* paola.berchialla@unito.it (P. Berchialla).
[1] These authors equally contributed to the work.

https://doi.org/10.1016/j.cmpb.2024.108326
Received 11 April 2023; Received in revised form 15 June 2023; Accepted 11 July 2024

methods) that are used to extract unstructured clinical information from sources like clinical notes, radiology and pathology reports, biomedical literature, and clinical trial documents. These systems are primarily used to extract drug names, dosages, specific symptoms, and pathologies. In addition, several libraries have been developed for use within dedicated software. Examples include rEHR for managing and analyzing EHR data [2], ctrdata for retrieving and analyzing clinical trials in public registers [3], and medExtractR for extracting medication information from clinical notes combining lexicon dictionaries and regular expressions [4], all of which are available for the popular R software. However, the unique characteristics of clinical text, such as its poor semantic structure and the presence of domain-specific vocabularies, make it challenging to develop a one-size- fits-all solution for NLP systems which are, instead, often tailored to specific domains. As an example, Regextractor [5] uses 39 manually designed regular expressions for retrieving specific information from a pulmonary function test text data, or COAT [6], which is a hybrid approach based on rules and machine learning algorithms specifically developed to retrieve the Gleason score, tumour stage, and margin status from pathology reports.

The machine learning approach to NLP solutions presents several challenges due to the requirement for training and fine-tuning on appropriate data to identify specific pieces of information of interest accurately. This process is time-consuming and requires experts with advanced skills in NLP. For example, Regular Expression Discovery Extractor (REDEx) is a supervised learning algorithm that uses regular expressions [7] that was trained on 268 primary care notes (plus snippets from another 300) and tested on 3561 notes, to retrieve bodyweight-related measures, such as weight, height, BMI, and abdominal circumference.

More generally, very few systems have been applied to multiple clinical domains, indicating that these systems may have a narrow focus. Thus, there is a need for NLP systems that are more widely applicable and flexible, capable of adapting to diverse domains.

An alternative solution for capturing and standardizing unstructured natural language clinical information is offered by Large Language Models (LLMs), such as OpenAI's Generative Pre-Trained Transformer 3 (GPT-3) [8]. LLMs are large deep neural networks based on the Transformer architecture, which have been able to produce human-like results on a wide range of different NLP tasks with minimal (or no) specialized training.

One of the key factors behind the impressive results of modern LLMs like GPT-3 is their self-supervised pre-training procedure, which leverages vast amounts of text available on the Internet. Pieces of text are fed to an LLM after masking some words or removing the end, and the model is trained to reconstruct or complete the missing parts. This allows LLMs to learn the task-independent structure of natural language from a vast amount of data, which would not be available in a labelled form [9]. Then the model can perform various tasks without being specifically trained for them, or at most after a fine-tuning phase on a much smaller amount of labelled data.

GPT-3 has been applied to tasks such as text and code completion, translation, correction, and optimization [10–12]. There is a consensus among healthcare researchers that GPT-3 and other LLMs could potentially be used to improve the efficiency of automatic tools for extracting clinical information from unstructured natural language texts present in EHRs [13,14]. However, the standard LLM learning procedure is not designed to instruct these systems to extract specific pieces of clinical information from a block of free text. For this reason, some additional partially or completely supervised training steps are required to align LLM outputs with user needs. An example is the procedure used to generate InstructGPT [15], a family of models directly derived from GPT-3 using an additional three-step procedure based on human feedback.

Our study aims to evaluate the performance of InstructGPT in extracting clinical data from unstructured natural language free text from published papers directly from the PDF file. Specifically, we provide prompts to InstructGPT to extract relevant patient information from medical case reports and assess its performance as a ready-to-use tool to assist researchers in similar routine tasks. Extracting this type of information from published case reports can be useful for conducting secondary analyses on published works and, for example, providing a descriptive overview of all the available literature on a specific topic of interest. Additionally, extracted information could be used for more advanced literature analyses, such as Structural Topic Modeling, which also allows the inclusion of covariates. Finally, we compare the advantages and disadvantages of LLMs versus dedicated NLP methods.

## 2. Materials and methods

### 2.1. Data source

We identified 208 publications related to case reports on pediatric body injuries from January 1, 2017, to December 31, 2021, searching PubMed, Scopus, and Web of Science (WoS). Foreign body injuries in children can be caused by various types of objects, such as small toys, coins, or food, and can occur in different parts of the body (ears, eyes, nose, etc.). They can range in severity and may require different types of treatment depending on the location and type of foreign body involved [16]. The aim of this collection of clinical case reports was to evaluate the performance of InstructGPT as a tool for extracting clinical information from a broad range of pediatric foreign body injuries, for which there is currently no automated tool available.

The articles gathered were manually downloaded in PDF format, with 95% (197 articles) written in English and the remaining written in Russian (4), French (3), Spanish (2), Danish (1) and Dutch (1). We used the Python library PDFPlumber version 0.7.6 with default parameters [17] to extract the text (including titles, page headers/footers, references, etc.) from the PDF [8].

### 2.2. Reference method: manual labelling from human

A single reviewer manually extracted information on sex, age, injured body part, and object that caused the injury for each patient from the articles. The gold standard should require two annotators, but in this case, given the descriptive nature of the case reports, the type of information to be extracted was straightforward for a human annotator. Thus, to carry out the task, a medical resident was employed. Moreover, a second different person validated the results of InstructGPT, thus allowing for verification of the correctness of the information extracted by the annotator. The corresponding labels for this information served as the gold standard reference method against which we compared the performance of InstructGPT. For articles written in Danish, Dutch, French, Russian, and Spanish, we used Google Translate to retrieve this information as none of the authors are proficient in these languages.

### 2.3. Testing method: InstructGPT

InstructGPT is a family of Deep Learning (DL) algorithms developed by OpenAI [18], derived by fine-tuning the Generative Pre-trained Transformer 3 (GPT-3) model via a technique called Reinforcement Learning from Human Feedback (RFHL). The fine-tuning process consists of three steps: (i) human labellers create pairs of input text (prompts) and corresponding desired output. The model is trained to generate the same output as the labeller through supervised learning. (ii) The model is given a prompt and generates multiple possible outputs, which are then ranked by human labellers based on their alignment with the prompt. These rankings are used to construct a Reward Model (RM), which assigns a higher reward to higher-ranked outputs. (iii) The model is further fine-tuned using the previously constructed RM through a reinforcement learning approach [15].

InstructGPT models are accessible through a public application programming interface (API) in Python, which allows users to provide

prompts to the trained model hosted in the cloud and receive the models' response [18].

InstructGPT processes text by dividing it into tokens. The amount of text that can be fed into the model is restricted to 4,097 tokens, which was not always sufficient to include the entire text of a medical case report. Therefore, to feed InstructGPT as much text as possible, we employed the following iterative procedure: (i) we initially provided InstructGPT with the entire text or the first 12,000 characters if the document was longer; (ii) if the tokenization process resulted in more than 4,097 tokens, we removed 2,000 characters from the end of the text and repeated the API invocation up to a maximum of 5 times.

We chose to exclude text from the end of the document as the majority of clinically relevant data is typically found in the initial pages of the report, whereas the final pages usually contain less important information, such as references and authors' bios.

We used the InstructGPT variant called "text-davinci-003", because it is the most capable among the four base models of the family and allowed us to use the highest number of tokens. Furthermore, text-davinci-003 can handle complex instructions.

When making an API request, users can specify additional parameters to influence the response provided by InstructGPT. The two most important are temperature and top_p, which affect the determinism of the model's response. The official documentation [18] recommends only adjusting one of these parameters, so in our experiments, we set the temperature to 0.5 and kept top_p fixed at 1. The chosen value of temperature strikes a good balance between determinism and creativity in the model's output. We wanted to extract specific pieces of information with a single correct answer (high determinism). Still, we also wanted the model's output to be in a machine-readable format that could be directly used for statistical analyses. With a well-designed prompt message, this can be achieved in a single attempt for most documents. However, since the model is trained on natural language texts, it may occasionally generate the response in an unexpected format. By avoiding a temperature that is too low, we reduced the likelihood of getting such a wrong format multiple times when repeating the request.

We asked InstructGPT to extract information on the child's sex, age, injured body part, and the type of object that caused the injury from the text of the articles. We follow the guidelines [18] to generate prompt messages for each of these extractions. The specific prompts provided to InstructGPT are shown in Table 1:

For sex, age, and body part, we only accepted responses from InstructGPT that were directly in a machine-readable format, i.e., a single letter (M or F) for sex extraction; one or two numbers followed by Y or M for age extraction (e.g., 3Y, 18 M, or 1Y6M); one of the categories listed in the prompt for body part extraction.

Responses in any other format were automatically rejected, and the extraction was repeated using a new API request. For the object extraction, we did not impose any constraints on the response as the number of possible objects is virtually unlimited and cannot be easily organized into categories. In this case, the accuracy of the response compared to the reference human annotation was checked manually.

The performance of InstructGPT was evaluated using accuracy, which measures the number of correctly extracted data compared to the total number of case reports containing that information, and Cohen's Kappa, to evaluate the concordance between InstructGPT and the gold standard (i.e., the manual human extraction).

All analyses were performed using Python [19] and R 4.2.1 software [20]. More in detail, Python was used to perform the text extraction from PDF and to interact with InstructGPT; while R was used to analyze the data retrieved.

**Table 1**
InstructGPT prompts for extraction of clinical information.

| Data to extract from PDF | InstrucGPT Prompt Text |
|---|---|
| **Sex** | "I will give you the text of a medical case report paper. Tell me the sex of the patient subject of the study. Write M if the sex is male and F if the sex is female. Write no other output. This is the text of the paper: {message}. The sex of the subject is:" |
| **Age** | "I will give you the text of a medical case report paper. Tell me the age of the patient subject of the study. Reply with a single number and a unit of measure: Y for years, M for months. Reply with N/A if you don't know the answer. Write no other output. This is the text of the paper: {message}. The age of the subject is: |
| **Part of the Body** | "I will give you the text of a medical case report paper. Tell me the body part of the patient that was injured. Write no other output. This is the text of the paper: {message}<br>Reply selecting the subject body part that was injured from one or more of these categories, as lowercase words separated by a comma.<br>- head<br>- eye<br>- ears<br>- nose<br>- mouth<br>- neck<br>- throat<br>- trachea<br>- esophagus<br>- stomach<br>- abdomen<br>- bowel<br>- lung<br>- bronchus<br>- bladder<br>- genitals<br>- arm<br>- leg<br>- other<br>The subject body part that was injured is:" |
| **Object** | "I will give you the text of a medical case report paper. Tell me the object that hurt the patient. This is the text of the paper: {message}. The object that hurt the subject is:" |

## 3. Results

The corpus of PDF publications on pediatric foreign body injuries published from 2017 to 2021 consists of 208 articles. When retrieving the child's sex, InstructGPT correctly classified 94% of the articles (with a Cohen's Kappa of 0.88), correctly identifying 78 females and 109 males. When extracting the age of the patients, InstructGPT correctly retrieved information from 82% of the case reports (168 articles), with a Spearman correlation coefficient of 0.99 when considering age expressed in months. Overall, InstructGPT failed to retrieve any information about age on 32 articles and provided a wrong age in 5 articles. Among them, two errors can be considered negligible (2 years instead of the reported 2.5 years and 2 instead of the reported 2.33 years). InstructGPT also performed well in extracting information about the object and part of the body that was injured, with accuracies of 94% and 89%, respectively. When extracting the part of the body that was injured, InstructGPT was unable to retrieve the information in 8 papers, while when extracting information about the object, it was unable to retrieve it in 2 papers. More detailed results can be found in Tables 2 and 3.

Excluding articles for which InstructGPT was unable to retrieve any information resulted in improving accuracies of 97% for the child's sex and age and 93% for the part of the body that was injured.

In 5 articles, InstructGPT produced incorrect extractions for all tasks.

**Table 2**
Confusion matrix between the actual class of children sex (Human) and the information extracted by InstructGPT from articles. NC = Not Classified.

| | | InstructGPT | | |
|---|---|---|---|---|
| | | *F* | *M* | *NC* |
| **Human** | ***F*** | 78 (39.2%) | 3 (1.5%) | 2 (1.0%) |
| | ***M*** | 3 (1.5%) | 109 (54.8%) | 4 (2.01%) |

**Table 3**
Number of articles for which child's sex, age, part of the injured body, and object were extracted correctly (accuracy) and wrongly or not extracted (errors + Not Classified (NC)).

| | Sex | Age | Part of the body | Object |
|---|---|---|---|---|
| **Correct (accuracy)** | 187 (94%) | 168 (82%) | 183 (89%) | 188 (94%) |
| **Mistake (errors + NC)** | 12 (6%) | 37 (18%) | 22 (11%) | 13 (6%) |
| **Total number of articles reporting the information** | 199 | 205 | 205 | 201 |

One of these articles contained a review in addition to the case report, and errors occurred during the text extraction from the PDF process for two others due to limitations of the PDFPlumber Python library, preventing InstructGPT from reading them.

In supplementary Table S1 we reported the errors committed by InstructGPT on the extraction of sex, age, part of the injured body, and object compared with the reference method.

## 4. Discussion

The study aimed to explore the ability of LLMs to extract information from raw PDFs of clinical case reports automatically. To this purpose, we used instructGPT on case reports about pediatric foreign body injuries to extract data on child's sex, age, injured body part, and foreign body type. Our analysis included 208 articles from PubMed, Scopus, and WoS written in various languages. The results of the study showed that InstructGPT was able to accurately extract clinical data with the highest performance achieved in retrieving the child's sex and type of foreign body that caused the injury (accuracy = 94%) and the lowest one obtained when retrieving age (accuracy = 82%). The relatively low accuracy on age could be due to the inconsistent reporting, i.e. age was sometimes reported using months, sometimes years and other times using decimal points (e.g., 2.5 years). This hypothesis is supported by the fact that InstructGPT exhibited more errors when retrieving ages expressed in months, an aspect that may not have been adequately accounted for in the prompt message. As a result, it is possible that better accuracy could be achieved by revising the prompt message to account for this variation.

This result suggests that pre-trained LLMs like InstructGPT can be effectively used to automatically extract information from raw PDF file format of clinical case reports without task-specific fine-tuning, making instructGPT an almost ready-to-use tool for information retrieval. The diverse nature of our corpus, which includes papers written in multiple languages and encompasses various fields with varying degrees of missing information, contributes to the strength of our study.

Among the several NLP systems trained to extract numerical information from text, Regular Expression Discovery Extractor (REDEx) and Regextractor showed the highest accuracy. REDEx achieves a precision of 98.3% to retrieve bodyweight-related numerical values such as weight, height, BMI, and abdominal circumference, whereas Regextractor [5] resulted in an accuracy of 99.5% on pulmonary function test text.

Regextractor and REDex outperformed InstructGPT in accuracy when applied to retrieving age. However, it is worth noting that the input data assigned to Regextractor was machine-generated, which may limit its ability to extract information from unstructured human-generated documents. On the other hand, REDex, which is a machine learning-based system, may require a larger amount of data for training to perform well in different clinical contexts. In contrast, InstructGPT has the flexibility to be easily adapted for use in various domains and can perform a wide range of tasks without the need for extensive training data.

One of the key advantages of IntructGPT is its flexibility in not being limited to specific types of output, such as numbers, but rather allowing for the extraction of multiple types of information. In the literature, other NLP systems exist that allow for multiple types of extraction, such as COAT and BioMedICUS [21]. For example, COAT achieves extremely higher accuracy in retrieving the Gleason score, tumour stage, and margin status from pathology reports (99.7%, 99.1%, and 97.2%, respectively).

However, several key differences between InstructGPT and COAT or BioMedICUS make InstructGPT a competitive option. First, while COAT and BioMedICUS are designed to work on specific types of documents with pre-defined formatting, InstructGPT can be applied to any document. For example, COAT is designed to extract information from pathology reports, which are typically formatted as attribute-value pairs (e.g., "Margin Status: Negative") [22], greatly simplifying the extraction, while BioMedICUS relies on rule-based techniques to identify section and subsection headings to localize the text that contains the required information (e.g., patients' family history). Furthermore, a rule-based system, although not requiring training data, necessitates an expert to create the rules themselves, in addition to typically having several configuration parameters that need to be correctly set for each type of extraction. On the contrary, the InstructGPT-based approach only requires the construction of a prompt in natural language, which is much easier to prepare, even for a medical professional and not a statistics/ computer science expert. Secondly, the more complex extractions performed by COAT and BioMedICUS require a machine learning model that has been specifically trained for that task. For instance, the COAT pipeline described in [22] uses a Support Vector Machine (SVM) trained on 782 pathology reports to extract surgical margin status, and BioMedICUS uses an SVM trained on 329 documents to identify sentences containing a patient's family history. In contrast, InstructGPT does not require any training to make it usable on small sets of documents as well. Thirdly, adapting COAT or BioMedICUS to extract new types of information can be challenging and requires advanced NLP skills to design a new pipeline and appropriate regular expressions. In contrast, with InstructGPT, users only need to provide a human-readable prompt in the form of those reported in Table 1, making it easier for non-NLP experts, such as physicians, to utilize by following simple guidelines.

In summary, the main advantage of using LLMs such as InstructGPT for data extraction from articles is that they do not require any data preprocessing or training, thus enabling an entirely "no-code" approach for the extraction. Furthermore, they do not require any programming skills in that it is sufficient to input the queries in natural language. The absence of the training phase also reduces the amount of data required, because no splitting into training/test set is necessary. Finally, since the model is hosted in cloud and accessible through a web API, combined with the absence of data pre-processing and training, makes it unnecessary to have high-performing computers.

This approach also overcomes language barriers, as we had papers written in Dutch, French, Russian, Danish, and Spanish, from which InstructGPT had no difficulties extracting the required information (Supplementary Table S2).

One potential limitation of the study is that paper processing cannot be fine-tuned to the wanted search. Linked to that, another limitation is that the LLM is used as a "black box", making it difficult to analyze, interpret, and diagnose extraction errors. In general, the effectiveness of the approach is completely dependent on the model provider (OpenAI in the case of InstructGPT). Other NLP approaches could provide more accurate results once tailored for a specific type of extraction. However, a solution such as InstructGPT could be very useful for small corpora of documents (not allowing task-specific training) or to perform unconventional extractions for which a dedicated NLP system does not exist.

## 5. Conclusions

This study explored the ability of pre-trained language models

(LLMs) to extract information from clinical case reports in raw PDF file. The results showed that InstructGPT effectively extracted data on pediatric foreign body injuries, without requiring task-specific fine-tuning. InstructGPT was found to be more flexible and user-friendly compared to other NLP systems. Its advantages include not requiring data preprocessing, programming skills, or extensive training data. These features make it an accessible and efficient tool for data extraction.

## 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.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2024.108326.

## References


[1] K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S.F. Jones, R. Forshee, M. Walderhaug, T. Botsis, Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review, J. Biomed. Inform. 73 (2017) 14–29, https://doi.org/10.1016/j.jbi.2017.07.012.

[2] D.A. Springate, R. Parisi, I. Olier, D. Reeves, E. Kontopantelis, rEHR: an R package for manipulating and analysing electronic health record data, PLoS One 12 (2017) 1–25, https://doi.org/10.1371/journal.pone.0171784.

[3] S. Li, G. Yu, ctrdata: clinical trials data, 2021. https://CRAN.R-project.org/package=ctrdata.

[4] H.L. Weeks, C. Beck, E. McNeer, M.L. Williams, C.A. Bejan, J.C. Denny, L. Choi, medExtractR: a targeted, customizable approach to medication extraction from electronic health records, J. Am. Med. Inform. Assoc. 27 (2020) 407–418, https://doi.org/10.1093/jamia/ocz207.

[5] M. Hinchcliff, E. Just, S. Podlusky, J. Varga, R.W. Chang, W.A. Kibbe, Text data extraction for a prospective, research-focused data mart: implementation and validation, BMC Med. Inform. Decis. Mak. 12 (2012) 106, https://doi.org/10.1186/1472-6947-12-106.

[6] L.W. D'Avolio, A.A.T. Bui, The clinical outcomes assessment toolkit: a framework to support automated clinical records–based outcomes assessment and performance measurement research, J. Am. Med. Inform. Assoc. 15 (2008) 333–340, https://doi.org/10.1197/jamia.M2550.

[7] M.A. Murtaugh, B.S. Gibson, D. Redd, Q. Zeng-Treitler, Regular expression-based learning to extract bodyweight values from clinical notes, J. Biomed. Inform. 54 (2015) 186–190, https://doi.org/10.1016/j.jbi.2015.02.009.

[8] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, (2020). 10.48550/ARXIV.2005.14165.

[9] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, (2019). 10.48550/arXiv.1810.04805.

[10] Microsoft Announced Its First Customer Product Features Powered By GPT-3 and @Azure, The AI Blog, 2021. https://blogs.microsoft.com/ai/from-conversation-to-code-microsoft-introduces-its-first-product-features-powered-by-gpt-3/ (accessed December 16, 2022).

[11] OpenAI Codex, OpenAI. (2021). https://openai.com/blog/openai-codex/(accessed December 16, 2022).

[12] A Robot Wrote This Entire article. Are you Scared yet, Human? The Guardian, 2020. https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3 (accessed December 16, 2022).

[13] S. Nath, A. Marie, S. Ellershaw, E. Korot, P.A. Keane, New meaning for NLP: the trials and tribulations of natural language processing with GPT-3 in ophthalmology, Br. J. Ophthalmol. 106 (2022) 889, https://doi.org/10.1136/bjophthalmol-2022-321141.

[14] D.M. Korngiebel, S.D. Mooney, Considering the possibilities and pitfalls of Generative Pre-trained Transformer 3 (GPT-3) in healthcare delivery, npj Digit. Med. 4 (2021) 93, https://doi.org/10.1038/s41746-021-00464-x.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, (2022). 10.48550/ARXIV.2203.02155.

[16] H. Ocagli, D. Azzolina, S. Bressan, D. Bottigliengo, E. Settin, G. Lorenzoni, D. Gregori, L. Da Dalt, Epidemiology and trends over time of foreign body injuries in the pediatric emergency department, Children 8 (2021) 938, https://doi.org/10.3390/children8100938.

[17] J. Singer-Vine, pdfplumber, (2022). https://github.com/jsvine/pdfplumber (accessed December 17, 2022).

[18] OpenAI, (n.d.). 2024 https://beta.openai.com/.

[19] Python Software Foundation, Python: A dynamic, Open Source Programming Language, Python Software Foundation, Arlington, VA, 2020. https://www.python.org/.

[20] R. Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2020. https://www.R-project.org/.

[21] R. Bill, S. Pakhomov, E.S. Chen, T.J. Winden, E.W. Carter, G.B. Melton, Automated extraction of family history information from clinical notes, AMIa Annu Symp. Proc. 2014 (2014) 1709–1717.

[22] L.W. D'Avolio, M.S. Litwin, S.O. Rogers, A.A.T. Bui, Facilitating clinical outcomes assessment through the automated identification of quality measures for prostate cancer surgery, J. Am. Med. Inform. Assoc. 15 (2008) 341–348, https://doi.org/10.1197/jamia.M2649.