Developing an Interactive Agent for Blind and Visually Impaired People

Vincent Stragier
vincent.stragier@umons.ac.be
Numediart Institute, ISIA Lab, Faculty of Engineering, University of Mons, Mons
Mons, Hainaut, Belgium

Omar Seddati
omar.seddati@umons.ac.be
Numediart Institute, ISIA Lab, Faculty of Engineering, University of Mons, Mons
Mons, Hainaut, Belgium

Thierry Dutoit
thierry.dutoit@umons.ac.be
Numediart Institute, ISIA Lab, Faculty of Engineering, University of Mons, Mons
Mons, Hainaut, Belgium

ABSTRACT

The aim of this project is to create an interactive assistant that incorporates different assistive features for blind and visually impaired people. The assistant might incorporate screen readers, magnifiers, voice synthesis, OCR, GPS, face recognition, and object recognition among other tools. Recently, the work done by OpenAI and Be My Eyes with the implementation of GPT-4 is comparable to the aim of this project. It shows the development of an interactive assistant has become simpler due to recent developments in large language models. However, older methods like named entity recognition and intent classification are still valuable to build lightweight assistants. A hybrid solution combining both methods seems possible, would help to reduce the computational cost of the assistant, and would facilitate the data collection process. Despite being more complex to implement in a multilingual and multimodal context, a hybrid solution has the potential to be used offline and to consume less resources.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

accessibility; assistive technology; visually impaired; blind; face recognition; OCR; object recognition; BLOOMZ; GPT-4; OpenAI; Open Source

ACM Reference Format:
Vincent Stragier, Omar Seddati, and Thierry Dutoit. 2023. Developing an Interactive Agent for Blind and Visually Impaired People. In *ACM International Conference on Interactive Media Experiences (IMX '23), June 12–15, 2023, Nantes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3573381.3596471

1 INTRODUCTION

The population of visually impaired people in Belgium is about one hundredth of the country's population[25]. Among which about one tenth is legally blind. Because people tend to live longer, it is speculated that this population will increase, since visual impairment tends to appear at an older age[8]. In the current context, it means, newly visually impaired people need to learn how to use the accessibility tools (including digital ones) without any previous knowledge on the usage of these tools. For example, if to gain some autonomy they need to use a smartphone, and it's accessibility features (TalkBack[9] on Android and VoiceOver[7] on iOS), they must learn a set of non fully intuitive gestures. This makes the interface hard to learn, moreover, when their cognition is affected by their health condition.

Indeed, some vocal assistants already exist (e.g. Siri, Alexa, Google Assistant, Cortana, etc.), but they are mostly not conceived with accessibility in mind. However, a lot of accessibility tools exist, and if the user knows how to handle them, they help to read some text, detect objects, recognize people, etc. Other specialized tools are implemented in hardware solutions (e.g. electronic reader, magnifying glass, etc.), making them nearly *unupgradable* and expensive, since produced at a low scale. For the most efficient tools, the expensiveness of those solutions is an issue. Moreover, in Belgium, the visually impaired and blind people can only purchase one of those tools, rendering the selection of such a tool delicate and crucial.

Therefore, building an interactive assistant for blind and visually impaired people is interesting. Such an assistant might incorporate screen readers, magnifiers, voice synthesis, OCR, GPS, face recognition, and object recognition among other tools. With the current advance in large language models, it is easier to implement such an assistant. However, these models are computer intensive (and thus power hungry). Implementing a hybrid solution cascading a more traditional method (e.g., using intent classification and named entity recognition) and a large language model might be a convenient solution to reduce the computational cost of the assistant. Moreover, it would facilitate the data collection process. Despite being more complex to implement in a multilingual and multimodal context, a hybrid solution has the potential to be used offline and to consume less resources.

2 RELATED WORK

The most notorious work so far in this regard is the virtual volunteer developed by OpenAI and Be My Eyes[11]. Their project uses GPT-4[10] to answer questions from visually impaired people. The model uses multiple modalities (at least text and image) to answer the user's requests. The app allows accessing a virtual volunteer that answers to specific questions. The strength of this project is the use of a large language model that interfaces the different modalities. It makes the virtual volunteer robust when it comes to understanding

IMX '23, June 12–15, 2023, Nantes, France

ACM ISBN 979-8-4007-0028-6/23/06.
https://doi.org/10.1145/3573381.3596471

the user requests. However, these models are really large, which makes them dependent on remote servers. This can be an issue for the visually impaired people, since from time to time they can find themselves disconnected from the internet.

Granted, other assistive tools exist. For example, the NVDA screen reader[3] is a free and open source screen reader for Microsoft Windows operating systems. It is developed by NV Access, a non-profit organization. You can also use the JAWS[1] screen reader, a commercial screen reader. These solutions are not designed to be used on smartphones, and are not conversational. On smartphones which are mainly running on Android or iOS, TalkBack and VoiceOver are the default accessibility tools. Again, they are not conversational tools, and not always intuitive to use. Despite this fact, they are, by default of better solutions, the most used tools by the visually impaired people, since they help them to gain some autonomy and access all the other apps.

Specialized applications (e.g., Seeing AI[6], Lookout[5], Envision AI[4], Be My Eyes[19], OOrion[12], Blindsquare[23], etc.), like the previously mentioned Be My Eyes, which allows connecting visually impaired people with sighted volunteers, are also available. Some mobile applications are focused on specific tasks, like OCR, which consists in reading text from a picture or object detection from a picture. In most cases, if not all, it is needed to access these apps via the accessibility tools (i.e., VoiceOver or TalkBack). This makes the usage of these tools cumbersome, and again not really intuitive.

The specialized applications often use computer vision to help visually impaired people. This shows the progress in this field, but does not solve the problem of the usability of the accessibility tools. Indeed, models like YOLO and SSD are used to detect objects in images. Google Tesseract[2] is really efficient to performing OCR. RetinaFace[13] allows detecting faces on images, while FaceNet[17] is used to recognize faces. All these models are virtually monomodal, which induce an interface must be built to use them. GPT-4 on the other hand is multimodal (text inputs, image inputs and text outputs)[26] and thus can more easily be used in a conversational interface. However, unlike the Virtual Volonteer, our goal is to have smaller models that can be used with less resources, and thus used offline most of the time. For more complexes or new tasks, the assistant can use a remote server to perform the task via larger models.

3 METHODOLOGY

The aim is to build a prototype of an efficient conversational interface that allows the visually impaired people to use the accessibility tools. The first step is to find which tools are the most used by the visually impaired people. The second step is to build a prototype of a conversational interface that allows the user to use the accessibility tools. The third step is to evaluate the interface, and to see if it is efficient.

Since it is not possible to directly find which features must be implemented, the order of the two first steps is not fixed, because, it may be necessary to rethink the list of features while working on the prototype. To find which tools exist and are used, various meetings have been held with experts, visually impaired people, and people working in the field. The goal was to find which tools are the most used, and which tools are the most efficient. Due to the extent of such an interactive assistant, the prototype is not yet fully implemented, and only a reduced set of tools will be integrated. The current focus is on facial recognition on 2-dimensional RGB images.

A first list of use cases that apply to both blind and visually impaired people has been produced for the prototype. The use cases are the following:

- **navigation** the user can ask to navigate to a specific location and the assistant will open the appropriate application (e.g., Google Maps, Apple Maps, etc.) and will guide the user to the destination.
- **face recognition** the user can ask to add or remove a face in its local database. She/he can also check if someone is in the room and if so, the assistant will propose to navigate to the person. The assistant can also produce a list of the people in the room.
- **age and sex recognition** if asked, the assistant will try to estimate the sex and the age of the person in front of the camera.
- **emotion recognition** if asked, the assistant will try to estimate the emotion of the person in front of the camera.
- **object recognition** using the camera, the assistant will try to detect objects in the room and will produce a list of the objects. The user can ask if a specific object is in the room and to be guided to it.
- **color recognition** the user can ask the color of a specific object. The user can ask for the dominant color of the room. The user can ask for the color of the visible objects in the room.
- **optical character recognition (OCR)** the user can ask to read a text from a specific object (a sign, a book, a display, etc.). The user can ask if some texts are present in the room, and the assistant will produce a list of objects containing text.
- **money recognition** the user can ask to recognize the value of a specific object (a coin, a banknote, etc.).
- **environment recognition** the user can ask for a description of the environment (similar to captioning models that will produce a summary of what is present in the field of view of the camera).

It is to be noted that some of these use cases are interdependent. For instance, face recognition, sex, age, and emotion recognition are all dependent on a face detection system. The same goes for object recognition, color recognition OCR, and money recognition, that all depend on an object detection system.

Another important point about the face related models (age, sex, emotion, etc.), is the fact that they are not always accurate[35]. Emotion recognition alone is affected by this issue, since facial expressions can be identical for different emotions in different context, and datasets include ethnic bias, etc.[14, 16]

3.1 Prototype of interactive assistant

In order to build an interactive assistant, it is mandatory to explore the various natural language processing (NLP) techniques that exist to build a conversational interface and the generic architecture of such an interface (see figure 1).

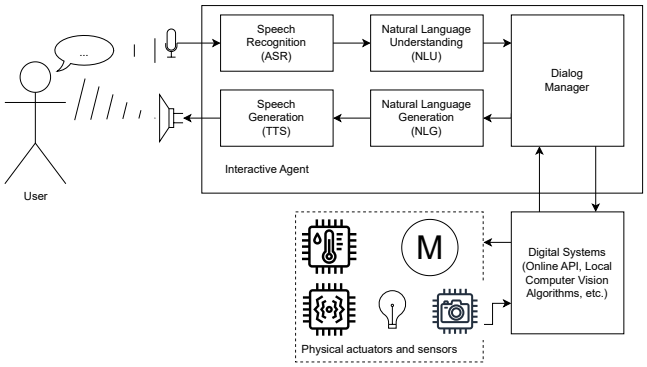

Figure 1: Generic architecture of a conversational interface[22].

3.1.1 Natural language processing techniques. Former techniques involved the use of a rule based system. These systems use a grammar to parse the user's input and then produce the appropriate response. The grammar is usually handcrafted, and thus the system is not really flexible (lack of robustness, and lack of scalability).

More recent NLP techniques rely on machine learning. For instance, models that are used to perform intent classification can be trained on specialized dataset (user requests as inputs and intents as outputs). This allows to roughly understand the intention of the user and in a second time to extract the entities (e.g., the location, the object, the name of a person, etc.) from the user's input. The entities and intention can then be used to perform the appropriate action.

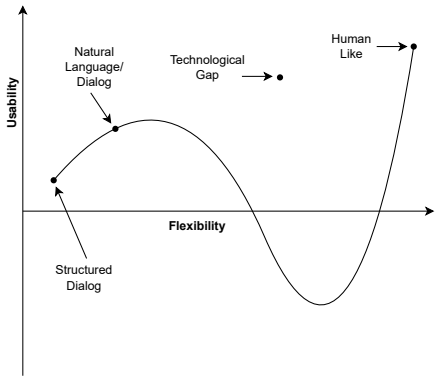

Figure 2: Approximative evolution of the usability of a spoken language interface in function of its flexibility[24].

The main difficulty lay in the design of the dataset, since it is application specific and thus depends on the implemented features. Furthermore, when trying to increase the flexibility of such systems, the usability tends to decrease (see figure 2). This may be related to the fact that increasing the flexibility is usually linked to the increase of possible user's inputs. Thus the system will be more prone to errors.

3.1.2 Large language models. The newest large language models are different. They can be considered as autocorrect/autofill models on steroid. They are alone *only* able to predict the likeliest next words. Using them as conversational agent is in some sense a hack. It may require to fine-tune the model for this specific task (e.g., ChatGPT uses/used gpt-3.5-turbo a model fine-tuned via reinforcement learning especially for conversational tasks[27]). Before, it was already possible to use a few-shot technique to make GPT-3 act as a conversational AI. The idea is to provide a few examples of conversation to the model, and to let it *learn* from them. This is a powerful technique that has been named *prompt engineering* and is still used with ChatGPT. The given prompt provides some context to the model and will make it reply in a predictable, defined manner (e.g., it could be asked to the model to take the role of a specialized assistant for visually impaired people).

As for previous NLP techniques, large language models are able to extract the intent and entities from the user's input[15]. Furthermore, they can directly generate the appropriate function code to trigger some task if the prompt is well-designed.

3.1.3 Hybrid approach. Another approach would be a hybrid system with a more classic deep learning approach to first handle the user input. Then, if the input is too complex, pass the input to a large language model to generate the appropriate response. The main advantage of the method is that the system is more robust to exotic input and more efficient than a pure large language model. The main drawback is that it requires to train neural networks for each language and task. However, the large language model can help to train the neural network (e.g., by assisting the data mining for the dataset). The hybrid approach can help to create a working Wizard of Oz[22] with the large language model, which will facilitate the design of specialized datasets.

Remains the implementation of a prototype, which can easily be achieved using Python's modules like Flask, pyttsx3 and some APIs, like the one of OpenAI, BLOOM[38] or BLOOMZ. A screenshot of the prototype's interface is shown in figure 3.

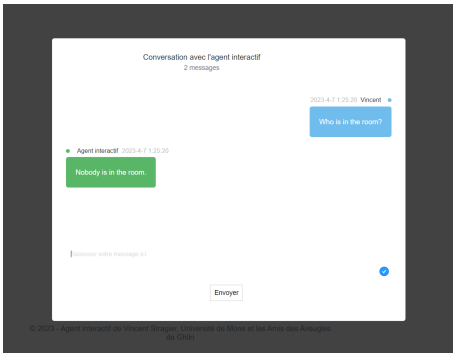

Figure 3: A screenshot of the prototype's interface.

On the technical side, the current prototype is a web app connected to a web server (which can run locally) and that calls the API of the large language model (BLOOMZ in this case). The particularity of BLOOMZ is that it runs on an open source distributed system (Petals). The figure 4 below shows the architecture of the prototype.

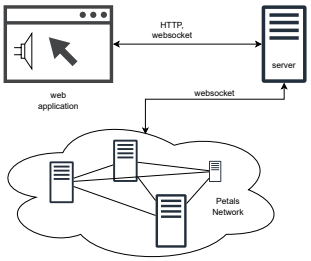

Figure 4: Abstractive technical view of the prototype.

In the current state, the interface is not functional, since no module is implemented to handle the user's inputs. At the time, the reply is mocked.

3.2 Prototype of facial recognition

Being able to recognize faces is socially important (e.g., to initiate a conversation). The current state of facial recognition allows recognizing faces in real time.

The first technique of facial recognition required to manually annotate specific features on the faces, then a computer was used to compute the distances between some features and compare them between faces[37]. Later, the annotation of facial features was automated, but the reliability of the system was variable[18]. These techniques were only working with portraits, then, PCA analysis (eigenfaces)[29], allowed to detect faces.

As shown in the previous paragraph, face detection is needed to perform face recognition. One of the efficient techniques used was the Viola-Johns algorithm[36] around 2001. More than one decade later, neural networks are used like YOLO[39], RetinaFace[13], MTCNN[20], SSD[21], Dlib[34] or MediaPipe[32]. The most accurate are RetinaFace followed by MTCNN[33].

Nowadays, for face recognition, pretrained models are used to embed the faces in a vector space. The faces are then compared using a distance metric (e.g., cosine similarity, Euclidean distance, etc.). The distance between two faces is then used to determine if the faces are the same or not. Similarly to the face detection task, multiple models exist. The current state-of-the-art models seem to be Facenet-512 (upscaled version of Facenet-128[30]) and VGG-Face[28].

Furthermore, as stated above, other models can be used for age, sex, ethnicity, and emotion (facial expression) estimation. Here, emotion estimation could be interesting, however, it seems to be far from being accurate (the accuracy on the train set is about 98.81 % on the train set and 56.31 % on the test set) as shown in table 1. The accuracy on the test set is not very high, but it is still better than random (14.29 %). The model used is a CNN with 3 convolutional layers, and a fully connected neural network. The model was trained on the FER2013 dataset[31]. The netbook used to retrain the model is accessible via this link: https://gist.github.com/Vincent-Stragier/1562c159b2e0e7cda804ce153f1a83d4.

Implementing a facial recognition system is not trivial, because, it is needed to create some sort of database (stored, either locally or on a server). Here, it has been decided for technical reason to keep the database on the server, storing some sort of ID, multiple

Table 1: Confusion matrix for facial expression recognition on FER2013[31].

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	200	2	57	45	80	14	49
Disgust	20	21	5	4	5	0	1
Fear	64	2	179	49	90	38	74
Happy	33	1	26	694	59	19	63
Sad	74	2	59	74	303	17	124
Surprise	15	0	37	21	20	301	21
Neutral	60	1	44	84	108	7	303

embeddings (depending on the used models and the number of images of the user), the faces used to compute the embeddings and the user's name. The figure 5 shows the technical view of the prototype.

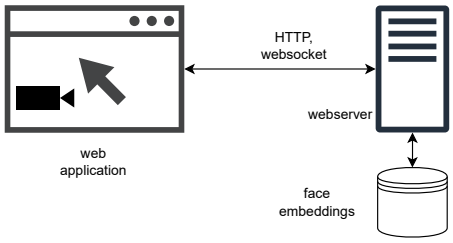

Figure 5: Technical view of the prototype of facial recognition system.

The current prototype allows selecting different actions, such as adding a new user (which only sends the current webcam frame to the server). It is able to detect faces using the selected model and then give an estimation of the facial expression, sex, age and ethnicity if requested. The figure 6 shows a screenshot of the prototype's interface. The source code is available on GitHub at https://github.com/Vincent-Stragier/webfacetools.

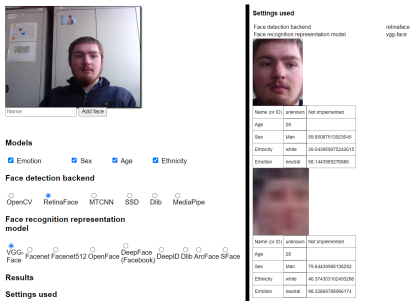

Figure 6: Two screenshots of the prototype's interface. On the left side, a live view of the camera is showed and various settings are available. On the right, two persons have been detected, and the interface shows the results of all the selected models.

4 CONCLUSION

The current state of the technologies shows that building a working interactive assistant is possible. The work of OpenAI and Be My Eyes is a good example of the potential of the technology. However, the current large language models are not the best fit for such a task, since they are power angry and sometime completely instable. Granted, they could be used to build less flexible but more reliable systems using current spoken language understanding systems. Eventually, open sourcing such a system could help to improve the current state of assistive technologies.

5 FUTURE WORK

In the next years, the goal is to continue to build a prototype of an interactive assistant for blind and visually impaired people. The prototype will be improved using the technologies described above. The prototype will be tested with blind and visually impaired people to evaluate the usefulness of the prototype. It will be open sourced to allow other people to improve it. Also, a special focus will be made on open source large language models like BLOOMZ or LLaMA.

ACKNOWLEDGMENTS

This research is funded by the European Regional Development Fund (ERDF). Vincent Stragier is funded through a PhD grant from the Œuvre fédérale Les Amis des Aveugles et Malvoyants ASBL-The Friends of the Blind and Visually Impaired Federal Charity-, Ghlin, Belgium.

REFERENCES

[1] [n. d.]. JAWS - Logiciel de lecture d'écran avec retour vocale et braille. https://sensotec.be/fr/produit/jaws/.
[2] 2012. Various Documents Related to Tesseract OCR. https://tesseract-ocr.github.io/docs/.
[3] 2017. NV Access.
[4] 2022. Envision App. https://www.letsenvision.com/app.
[5] 2022. Lookout - Vision Assistée – Applications Sur Google Play.
[6] 2022. Seeing AI App from Microsoft. https://www.microsoft.com/en-us/ai/seeing-ai.
[7] 2023. Activer VoiceOver et s'entraîner à utiliser les gestes sur l'iPhone. https://support.apple.com/fr-fr/guide/iphone/iph3e2e415f/ios.
[8] 2023. Cécité et déficience visuelle. https://www.who.int/fr/news-room/fact-sheets/detail/blindness-and-visual-impairment.
[9] 2023. Get Started on Android with TalkBack - Android Accessibility Help. https://support.google.com/accessibility/android/answer/6283677?hl=en-GB.
[10] 2023. GPT-4. https://openai.com/product/gpt-4.
[11] 2023. Introducing Our Virtual Volunteer Tool for People Who Are Blind or Have Low Vision, Powered by OpenAI's GPT-4. https://www.be-myeyes.com/blog/introducing-be-my-eyes-virtual-volunteer.
[12] 2023. OOrion. https://apps.apple.com/fr/app/oorion/id1567957213.
[13] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. RetinaFace: Single-stage Dense Face Localisation in the Wild. https://doi.org/10.48550/arXiv.1905.00641 arXiv:1905.00641 [cs]
[14] P. Drozdowski, C. Rathgeb, A. Dantcheva, N. Damer, and C. Busch. 2020. Demographic Bias in Biometrics: A Survey on an Emerging Challenge. *IEEE Transactions on Technology and Society* 1, 2 (June 2020), 89–103. https://doi.org/10.1109/TTS.2020.2992344 arXiv:2003.02488 [cs] Comment: 15 pages, 3 figures, 3 tables. Submitted to IEEE Transactions on Technology and Society. Update after first round of peer review.
[15] Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S. Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. 2022. Structured Information Extraction from Complex Scientific Text with Fine-Tuned Large Language Models. arXiv:2212.05238 [cond-mat]
[16] Damien Dupré, Eva G. Krumhuber, Dennis Küster, and Gary J. McKeown. 2020. A Performance Comparison of Eight Commercially Available Automatic Classifiers for Facial Affect Recognition. *PLOS ONE* 15, 4 (April 2020), e0231968. https://doi.org/10.1371/journal.pone.0231968
[17] Charles F. Jekel and Raphael T. Haftka. 2018. Classifying Online Dating Profiles on Tinder Using FaceNet Facial Embeddings. https://doi.org/10.48550/arXiv.1803.04347 arXiv:1803.04347 [cs, eess, stat] Comment: 6 pages, 7 figures.
[18] Takeo Kanade. 1977. *Computer Recognition of Human Faces.* Birkhäuser Basel, Basel. https://doi.org/10.1007/978-3-0348-5737-6
[19] Parminder Kaur, Mayuri Ganore, Rucha Doiphode, Ashwini Garud, and Tejaswini Ghuge. 2017. *Be My Eyes : Android App for Visually Impaired People.* https://doi.org/10.13140/RG.2.2.12307.48164
[20] Edgar Kaziakhmedov, Klim Kireev, Grigorii Melnikov, Mikhail Pautov, and Aleksandr Petiushko. 2019. Real-World Attack on MTCNN Face Detection System. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON).* 0422–0427. https://doi.org/10.1109/SIBIRCON48586.2019.8958122
[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. Vol. 9905. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2 arXiv:1512.02325 [cs] Comment: ECCV 2016.
[22] Birgit Lugrin, Catherine Pelachaud, and David Traum (Eds.). 2021. *The Handbook on Socially Interactive Agents: 20 Years of Research on Embodied Conversational Agents, Intelligent Virtual Agents, and Social Robotics Volume 1: Methods, Behavior, Cognition* (first ed.). ACM, New York, NY, USA. https://doi.org/10.1145/3477322
[23] MIPsoft. 2023. BlindSquare. https://apps.apple.com/fr/app/blind-square/id500557255.
[24] Roger K. Moore. 2019. A 'Canny' Approach to Spoken Language Interfaces. arXiv:1908.08131 [cs] Comment: Presented at the CHI 2019 Workshop on Mapping Theoretical and Methodological Perspectives for Understanding Speech Interface Interactions, 4-9 May 2019, Glasgow, UK.
[25] Rafal Naczyk. 2020. Eqla. https://eqla.be.
[26] OpenAI. 2023. GPT-4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774 arXiv:2303.08774 [cs] Comment: 100 pages.
[27] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. https://doi.org/10.48550/arXiv.2203.02155 arXiv:2203.02155 [cs]
[28] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. In *Procedings of the British Machine Vision Conference 2015.* British Machine Vision Association, Swansea, 41.1–41.12. https://doi.org/10.5244/C.29.41
[29] Alex P Pentland. [n. d.]. Face Recognition Using Eigenfaces. ([n. d.]).
[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 815–823. https://doi.org/10.1109/CVPR.2015.7298682 arXiv:1503.03832 [cs] Comment: Also published, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015.
[31] Sefik Serengil. 2018. Facial Expression Recognition with Keras.
[32] Sefik Serengil. 2022. Deep Face Detection with Mediapipe.
[33] Sefik Ilkin Serengil. 2023. Deepface.
[34] S. Sharma, Karthikeyan Shanmugasundaram, and Sathees Kumar Ramasamy. 2016. FAREC — CNN Based Efficient Face Recognition Technique Using Dlib. In *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT).* 192–195. https://doi.org/10.1109/ICACCCT.2016.7831628
[35] SITNFlash. 2020. Racial Discrimination in Face Recognition Technology.
[36] Paul Viola and Michael J. Jones. 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision* 57, 2 (May 2004), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
[37] Relly Victoria Virgil Petrescu. 2019. Face Recognition as a Biometric Application. *Journal of Mechatronics and Robotics* 3, 1 (Jan. 2019), 237–257. https://doi.org/10.3844/jmrsp.2019.237.257
[38] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle

Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M. Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A. Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S. Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. https://doi.org/10.48550/arXiv.2211.05100 arXiv:2211.05100 [cs]

[39] Wang Yang and Zheng Jiachun. 2018. Real-Time Face Detection Based on YOLO. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*. 221–224. https://doi.org/10.1109/ICKII.2018.8569109

Received 7 April 2023