QUA4I: Question Answering for the Industry 4.0 Domain. An Application of Intelligent Virtual Assistants

Iñigo López-Riobóo-Botana[1], Dana Gallent-Iglesias[1] and Sonia Gonzalez-Vázquez[1]

[1]Instituto Tecnológico de Galicia - ITG - Centro Tecnológico Nacional, Cantón Grande 9, Planta 2, 15003, A Coruña, España

Abstract

The recent advancements with LLMs (Large Language Models) have led to many natural language applications solving different NLP (Natural Language Processing) tasks. In such systems, both NLG (Natural Language Generation) and NLU (Natural Language Understanding) play a crucial role. The notion of instruction-based LLMs caused an important impact in the development of chatbot-styled NLP applications. In the past months, we have seen an incredible amount of LLMs making use of this idea to solve tasks including, but not limited to, QA (Question Answering), information extraction, intent recognition, dialogue or language generation. Whereas these systems have very good generalisation capabilities, in which the AGI (Artificial General Intelligence) idea has spread to the NLP research field, we still lack of good domain adaptation techniques for tailored ADI (Artificial Domain Intelligence) systems. Apart from prompt engineering, we cannot fully control the outputs of this new chatbot-styled LLMs to obtain our desired output given a specific domain.

In the context of the CEL.IA network, we present **QUA4I** (**QU**estion **A**nswering **for** the **I**ndustry 4.0), a chatbot-oriented application or IVA (Intelligent Virtual Assistant) for the industry 4.0 domain, mixing the NLU and NLG techniques using the Rasa chatbot framework. We designed a custom demo for question answering and information extraction about the industry 4.0 topic, including a dialogue system which can also generate automatic responses in natural language. We included both ASR (Automatic Speech Recognition) and TTS (Text To Speech) modules, so we can also interact with the bot using spoken language in Spanish.

Keywords

question answering, information extraction, IVA, NLU, NLG, ASR, TTS, industry 4.0, Rasa, chatbot

1. Introduction

In recent years, we have seen an incredible amount of new chatbot-oriented applications [1, 2]. Chatbots and IVAs (Intelligent Virtual Assistants) can automate tasks without human intervention, saving a lot of time and effort with automation. Chatbots can be applied to several domains, including healthcare, retail, banking, among others. For example, these systems can be integrated in websites or VR (Virtual Reality) environments. We can provide chatbots with more interfaces in addition to text, like human voice, so we can interact with them in different ways. Moreover, the great advancements in the NLP (Natural Language Processing) field have led to a very good understanding of the language and incredible improvements in generation capabilities, thanks to LLMs (Large Language Models). LLMs are decoder-only transformer models, such as GPT-3 [3] (OpenAI), BLOOM [4] (open source), PaLM [5] (Google) or LLaMA [6] (Meta). The main task of these models is to predict the next token given a sequence, but we have seen a drift towards dialogue capabilities recently, which is a clear improvement for instruction-based applications. Recent and well-known examples are DialoGPT [7] (Microsoft), InstructGPT [8] and ChatGPT [9] (OpenAI) or LaMDA [10] and Bard [11] (Google). LLMs have grown to the point that they can mimic or even outperform human answers in some NLP tasks [12]. These breakthroughs are possible thanks to the continuous and competitive scaling of deep learning models [13] and the proposal of new architectures [14].

In Section 1.1, we present our motivation to carry out this project. In Section 1.2 we enumerate our main contributions. In Section 2, we depict the architecture and methodology followed for this project, describing its limitations in Section 3. Finally, we conclude with some ideas and future work in Section 4.

1.1. Motivation

For domain-specific and ad hoc conversational agents (i.e., to fulfil special needs of a use case), we need to adjust and control the chatbot output thoroughly. The chatbot-oriented LLMs fit in AGI (Artificial General Intelligence) contexts and they can be somewhat "fine-tuned" with prompt engineering [15, 16, 17, 18], but this is not enough

SEPLN-PD 2023: Annual Conference of the Spanish Association for Natural Language Processing 2023: Projects and System Demonstrations

ibotana@itg.es (I. López-Riobóo-Botana); dgallent@itg.es (D. Gallent-Iglesias); sgonzalez@itg.es (S. Gonzalez-Vázquez)
https://www.linkedin.com/in/%C3%AD%C3%B1igo-luis-l%C3%B3pez-riob%C3%B3o-botana-4a43001a2/ (I. López-Riobóo-Botana); https://www.linkedin.com/in/dana-gallent-iglesias-7a0038247/ (D. Gallent-Iglesias); https://www.linkedin.com/in/phd-sonia-gonz%C3%A1lez-v%C3%A1zquez-38b14a8b/ (S. Gonzalez-Vázquez)
0000-0002-7310-0702 (I. López-Riobóo-Botana); 0009-0006-2782-2567 (S. Gonzalez-Vázquez)

for guided ADI (Artificial Domain Intelligence) systems. Adaptation and customisation of these chatbots is still a work-in-progress [19, 20].

In the context of the CEL.IA network[1], we propose a domain-specific IVA (Intelligent Virtual Assistant) for the Industry 4.0 domain. We mixed the concepts of NLU (Natural Language Understanding) and NLG (Natural Language Generation) using the Rasa chatbot framework [21]. In this paper, we describe our first demo version for the question answering and information extraction NLP tasks, including a dialogue system which can also generate answers in natural language. We considered both ASR (Automatic Speech Recognition) and TTS (Text To Speech) modules to enhance the communication interfaces with the chatbot.

1.2. Contributions

Our main contributions are as follows:

- **We implemented a domain-specific IVA system with Rasa framework** for NLU and dialogue. We solve the QA (Question Answering) and information extraction NLP task in the context of Industry 4.0.
- **We avoided relying on third-party APIs and deployed our own on-premise models**. These models are currently available through REST services to improve the capabilities of the chatbot.
- **We provide both voice and text interfaces**, so that we can communicate with spoken language in Spanish with the chatbot, as well as listen to the answers.
- **We handle out-of-scope questions with automatic dialogue generation** using our own GPT-based Spanish model for chitchat, fine-tuning a DialoGPT-2 [7] model[2].

2. Methodology

We designed and implemented the following modules:

- **NLU and dialogue modules**: These are the core modules for intent recognition and dialogue flow, respectively. The Rasa framework is in charge of this management.
- **QA module**: This module is in charge of providing the answer according to an user information need, following an information extraction task with a QA approach. We manage our own document store in our servers, containing information about the Industry 4.0 domain in several files. This knowledge base, which basically contains descriptions and definitions, can be extended if new information needs arise. This module was implemented as a REST service.
- **NLG module**: This module is in charge of generating automatic responses when the answer is neither known by Rasa framework nor by the QA system. In such situation, we implemented a single turn dialogue or chitchat mechanism with the chatbot, trying to preserve a logical conversation flow. This module was also deployed as a REST endpoint.
- **ASR module**: The speech recognition module is implemented in our custom NLP services, enabling us to transcribe from audio to text and then send it back to the chatbot. After that, it is processed in the NLU module. We based our implementation on the Spanish *stt_es_citrinet_512* model, publicly available in the NVIDIA NeMo toolkit[3].
- **TTS module**: When fetching the chatbot response, we transform the text output to human voice again so that we can easily interact with the IVA system. This service is also implemented in our custom NLP services. We based our implementation on the Spanish *glow-speak:es_tux* model, publicly available for production-ready environments using the OpenTTS framework[4].

The general diagram of the IVA system is depicted in Figure 1. In Section 2.1, we study in more detail the NLU and dialogue implementations for the chatbot. In Section 2.2, we present our approach for the question answering functionality of the chatbot. In Section 2.3, we describe our language generation method for the chitchat situations. We provide an example conversation in Figure 2.

2.1. NLU and dialogue module

These two modules are the principal components of Rasa open source [21].

1. **The NLU subsystem** is in charge of the intent recognition NLP task. Rasa projects follow a data-driven approach, providing several data files with text samples for each intent and configuration files to adjust the pipelines for model training.
2. **The dialogue module** relies on a combination of a rule-based system and a "user stories" mechanism to infer the next step in the conversation. These actions can be **(1) direct chatbot**

[1] https://www.redcelia.es/

[2] https://huggingface.co/ITG/DialoGPT-medium-spanish-chitchat

[3] https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_es_citrinet_512

[4] https://github.com/synesthesiam/opentts#voices

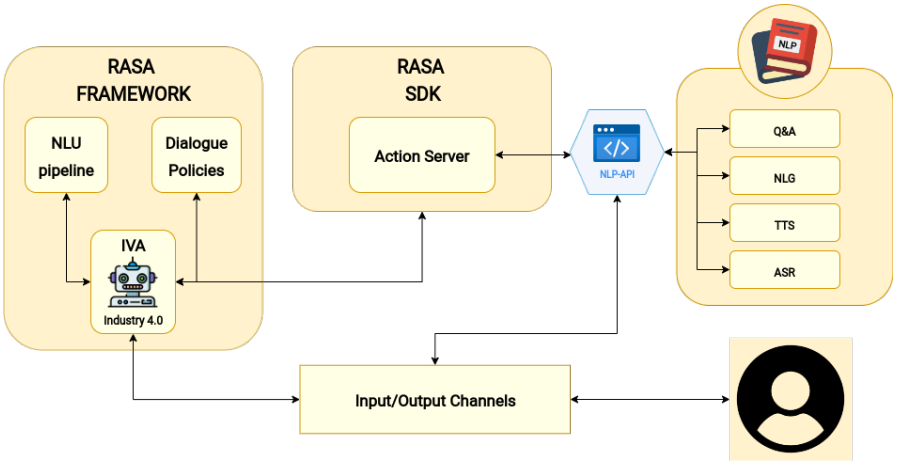

Figure 1: Architecture of the IVA chatbot system for the Industry 4.0 domain, with all the corresponding modules. The **Rasa framework is in charge of the NLU (intent recognition) task**. For each user information need detected by Rasa, the **dialogue subsystem is responsible for providing the answer**. However, **when the user intent is unrecognisable by Rasa, the SDK action server will automatically make use of our custom NLP service for QA**. If the answer cannot be fetched from the document store with enough confidence, **the NLG service takes action for automatic response generation (out-of-scope chitchat)**. The communication with the IVA system is through the **input/output channels, using not only text interface but also voice**, thanks to our custom ASR and TTS services.

responses (when we already fulfilled the information need from Rasa) or **(2) delegations in the independent SDK action server to access third-party services** (i.e., our aforementioned custom NLP services).

In order to train our Rasa model for intent recognition, we used the following Rasa components [22]:

1. **Tokenizer**: Rasa component in charge of splitting each sentence into tokens or words. We used the simple *WhiteSpaceTokenizer* to get the tokens splitting with white spaces.
2. **Feature extractor**: Rasa component in charge of feature engineering to transform the corresponding tokens into numerical vector representations. We made use of feature extractors from pre-trained transformer-encoder backbones. Specifically, we used the deep word embeddings from the last encoder layer of a mDistilBERT transformer model [23]. We used the multilingual version with support for Spanish, with a total size of 134M parameters (compared to 177M parameters for mBERT-base [24]). On average mDistilBERT is twice as fast as mBERT-base [25].
3. **Intent classifier**: Rasa component in charge of the intent recognition NLP task. We used the DIET classifier from the Rasa framework authors [26]. DIET is a multi-task modular transformer architecture that handles both intent classification and entity recognition together. It provides the ability to plug and play various pre-trained embeddings like BERT (and variants), GloVe, ConveRT, among others [27].
4. **Fallback classifier**: This Rasa component is in charge of triggering the default intent when the intent prediction from the DIET classifier does not have a confidence above a pre-specified threshold. If so, the default intent will diverge from the regular conversation, following a custom action. This is the situation in which the Rasa SDK action server uses our custom NLP services.

Nowadays, the most amount of NLP pipeline approaches **are end-to-end and avoid the feature extraction step [28]**. However, we are following a more classical approach. We do not have enough data for an end-to-end fine-tuning of a transformer model for the specific task of Industry 4.0 intent classification. Moreover, relying on multilingual deep word embeddings to define the input features for the DIET classifier has some advantages, considering that this model outperforms fine-tuned BERT and is about six times faster to train [26, 27].

2.2. QA module

For the QA task, we followed a transformer-based approach using encoder models (BERT-like) for information extraction using an input context [29]. We propose an extractive QA approach, using question answering models already fine-tuned in Spanish corpora. Basically, these

models add to a pre-trained encoder backbone two different classification heads: (1) one of the heads is in charge of predicting the starting point (index) of the answer and (2) the other is in charge of predicting the ending point. These models receive both the question and the context as the input. They can generalise well to many different input contexts. Then, the answer is provided if the prediction is above a pre-specified confidence threshold.

After exploring all the available models for extractive QA in Spanish in the Huggingface repository, we ended up with the Spanish BERT (BETO)[5] fine-tuned in the SQuAD-es-v2.0 Spanish dataset[6]. This fine-tuned model is available in the Huggingface repository[7]. We used the pipeline implementation for question answering from Huggingface, so that we can handle long input contexts (i.e., process long context documents for the Industry 4.0 domain). These industry-related documents, stored in our servers, define the knowledge base for the topic. We used the inference parameters depicted in Table 1. Note that we set the pipeline's maximum input length (question + context) to the maximum input length allowed by the BERT-based model by design (up to 512 tokens). Since the context documents are longer than that, we followed a window-based approach of the input, applying overlaps.

Table 1
Inference parameters for the question answering implementation using Huggingface QA pipeline.

Inference parameters	Description	Value
Input length	The maximum number of input tokens allowed (question + context)	512
Stride	Input tokens overlap for every document chunk (only if input context exceeds input length)	170
Maximum answer length	The maximum number of tokens allowed in the answer	60

2.3. NLG module

For the NLG task, we followed a transformer-decoder approach, using our own version of a GPT-2 model [30], from the pre-trained 345M parameters DialoGPT-medium model for dialogue [7].

We further fine-tuned this model in an auto-regressive and self-supervised fashion, following the CLM (Causal Language Modelling) objective. We optimised the next token prediction task using a chitchat dataset including some single-turn professional-styled flows[8]. We present the training hyper-parameters in Table 2. This model is publicly available in our Huggingface repository[9].

Table 2
Fine-tuning hyper-parameters for our chitchat 345M parameters DialoGPT-spanish-medium model.

Hyper-parameter	Value
Validation data partition (%)	20%
Training batch size	8
Learning rate	5e-4
Max training epochs	20
Warmup training steps (%)	6%
Weight decay	0.01
Optimiser ($\beta_1, \beta_2, \epsilon$)	AdamW ($0.9, 0.999, 1e-08$)
Monitoring metric (Δ, patience)	validation loss (0.1, 3)

3. Limitations

Some problems could arise when using our chatbot implementation. These issues are related to the following topics:

- **Natural language generation**: Considering that we are using a custom NLG model for chitchat, when the user asks for out-of-scope information, we cannot fully control the outputs. To mitigate this issue, we decided to fine-tune the model with some overfitting so that the probability distribution for the next word is guided in the style of the chitchat behaviour. Moreover, we adjusted the inference generation parameters (e.g., temperature) so that we favour the highest probability words to be predicted.
- **Rasa knowledge base for dialogue and intent recognition**: When new information needs in relation to Industry 4.0 arise, we have to manually add new data for new intent recognition by the Rasa NLU module. We also need new tailored answers. Despite this disadvantage, we gain control over the output, avoiding generation hallucination [31], one important problem in the LLM field nowadays.
- **Document store for information extraction**: As stated in the previous point, the same problem happens in the extractive QA approach. We have to manually maintain the document store in our servers to provide additional information when required.

[5] https://github.com/dccuchile/beto
[6] https://github.com/ccasimiro88/TranslateAlignRetrieve
[7] https://huggingface.co/mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es
[8] https://github.com/microsoft/botframework-cli/blob/main/packages/qnamaker/docs/chit-chat-dataset.md
[9] https://huggingface.co/ITG/DialoGPT-medium-spanish-chitchat

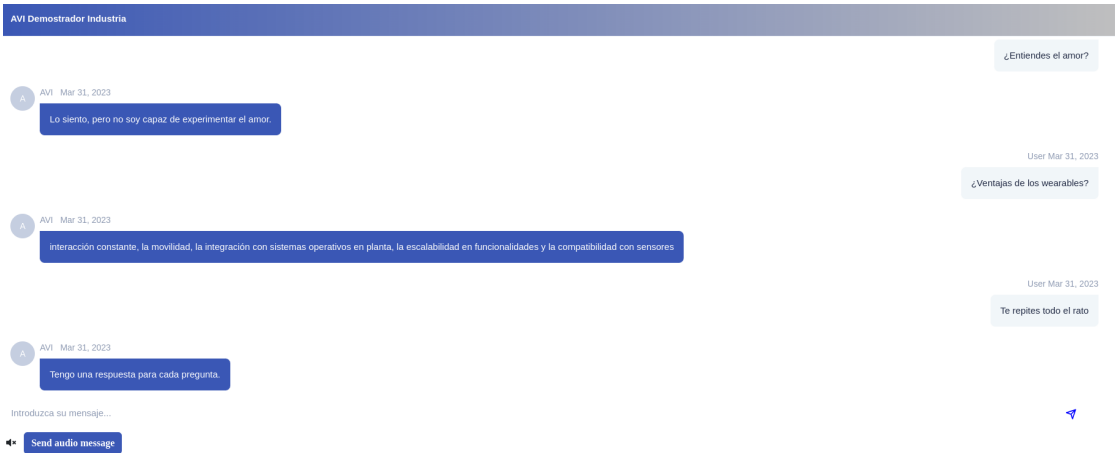

Figure 2: We can see how **the QA and NLG chitchat services take turns depending on the user prompt**.

4. Conclusion and future work

In this work, we present **QUA4I**, a chatbot-oriented application for the Industry 4.0 domain. We included a dialogue system which can also generate automatic responses in natural language. We developed a tailored IVA tool with controlled text generation capabilities, avoiding the LLM hallucination problem. We mixed both NLU and NLG techniques using the Rasa chatbot framework, designing a custom demo for question answering and information extraction. We also included both ASR and TTS modules to improve the chatbot interfaces with the user.

We plan to integrate this chatbot service in a AR (Augmented reality) or MR (mixed reality) environment for assistance in some industry-related domains. Our medium-term objective is to fulfil not only information needs but also take action in response to user commands. We are considering NLG improvements by fine-tuning other state-of-the-art dialogue models, including the Galician version of them. Moreover, we want to explore NLU improvements following an end-to-end approach with a custom intent classifier based on transformer-encoders, avoiding the current feature extraction step in the pipeline. To do so, we are exploring data augmentation techniques based on cross-language translation and NLG models.

Acknowledgments

This project belongs to the CEL.IA network initiative[10], which is supported by the Ministerio de Ciencia e Innovación through the CDTI (Centro para el Desarrollo Tecnológico Industrial) (grant CER-20211022).

[10] https://itg.es/cervera-celia/

References

[1] G. Caldarini, S. Jaf, K. McGarry, A Literature Survey of Recent Advances in Chatbots, Information 13 (2022). URL: https://www.mdpi.com/2078-2489/13/1/41. doi:10.3390/info13010041.

[2] G. Caldarini, S. Jaf, Recent Advances in Chatbot Algorithms, Techniques, and Technologies: DESIGNING CHATBOTS, in: Trends, Applications, and Challenges of Chatbot Technology, IGI Global, 2023, pp. 245–273. URL: http://sure.sunderland.ac.uk/id/eprint/15712/. doi:10.4018/978-1-6684-6234-8.ch011.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, et al., Language Models are Few-Shot Learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[4] B. Workshop, BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, 2022. URL: https://arxiv.org/abs/2211.05100. doi:10.48550/ARXIV.2211.05100.

[5] A. Chowdhery, S. Narang, J. Devlin, et al., PaLM: Scaling Language Modeling with Pathways, 2022. URL: https://arxiv.org/abs/2204.02311. doi:10.48550/ARXIV.2204.02311.

[6] H. Touvron, T. Lavril, G. Izacard, et al., LLaMA: Open and Efficient Foundation Language Models,

2023. URL: https://arxiv.org/abs/2302.13971. doi:10.48550/ARXIV.2302.13971.

[7] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, et al., DIALOGPT : Large-scale generative pre-training for conversational response generation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 270–278. URL: https://aclanthology.org/2020.acl-demos.30. doi:10.18653/v1/2020.acl-demos.30.

[8] L. Ouyang, J. Wu, X. Jiang, D. Almeida, et al., Training language models to follow instructions with human feedback, 2022. URL: https://arxiv.org/abs/2203.02155. doi:10.48550/ARXIV.2203.02155.

[9] OpenAI, Introducing ChatGPT, 2022. URL: https://openai.com/blog/chatgpt/.

[10] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, et al., LaMDA: Language Models for Dialog Applications, CoRR abs/2201.08239 (2022). URL: https://arxiv.org/abs/2201.08239. arXiv:2201.08239.

[11] S. Pichai, An important next step on our ai journey. Introducing Bard, 2023. URL: https://blog.google/technology/ai/bard-google-ai-search-updates/.

[12] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, et al., Large Language Models Are Human-Level Prompt Engineers (2022). arXiv:2211.01910.

[13] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, et al., Compute Trends Across Three Eras of Machine Learning, 2022. URL: https://arxiv.org/abs/2202.05924. doi:10.48550/ARXIV.2202.05924.

[14] X. Amatriain, Transformer models: an introduction and catalog, 2023. URL: https://arxiv.org/abs/2302.07730. doi:10.48550/ARXIV.2302.07730.

[15] S. Qiao, Y. Ou, N. Zhang, X. Chen, et al., Reasoning with Language Model Prompting: A Survey, 2022. URL: https://arxiv.org/abs/2212.09597. doi:10.48550/ARXIV.2212.09597.

[16] J. Huang, K. C.-C. Chang, Towards Reasoning in Large Language Models: A Survey, 2022. URL: https://arxiv.org/abs/2212.10403. doi:10.48550/ARXIV.2212.10403.

[17] Q. Dong, L. Li, D. Dai, C. Zheng, et al., A Survey on In-context Learning, 2023. URL: https://arxiv.org/abs/2301.00234. doi:10.48550/ARXIV.2301.00234.

[18] G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, et al., Augmented Language Models: a Survey, 2023. URL: https://arxiv.org/abs/2302.07842. doi:10.48550/ARXIV.2302.07842.

[19] J. Novet, Microsoft will let companies create their own custom versions of ChatGPT, 2023. URL: https://www.cnbc.com/2023/02/07/microsoft-will-offer-chatgpt-tech-for-companies-to-customize-source.html.

[20] Discussion, ChatGPT fine-tuning as a service, 2023. URL: https://community.openai.com/t/chatgpt-fine-tuning-as-a-service/33803.

[21] Rasa, Rasa Architecture Overview, 2023. URL: https://rasa.com/docs/rasa/arch-overview/.

[22] Rasa, Rasa Components, 2023. URL: https://rasa.com/docs/rasa/components/.

[23] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019. URL: https://arxiv.org/abs/1910.01108. doi:10.48550/ARXIV.1910.01108.

[24] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[25] HuggingFace, How to use DistilBERT, 2023. URL: https://github.com/huggingface/transformers/tree/main/examples/research_projects/distillation#how-to-use-distilbert.

[26] T. Bunk, D. Varshneya, V. Vlasov, A. Nichol, DIET: Lightweight Language Understanding for Dialogue Systems, 2020. URL: https://arxiv.org/abs/2004.09936. doi:10.48550/ARXIV.2004.09936.

[27] M. Mantha, Introducing DIET: state-of-the-art architecture that outperforms fine-tuning BERT and is 6X faster to train, 2020. URL: https://rasa.com/blog/introducing-dual-intent-and-entity-transformer-diet-state-of-the-art-performance-on-a-lightweight-architecture/.

[28] A. Rahali, M. A. Akhloufi, End-to-End Transformer-Based Models in Textual-Based NLP, AI 4 (2023) 54–110. URL: https://www.mdpi.com/2673-2688/4/1/4. doi:10.3390/ai4010004.

[29] K. Pearce, T. Zhan, A. Komanduri, J. Zhan, A Comparative Study of Transformer-Based Language Models on Extractive Question Answering, CoRR abs/2110.03142 (2021). URL: https://arxiv.org/abs/2110.03142. arXiv:2110.03142.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, 2019.

[31] Z. Ji, N. Lee, R. Frieske, T. Yu, et al., Survey of Hallucination in Natural Language Generation, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3571730. doi:10.1145/3571730.