Studies in Conversational AI: Multilingual Capabilities, World Knowledge, and Evaluation Strategies

Maxime De Bruyn

Supervisor **prof. dr. W. Daelemans**

Thesis submitted in fulfilment of the requirements for the degree of doctor in linguistics
Faculty of Arts | Antwerpen, 2024

University of Antwerp

Faculty of Arts

Studies in Conversational AI: Multilingual Capabilities, World Knowledge, and Evaluation Strategies

Thesis submitted in fulfilment of the requirements for the degree of
doctor in linguistics
at the University of Antwerp

Maxime De Bruyn

Antwerpen, 2024

Supervisor
prof. dr. W. Daelemans

Contact
Maxime De Bruyn
University of Antwerp
Faculty of Arts
CLiPS
Stadscampus L
Lange Winkelstraat 40-42
2000 Antwerp, Belgium

Abstract

This thesis studies the evolving landscape of conversational AI. The main research objective is to improve the conversational abilities of conversational agents, with a focus on integrating real-time knowledge and expanding multilingual capabilities.

Integration of External Knowledge The thesis investigates how to incorporate external knowledge into conversational agents without the need for retraining the entire model. This aspect is crucial as it deals with the dynamic nature of information and the need for AI agents to stay updated.

Assessment of Inherent World Knowledge Another critical problem is how to evaluate the inherent world knowledge that users expect from conversational agents. This involves benchmarking the agents' common sense and broad understanding of the world, which is essential for natural and relevant interactions.

Refinement of Agent Responses The research also explores refining the agents' ability to select the most appropriate response from a set of potential replies.

Multilingual Capabilities The thesis recognizes the growing need for conversational agents to be proficient in languages other than English. It examines how additional datasets can be used to develop agents capable of operating effectively across a multitude of languages.

Evaluation Metrics Finally, the thesis addresses the challenge of evaluating conversational agents. Unlike many machine learning applications where a gold standard or reference exists, conversational agents require metrics that acknowledge the multifaceted and subjective nature of conversations, where multiple valid continuations exist.

Overall, the thesis presents a comprehensive approach to enhancing knowledge-grounded conversations, emphasizing better access to external and world knowledge, enhancing non-English language capabilities, and developing more effective evaluation strategies. It synthesizes findings from various interdisciplinary studies and sets a path for future research in the field of conversational AI.

Samenvatting

Deze thesis onderzoekt het steeds evoluerende landschap van conversational AI. Het hoofddoel van het onderzoek is het verbeteren van de gespreksvaardigheden van conversational agents, met een sterke focus op het integreren van real-time kennis en het uitbreiden van meertalige vaardigheden. Dit omvat het aanpakken van verschillende relevante onderwerpen.

Integratie van Externe Kennis De thesis onderzoekt hoe continu veranderende externe kennis geïntegreerd kan worden in conversational agents zonder de noodzaak om het hele model opnieuw te trainen. Dit aspect is cruciaal omdat het te maken heeft met de dynamische aard van informatie en de noodzaak voor AI agents om up-to-date te blijven.

Beoordeling van Inherente Wereldkennis Een andere kritische vraag is hoe de inherente wereldkennis, die gebruikers van conversational agents verwachten, geëvalueerd kan worden. Dit omvat het benchmarken van het logisch redeneren van de agents en hun brede begrip van de wereld, wat essentieel is voor natuurlijke en relevante interacties.

Verfijning van Agent Reacties Het onderzoek verkent ook het verfijnen van de vaardigheid van agents om de meest geschikte reactie te selecteren uit een reeks mogelijke antwoorden.

Meertalige Vaardigheden De thesis erkent de groeiende behoefte aan conversational agents die bekwaam zijn in andere talen dan het Engels. We onderzoeken hoe extra datasets gebruikt kunnen worden om agents te ontwikkelen die effectief kunnen opereren in een scala van talen.

Evaluatie van Conversaties Ten slotte behandelt de thesis de uitdaging van het evalueren van conversational agents. In tegenstelling tot veel machine learning

toepassingen waar een gouden standaard of referentie bestaat, vereisen conversational agents metrics die de veelzijdige en subjectieve aard van gesprekken erkennen, waar meerdere geldige verlopen van een conversatie.

Over het algemeen presenteert de thesis een uitgebreide aanpak voor het verbeteren van knowledge-grounded gesprekken, met de nadruk op betere toegang tot externe en wereldkennis, het verbeteren van niet-Engelse taalvaardigheden, en het ontwikkelen van effectievere evaluatiestrategieën. Het synthetiseert bevindingen uit verschillende interdisciplinaire domeinen en zet een pad uit voor toekomstig onderzoek in het veld van conversational AI.

Preface

As I write the preface for my doctoral thesis, my mind is filled with a deep sense of appreciation for this journey that has been both challenging and rewarding.

First and foremost, I owe a debt of gratitude to Professor Walter Daelemans. The opportunity he extended to me – studying computational linguistics while being paid – was a once in a lifetime chance.

The past four years have witnessed tremendous growth in the field of NLP. My journey began around the time OpenAI released the weights of the large GPT-2 model – a controversy at the time which sounds ridiculous now. This rapid evolution of technology has been a double-edged sword. While it presented exciting opportunities for exploration, it also posed the significant challenge of keeping up in a field where maintaining a singular research track became increasingly challenging. How does one compete with powerful models capable of performing many tasks simultaneously?

This thesis is a reflection of my shifting attention: a constant exploration and a thirst for learning and adaptation. While this approach may not have led to the most straightforward path in my research, it enriched me with diverse experiences across various facets of NLP. Each exploration was a learning curve, adding depth and breadth to my understanding of computational linguistics.

I would like to extend my gratitude to my colleagues, Ehsan and Jeska, for their valuable assistance and feedback throughout these four years. Their insights and perspectives have significantly contributed to my research and personal growth.

Last but certainly not least, I extend my heartfelt thanks to Camille. Her support, patience, and belief in my capabilities have been the foundation of my journey. This thesis would not have been possible without her.

I would also like to express my deep gratitude to the jury members for their insightful comments, constructive criticism, and valuable time devoted to reviewing my work.

In conclusion, this journey through the ever-evolving landscape of NLP has been an enriching and enlightening experience, one that I will cherish forever. The

knowledge and experiences gained during these four years have laid a strong foundation for my future endeavors in the field of computational linguistics.

Contents

Chapter 1

Introduction

> The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.
>
> *Unknown author. Ironically, this quote is often wrongly attributed to Stephen Hawking.*

The king is dead, long live the king!

Technological evolution often follows a cyclical pattern: one innovation makes way for another, leading to new paradigms and shifting our perspectives. In the pre-internet era, our quest for knowledge was largely limited by physical libraries, personal consultations with experts, and voluminous reference books. Yet, the rise of the internet, particularly the emergence of search engines like Google, fundamentally transformed this landscape. What was once a time-consuming activity became a matter of a few quick keystrokes, placing vast amounts of information at our fingertips.

Today, we stand on the brink of another transformative era: the age of Large Language Models (LLMs) such as ChatGPT. These are not mere directories pointing to information. They actively engage, respond, and offer insights in a way similar to human conversation. The change is significant, much like the difference between passively reading a book and actively talking with a knowledgeable friend.

This thesis aims not only to shed light on the underlying mechanisms of these language models but also to enhance their conversational skills. While the current generation of agents displays vast knowledge, there remains room for improvement, especially concerning real-time knowledge integration and multilingual capabilities.

1.1 Research Questions

Building upon the aforementioned challenges, this thesis focuses on improving several aspects of conversational agents:

- Although the knowledge embedded in a conversational agent is limited by the parameters of its model, new information continually emerges. Our first research question is: How can we integrate external knowledge into a conversational agent without re-training the model?
- When interacting with a conversational agent, users often expect it to possess common sense and a broad understanding of the world. How can we assess or benchmark this inherent world knowledge in large-scale language models?
- Conversational agents, particularly those based on the Transformer architecture, are inherently probabilistic models. As such, they can occasionally produce unexpected or inappropriate outputs. How can we refine these agents to select the most appropriate response from a set of potential replies, thereby ensuring consistent and desirable behaviour?
- Most evaluations of Transformer-based conversational agents focus primarily on tasks in English. However, these agents are increasingly utilized across a spectrum of languages. How can we leverage additional datasets to develop agents proficient in multiple languages?
- Evaluating conversational agents presents unique challenges. While many machine learning applications are evaluated against a gold standard or reference, conversational agents diverge in that multiple valid continuations exist for any given conversation. How can we formulate a metric that captures the multifaceted nature of conversations?

1.2 A Brief History of Language Models

Although to the general public, it might appear that ChatGPT emerged as an overnight success, its foundation rests on years of research in the broader field of Natural Language Processing (NLP). The following subsections provide a succinct overview of the historical developments leading to the emergence of Large Language Models and the technical innovations that made them possible to power today's conversational agents.

1.2.1 Early Days of NLP

The genesis of NLP can be traced back to the 1950s with the advent of digital computers. During this time there was a keen interest in translating scientific documents from other nations. The initial machine translation systems, such as the Georgetown-IBM experiment in 1954, though groundbreaking, were extremely limited. They often relied on basic word-level look-ups and rule-based mechanisms to handle inflexions and word order (Manning, 2022). Notably, the authors of the Georgetown-IBM experiment expressed their confidence that machine translation would be a solved problem within the next 5 years.[1] This optimism is reminiscent of today's claims about Artificial General Intelligence being just around the corner.

1.2.2 Text-based Dialogue Systems

Transitioning from these pioneering efforts, the 1960s and 70s witnessed the rise of specialized expert systems designed for question-answering. Notable among these were BASEBALL (Green et al., 1961) and LUNAR (Woods, 1978). These systems incorporated a core database meticulously crafted by domain experts. These systems were proficient, even by today's standards. However, they were purposely limited to a well-defined domain, contrasting with many modern open-domain QA systems that can answer questions about a vast range of topics using a single general model.

1.2.3 Statistical Revolution

In the 1980s and 1990s, the focus transitioned from rule-based systems to statistical models that harnessed the growing abundance of digital text. By learning directly from data without the need for manual rule creation (Daelemans et al., 1996; Elman, 1990), these statistical approaches became vital for tasks like speech recognition and part-of-speech tagging (Manning, 2022). This evolution set the stage for the empirical machine-learning models that now remain central to NLP.

1.2.4 Transformer Architecture and Self-Attention

Building upon these data-driven techniques, 2017 marked a significant milestone with the introduction of the Transformer (Vaswani et al., 2017b) architecture. By improving upon the recurrent layers of models like LSTMs (Hochreiter and

[1] Georgetown IBM Experiment

Schmidhuber, 1997; Mikolov et al., 2011; Cho et al., 2014; Sutskever et al., 2014a) and emphasizing self-attention mechanisms, the Transformer brought a seismic shift in NLP, starting once again with the purpose of improving translation. This architectural change, combined with an influx of textual data, gave rise to influential models such as BERT (Devlin et al., 2019), GPT (Radford and Narasimhan, 2018), and later ChatGPT (OpenAI, 2023). These models exhibited proficiency across multiple NLP tasks without specialized tuning, signalling a new era of transfer learning in NLP (Islam et al., 2023).

Architecture The transformer model, at its core, is elegantly simple and can be represented by the mathematical function $y = f(x)$. Given a text-based prompt or conversation x as input, it assigns a score to each token of its vocabulary, predicting the likelihood of the subsequent token. Text generation then becomes a matter of iteratively applying this function. The preliminary step of transforming raw text into a numerical representation suitable for $f(x)$ is called tokenization. Byte-Pair Encoding (BPE), originally devised for representing absent words in machine translation (Sennrich et al., 2016), is the predominant method for this purpose.

Training As we have seen, a Transformer-based language model assigns a score to each vocabulary item to predict the next token. Training these models is performed in the same way: given a large string of text, the model must predict the next token in the sequence without "seeing" the subsequent tokens, a method also called teacher forcing (Lamb et al., 2016). For example given the string *The quick brown fox jumps over the lazy dog*. The model would first have to predict *quick* from only seeing *The*, then predict *brown* from *The quick*, and so on. Thanks to a clever masking algorithm the model can predict each token in parallel. Despite the extremely simple training objective, these models can perform an outstanding number of tasks, sometimes on the fly or given a few examples (Islam et al., 2023).

Data While this method of training is conceptually simple, it requires an enormous amount of data to perform well. Models with the calibre of Llama 2 (Touvron et al., 2023) and GPT-3 (Brown et al., 2020a) excel when trained on colossal datasets crawled from the web. For example, the Llama 2 models were trained on 2,000,000,000,000 tokens (2T).[2] This is an enormous amount of tokens: much more than any human could ever read. A human reading 300 words per minute (Brysbaert, 2019) for a continuous 80 years would be exposed to only 0.4% of the training data of LLama 2. However, this wealth of data is not without its pitfalls. Web-based data might carry inaccuracies, causing models to assimilate

[2]Llama 1 has done only a single pass over its training data, except for Books and Wikipedia.

and perpetuate biases or mistakes (Lucy and Bamman, 2021). Therefore, while abundant data is advantageous, its quality is pivotal for ethical and faithful NLP implementations.

Hardware & Scaling Up In retrospect, very little has happened to Transformers-based language models since their release — except for model size (x300) and computational power which grew exponentially large in a few years. Modern advancements in hardware have significantly contributed to the success and feasibility of training large-scale Transformer models. Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) have been at the forefront, offering immense computational power to handle the intense requirements of these models. For example, while the initial Transformer model (213 million parameters) required around 576 GPU hours[3], the recent Llama 2 (70 billion parameters) (Touvron et al., 2023) required 1,720,320 GPU hours — around 3,000 times more training time[4].

Black Box Evaluations We understand the mathematics involved in training Transformer models — each neuron in a neural network performs simple arithmetic operations — but we do not have an explanation for why those mathematical operations result in the behaviours we see. This makes it hard to diagnose problems such as hallucinations, hard to know how to fix them, and hard to certify that a model truly *knows* what it is generating (Bricken et al., 2023). Empirical evaluations have consistently highlighted the Transformer's capabilities in achieving state-of-the-art performance across diverse NLP benchmarks. However, most of these analyses treat the model as a black box and focus on the output of the model, providing no insights into the inner workings of Transformers-based language models (Chang and Bergen, 2023). Researchers have yet to develop a comprehensive theoretical foundation explaining why and how the Transformer works internally (Geva et al., 2022; Voita et al., 2023; Bricken et al., 2023).

1.3 LLMs as Knowledge Reservoirs

Large Language Models (LLMs) like GPT-4 can be seen as vast reservoirs of human knowledge, a reflection of the digital corpus from which they were trained. The voluminous nature of the data these models have processed offers them a sweeping insight into diverse domains, ranging from scientific facts to cultural nuances (Chang and Bergen, 2023). However, quantifying the breadth of this

[3] 3 days with a single node consisting of eight P100 GPUs
[4] 35 days and a cluster of 2,048 A100 GPU

reservoir is not straightforward due to the non-symbolic storage and distribution of information in these models. Moreover, while LLMs excel in certain forms of knowledge representation, they exhibit unique vulnerabilities like knowledge hallucinations. This section delves into the depth, breadth, and limitations of the knowledge encapsulated by LLMs, shedding light on their strengths and areas of caution.

1.3.1 Common Sense & World Knowledge

Unlike humans, language models don't learn from physical experiences or interactions. Our common sense often arises from the real-world experiences we have interacting with our environment while growing up. A language model doesn't have a body, sensory experiences, or the ability to interact with the physical world in the way humans do. While humans continuously update their knowledge and refine their understanding based on new experiences and feedback, static language models like GPT-4 cannot learn over time or adjust their knowledge post-training.

Consider a simple yet intriguing question: "Does water make things wet?" GPT-4 would probably respond affirmatively. This might seem trivial, but it underscores an important point. GPT-4 does not possess experiential knowledge; it doesn't "feel" wetness. Instead, it has been exposed to countless textual references about water and its properties during its training phase. Through unsupervised training on vast datasets, language models thus learn a form of common sense reasoning by correlating recurring patterns in data.

Unfortunately, directly measuring the depth of knowledge and common sense within LLMs is elusive as Transformer-based models encode their knowledge non-symbolically, distributing it across their parameters. Hence, gauging their relative knowledge often involves comparing one model to another on a collection of static benchmarks.

Chapter 7 probes the world knowledge and common sense acumen of LLMs, using the game of Twenty Questions. Our analysis highlights GPT-3's challenges with size comparisons.

1.3.2 Knowledge Hallucination

Generative models under the Transformers umbrella, like GPT-4, are prone to "knowledge hallucinations": instances where the model generates seemingly logical, yet unfounded or misleading information (Shuster et al., 2021a; Zhang et al., 2023; Peng et al., 2023; Agrawal et al., 2023). Such anomalies stem from their

training objective, which prioritizes word sequence prediction over factual accuracy. Consequently, they may occasionally generate coherent or probable, but factually erroneous outputs.

Knowledge hallucinations are problematic when using these models in a corporate environment where one typically wants to control what the model can say. For instance, what if the bot offers an unwarranted discount? For these reasons, retrieval models have gained popularity (Henderson et al., 2020). Instead of generating an answer from scratch, the model selects the best possible answer from a set of pre-defined candidates. While this method provides control, it's limited in coping with unforeseen situations. In Chapter 4, we introduce a Transformer-encoder model designed to select the best possible answer in Dutch. During its development, we identified a lack of data for pre-training our model to make optimal selections. Though our approach was effective, scaling it to other languages proved challenging. To address this, Chapter 5 details our efforts to leverage FAQs from the web, allowing us to pre-train a model capable of selecting suitable answers for a question across 21 different languages.

1.3.3 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) combines the strengths of large-scale generative models with external retrieval mechanisms (Izacard et al., 2022; Lewis et al., 2020c; Izacard and Grave, 2020; Borgeaud et al., 2022; Khandelwal et al., 2019). RAG aims to broaden the scope of purely generative models by equipping them with tools to extract information from external repositories before, or during the generation phase.

The conventional RAG framework integrates two core components:

1. **Retriever** Before crafting a response, the model searches through a vast corpus to pinpoint pertinent documents or snippets, often assisted by adept retrieval systems such as Dense Retriever.
2. **Generator** With the relevant passages at hand, the generative model formulates a comprehensive response, merging the external information with its intrinsic knowledge.

RAG's dual-step synergy facilitates responses beyond the model's pre-training data limits. For example, if a model's training data stops at a certain year, the retrieval component can source information on subsequent events, which the generator subsequently weaves into its answer.

Furthermore, RAG can potentially mitigate generative models' constraints, including knowledge hallucinations (Shuster et al., 2021b). By corroborating and cross-referencing information from external data sources, RAG models stand a better chance of producing unfounded or inaccurate data.

Chapter 2 unveils a RAG-integrated BART model leveraging Wikipedia passages to foster knowledge-anchored conversations.

1.4 Evaluation Metrics

This section delves into the metrics utilized to evaluate the efficacy and reliability of conversational agents. We will discuss various aspects, beginning with the importance of 'small talk' in human-machine interaction.

1.4.1 Small Talk

Small talk is of paramount importance to humans: it promotes social cohesiveness, reduces inherent threat values of social contact, and helps to structure social interaction (Coupland, 2003). However, conversationalists tend to view the ability to do small talk as something not relevant to be studied (Coupland, 2003).

Although literature may have taken small talk for granted, this is not the case in popular culture. Popular books such as *How to Make Friends & Influence People* or *How to Talk to Anyone: 92 Little Tricks for Big Success in Relationships* provide insights into improving one's small talk abilities, especially in a business setting. These insights could be summarized this way:

- Effective interpersonal communication requires genuine interest and active listening.
- Encouraging individuals to share their experiences and aligning the discourse with their interests deepens the connection.
- Practices such as prolonged eye contact and echoing a speaker's words enhance comprehension and engagement.
- Genuine compliments, skilful introductions, and strategic conversation starters are crucial for cultivating meaningful relationships.
- At the heart of these strategies is the sincere valuation of the conversational partner, highlighting the importance of authenticity in dialogues.

However, although these strategies may help humans or serve as guidelines for a large language model's prompting, their qualitative nature makes them challenging to directly translate into quantifiable metrics for evaluating the quality of small talk.

One-To-Many The fundamental problem of evaluating the quality of a conversation lies in the multi-faceted aspect of conversation. A successful conversation can be continued in any direction. However, in machine learning, we tend to evaluate the quality of output with regard to how closely it mimics the "ground truth". In the case of a conversation, the "ground truth" is the next gold utterance. This evaluation approach is far from ideal, as it doesn't account for the diverse and varied ways in which conversations can naturally unfold.

In human interactions, the richness of conversation comes from the unpredictability and the multitude of potential responses. A question like "How was your day?" can elicit a myriad of valid responses, ranging from detailed recounts of one's day to abstract reflections, to simple affirmations or negations. To confine the evaluation of a conversational agent's response to a single "correct" answer neglects the essence of human conversation.

In Chapter 3, we used another approach and estimated the probability that a large language model would continue a given conversation with a negative utterance as a signal for the quality of the conversation or turn.

1.4.2 Memorization & Generalization

Generalization One of the paramount concerns for conversational agents, and language models in general, is their ability to generalize. Generalization refers to the capacity of a model to perform effectively on data it has not seen before. For instance, consider a conversational agent designed to assist users in booking flights. While the agent might have been trained on numerous queries like *"Book a flight to Brussels"* or *"Show me flights on Friday"*, real-world users might present it with unconventional queries like *"Find me a plane to Zaventem"* or *"Which flights are there on the day after tomorrow?"*. A model that generalizes well would be able to understand and process these novel inputs without having been explicitly trained on them. This ability is pivotal because real-world applications are filled with unpredictable and unique inputs, and only a model that generalizes effectively can navigate such unpredictability.

Unlike many classical machine learning models, Transformer-based language models are usually pre-trained on an enormous amount of text data sourced from the Internet, which can drastically improve the generalization capabilities

of the model.

Memorization & Contamination A corollary to the aforementioned discussion on generalization arises: is the model genuinely generalizing or memorizing the trillions of tokens from its pre-training data? A recent analysis by Grosse et al. (2023) used influence functions to analyze the contribution of pre-training data on the text generated by large language models. The main conclusion is that the larger the model the more sophisticated the generalization patterns: smaller models stick to lexical similarity while larger models can generalize to more abstract concepts. Furthermore, the authors studied the output of an AI assistant and were unable to find instances where the model simply regurgitated instances from the pre-training data (except in the case of famous quotes or passages targeting explicit memorization).

As it is impossible to analyze the knowledge and capabilities of language models directly, we usually resort to analyzing their performance on a variety of static benchmarks: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021a), Human-Eval (Chen et al., 2021a), etc. While it is the most straightforward approach to evaluate and compare these models, this approach is prone to intentional or unintentional *contamination* depending on which source is included in the pre-training data. For example, Touvron et al. (2023) found LLama-2 70B to be substantially better on test set examples with a large overlap with the pre-training set. More often than not, the pre-training data used to train these large language models is not publicly available which makes it impossible to test for a possible contamination issue.

The problem of generalization and contamination does not stop at memorizing the pre-training data, language models are often fine-tuned for specific tasks. We also want to know how well a model trained on a given dataset will transfer to instances which do not resemble the training set. In this case, the distinction between generalization and memorization is vital because true generalization implies the ability to extrapolate knowledge and concepts to new, unseen instances, whereas memorization suggests the model is merely recalling specifics from its extensive training (Chowdhery et al., 2022a). To make matters worse, several datasets contain duplicates (or close duplicates) between the training and test set (Lewis et al., 2021a).

Metrics Current evaluation metrics often struggle to differentiate between a model's genuine generalization and memorization. This highlights a need for improved evaluation techniques that consider the similarity between test and training data. In chapter 8, we attempted to gauge world knowledge by accounting for overlap between training and validation sets, revealing that language

models indeed perform better on frequently seen topics in pre-training data.

Data Augmentation Using data augmentation techniques exacerbates this problem. In Chapter 6, we employed large language models to augment the size of a multilingual intent and slot-filling dataset. As the size of the generated data was extensive, part of it was also included in the test set. Can we conclude that the model trained on the augmented dataset is better? To answer this question, we used logistic regression to measure the performance of both models on the test set while taking into account the similarity to the training set.

1.5 Multilingual

Conversational agents serve a diverse linguistic audience. Historically, Transformer-based language models like BERT and GPT were built for English. For multiple languages, separate models were required: Bertje for Dutch (de Vries et al., 2019), CamemBERT for French (Martin et al., 2020). If none were available, a multilingual could also be used such as mBERT or XLM-RoBERTa (Conneau et al., 2020; He et al., 2021). However, the introduction of models like GPT-3 signalled a shift. Although designed predominantly for English, GPT-3's training data comprised around 7% of non-English text. Despite its unsupervised training, GPT-3 exhibited impressive multilingual capabilities, and the trend seems to have continued with successors like GPT-3.5 and GPT-4.

However, this multilingual approach has its costs. The tokenizers of models like GPT-3.5 and GPT-4, tailored predominantly for English, require more tokens for non-English texts, leading to increased costs, latency issues, and limitations in handling long-term dependencies (Petrov et al., 2023). For example the English query *Book a flight to Brussels* requires only 5 tokens as each word belongs to a token in the vocabulary, Dutch requires 12 tokens which is about 2 times more expensive, 11 for French, 33 for Russian, and 56 in Thaï — 11 times more expensive than English. This simple example shows the wide disparity in cost associated with the use of large language models for non-English languages.

Benefits of Multilingual Models Another possibility is to use multi-lingual tokenizers, such as the tokenizer from XLM-RoBERTa. Using this tokenizer the query *Book a flight to Brussels* only requires 6 tokens for English, Dutch, and French. Russian and Thai require 7 and 8 tokens respectively. From our experience, a standard approach for chatbots has been maintaining separate models for each language. Still, this isn't cost-effective in the long run and hampers cross-language learning. An all-encompassing model for all languages would simplify

operations, allowing for efficient cross-language learning and system maintenance. In Chapter 5 and 6 we used multilingual models to answer frequently asked questions and perform intent & slot filling using multilingual models. Our results demonstrated that training in multiple languages at once enhances the results compared to the single-language approach.

1.5.0.1 Tokenization Issues

Non-spaced Languages The way to train an intent detection & slots filling model is to annotate existing queries for slots and intents. For example, the query *Book a flight to [destination: Brussels]* is annotated with Brussels as the destination. This approach works universally, even for non-spaced languages such as Chinese or Japanese. However, things start to break down when the tokenizer is involved as the boundaries of slots and tokens do not always overlap. We explore this problem in Chapter 6.

The Underestimated Significance of Tokenizer Quality A surprising revelation in the realm of large models is that despite the vast resources dedicated to the training of GPT-4, it persists with the same tokenizer as GPT-2, bringing along its inherent limitations. A striking example of this is its handling of the string "davidjl". When prompted with *Please write the string: "davidjl".*, GPT-4 generates just a single quotation mark. Yet, other word tests appear to pass without hiccups. This anomaly can be traced back to "davidjl" being treated as a singular token in the GPT-2/3/4 tokenizer[5]. Worryingly, this token frequently surfaced in contexts of escalating numbers, largely devoid of meaningful content. This suggests the tokenizer's exposure to potentially dubious data during training, as evidenced by its data sources like Reddit, where the user davidjl123 took part in numeral sequences[6]. Such revelations underline a pressing concern: the quality and integrity of tokenizers, often overshadowed by other aspects, deserve more attention and scrutiny. Apart from this, the tokenizer also grapples with handling nuances in capitalization and spacing. A case in point is the multiple tokens linked to the word "yes" in the sentence *"Yes yes Yes.yes"*[7].

1.6 Conclusion

In conclusion, the age of Large Language Models started a new era of conversational agents that possess the ability to actively engage, respond, and offer insights

[5]Token id 23282
[6]https://www.reddit.com/user/davidjl123/
[7][5297, 3763, 3363, 13, 8505]

in conversations. This development represents a significant shift from the pre-internet era, where knowledge acquisition was limited to physical libraries and reference books. However, despite the impressive capabilities of current models, there are still important challenges to overcome. This thesis aims to tackle some of these challenges by investigating how to integrate external knowledge into conversational agents, assess their common sense and world knowledge, refine their selection of appropriate responses, improve their multilingual capabilities, and develop metrics that capture the multifaceted nature of conversations.

Chapter 2

BART for Knowledge Grounded Conversations

This chapter was published in the Proceedings of KDD Workshop on Conversational Systems Towards Mainstream Adoption (KDD Converse'20). ACM, New York, NY, USA - ISSN 1613-0073 - 2666(2020), p. 1-6

2.1 Introduction

Large transformer-based language models have shown excellent capabilities in generating human-like conversations (Adiwardana et al., 2020a; Roller et al., 2021). While powerful, these models have a major drawback: they cannot expand their factual knowledge of the world without being trained on new data (Lewis et al., 2020b). As an example, all models trained before the COVID-19 outbreak have no knowledge about the coronavirus epidemic.

It should be possible to allow open-domain conversational models to use additional external knowledge sources for factual knowledge. Their knowledge source should be easily extendable with recent information.

Current knowledge grounded open-domain agents limit the external world knowledge to one sentence (Dinan et al., 2019; Roller et al., 2021), or to a single vector (Fan et al., 2020). We believe limiting models this way is insufficient for open-domain conversational agents, and show that increasing the number of passages retrieved from memory leads to more human-like replies from the agent.

2.2 Related Work

Knowledge grounded dialog systems can be described as sequence-to-sequence models (Sutskever et al., 2014b) conditioned on an external knowledge source. Grounding a conversation in external knowledge requires two different abilities: retrieving the right knowledge amongst multiple candidates and effectively using this knowledge to generate the next utterance.

One way of providing context to the model is to concatenate the chat history with the knowledge source. (Budzianowski and Vulic, 2019) concatenated the context, belief state, and database as input to a task-oriented GPT2 model (Radford et al., 2019). (Wolf et al., 2019b) concatenated the agent's persona with the previous chat history. (Liu et al., 2018) find that this approach struggles with handling longer contexts. (Wang et al., 2019) separate source and context encoding and interleave source and context attention when decoding.

In some cases, the length of the context may be too large to be concatenated with the chat history (e.g. multiple Wikipedia articles). (Dinan et al., 2019) introduce the Transformer Memory Network models, capable of retrieving and attending to knowledge and outputting a response, either in retrieval or generative mode. (Fan et al., 2020) present a KNN-based information fetching module that learns to identify relevant information from external knowledge sources in the context of a dialogue dataset. The Wizard Generative Model (Roller et al., 2020a) uses a Poly-encoder (Humeau et al., 2019) to retrieve a single sentence from an external knowledge source.

Retrieval dialog systems (Weston et al., 2018; Dinan et al., 2019) can also be framed as knowledge grounded agents where the knowledge source is a fixed set of utterances and the task is to select the most appropriate utterance given the context.

In this paper, we expand on the work of (Dinan et al., 2019). We fine-tune a BART model (Lewis et al., 2019) to retrieve multiple sentences (instead of a single one) from an external knowledge source, and use it effectively to generate the next utterance in a conversation.

2.3 Dataset

2.3.1 Wizard of Wikipedia

We use the Wizard of Wikipedia dataset (Dinan et al., 2019) where two participants engage in chit-chat conversations. One of the participants is the wizard,

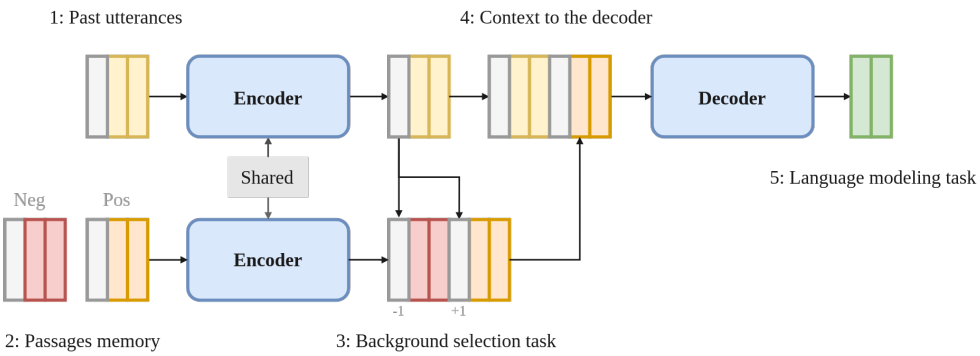

Figure 2.1: BART model adapted for knowledge grounded conversations. 1: The chat history is tokenized, a $query$ token is prepended, the result is encoded by the encoder. 2: Each memory passage is tokenized, and prepended with a key token. The resulting matrix is encoded by the encoder. 3: The $query$ vector is compared against the key vectors from the memory with a dot product attention. The first k passages with the highest score are selected. 4: The full sequence of the chat history and the full sequence from the selected memory passages are concatenated and given as context to the decoder. 5: Generation of the next utterance of the Wizard.

and the other the apprentice. The wizard plays the role of a knowledgeable expert while the other is a curious learner. The goal of the wizard is to inform its conversation partner about a topic that one of them will choose. The wizard has access to an information retrieval system that shows paragraphs from Wikipedia possibly relevant to the conversation. Before each conversation turn, the wizard can read these paragraphs and then potentially base its next reply on that observed knowledge.

The authors collected 22,311 conversations with a total of 201,999 turns on 1365 open-domain dialog topics (e.g. commuting, gouda cheese, bowling).

The dataset is divided in a train, validation and test set. The validation and test sets are sub-divided into seen and unseen. The seen sets share conversation topics with the training set while the unseen sets do not.

2.4 Model

Our goal with this dataset is to train an agent capable of conversing about any domain. We use a model to replace the wizard in the conversations. To generate the next utterance x_{t+1}, the model has access to the previous conversation turns $x_1, ..., x_t$ and to a hierarchical knowledge source: M. Each element of the

knowledge source, m_i, is composed of a topic and a sentence belonging to that topic.

Similar to the End-to-End (E2E) Transformer MemNet of (Dinan et al., 2019), we use an encoder-decoder Transformer (Vaswani et al., 2017b) as our base sequence-to-sequence model. Instead of pre-training our model on a Reddit corpus, we use a pre-trained BART model (Lewis et al., 2019). An illustration of our model is shown in Figure 2.1.

The knowledge source M is filtered before each turn using the same procedures as in (Dinan et al., 2019).

2.4.1 Encoder

While some approaches choose to have a separate encoder for knowledge retrieval and for conversation modeling (Dinan et al., 2019; Fan et al., 2020), we use a shared encoder to encode the conversation context x, and the filtered knowledge source M. Every m_i is encoded independently.

We choose to share the encoder because the purpose of an encoder is to understand text, it does not make sense to have two encoders do the same thing but for different sources (chat history and knowledge memory). This architectural choices also reduces the model size.

To let the model recognize the difference between a knowledge piece and a conversation history, we use segment embeddings (Wolf et al., 2019b). We introduce three segment embeddings, one for the wizard's turn, one for the apprentice's turn, and one for the knowledge passages. We prepend the conversation context x with a special token q, the query token. We prepend each knowledge source candidate with another special token k, the key token. After decoding, the query and key vectors are projected to a lower dimension using a linear layer.

2.4.2 Knowledge Selection

After the encoding step, our key and query tokens become the key and query vectors. We concatenate the key vectors k_i from the knowledge source encoding into the query matrix K.

We then train the model to recognize which single knowledge passage (the gold knowledge) k_i was selected by the wizard. The query vector q from the conversation history is compared against K to produce a dot product attention over the

knowledge source. We train the model to select the gold knowledge passage with a cross-entropy loss over the knowledge attention.

2.4.3 Decoder

We concatenate the full sequence of the n first knowledge candidates m_i with the chat history. This context matrix is then given as memory to the decoder. To let the decoder know that it is generating the next utterance of the wizard, we use the same segment embedding for the wizard as in the encoding step. We train the model to minimize the negative log-likelihood of the response utterance x_{t+1}.

To summarize, our model uses a pre-trained BART model to retrieve relevant knowledge and to generate the next utterance. We improve on the current methods in two ways. First we introduce a key and $query$ token to perform the retrieval step with a shared encoder, second we allow the model to retrieve multiple full (i.e. not a vector representation) passages from the memory.

2.5 Experiments

We conduct several experiments to analyze the ability of our model to select and use knowledge in dialogue.

The model uses the large version of BART (Lewis et al., 2019), which has 12 layers in the encoder and 12 layers in the decoder and an inner dimension of 1024. The model has approximately 400M parameters. We use the BART implementation from HuggingFace (Wolf et al., 2019a).

On top of the original implementation, we add a segment embedding layer and two additional tokens (query and key token) to the vocabulary. We also add two linear layers to project the key and query vector to a dimension of 512.

Before each turn, the wizard is presented with a varying number of memory passages retrieved by an IR system: the visible passages (see Dinan et al. (2019) for a detailed description). We feed a subset of the visible passages to the model (40 sentences per utterance). The visible passages can be divided into positive (gold passage) and negative examples (non-gold passages). We pool together the negative examples of a single batch to increase the difficulty of the task at a reduced computational cost (the model has to choose from a larger pool of already computed key vectors).

During training, we use a forcing teacher strategy and disregard the results from the knowledge retrieval step. Instead, we give as context (memory) to the decoder,

Method	Seen Test		Unseen Test	
	R@1	F1	R@1	F1
Random	2.7	13.5	2.3	13.1
IR baseline	5.8	21.8	7.6	23.5
BoW MemNet	23.0	36.3	8.9	22.9
Transformer	22.5	33.2	12.2	19.8
Transformer (+Reddit pretraining)	24.5	36.4	**23.7**	**35.8**
Transformer (+Reddit pretraining, +SQuAD training)	25.5	36.2	22.9	34.2
Retrieval Transformer MemNet (no auxiliary loss)	12.9	24.6	14.6	26.3
Gen. E2E Transformer MemNet (no auxiliary loss)	13.4	28.3	11.8	25.9
Gen. E2E Transformer MemNet (w/ auxiliary loss)	21.1	32.8	14.3	22.8
BART	**26.0**	**38.9**	19.9	33.9

Table 2.1: Knowledge retrieval performance on the seen and unseen test set. The BART model outperforms all methods on the seen test set (unigram F1 and perplexity) and comes close to the best performing methods on the unseen test set, even though it does not have a separate module specialized in knowledge retrieval (as the Transformer models). The seen test set shares conversation topic with the training set, while the unseen test does not.

the first five passages from the gold topic (the gold passage is always the first one). By feeding it multiple sentences, the model is trained to further select the relevant piece of information in the decoder. We believe this makes the decoder more robust to noise coming from the knowledge retrieval step.

We train the model to simultaneously optimize for the knowledge selection task and the language modeling task for three epochs, with a constant learning rate of 10^{-5}, linearly increased from zero over 1000 steps.

We did not test using a separate encoder (see *Shared* link in Figure 2.1) as this would increase the parameters count by 50%.

2.6 Results

We analyze the performance of the model on two axes: knowledge retrieval and next utterance generation.

2.6.1 Retrieval Task

Similar to (Dinan et al., 2019) we use recall@1 and unigram F1 between the retrieved knowledge and the gold knowledge item as evaluation metrics. The

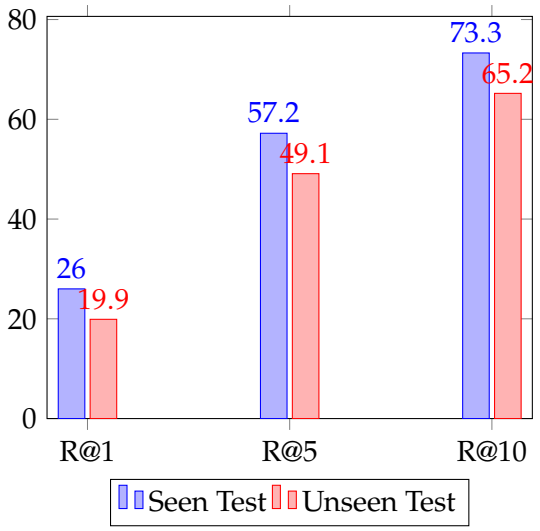

Figure 2.2: Recall metrics on seen and unseen test set. The model retrieves the right memory passage 26% of the time on the seen test set. When retrieving the first 10 passages, the gold passage is included in the retrieved results 73% of the time. These results show the importance of retrieving multiple passages from the memory.

results are displayed in Table 2.1.

Our model uses a shared encoder to encode the conversation context and to retrieve the relevant knowledge. It is therefore best compared against the Generative E2E Transformer MemNet of (Dinan et al., 2019), the other models have a separate knowledge retrieval module. We show that a shared encoder can achieve similar performance on this task as specialized modules.

As our model is capable of handling multiple knowledge pieces in the decoder, we also report recall@5 and recall@10 in Figure 2.2. The first five results contain the gold passage around 50% of the time.

The difference in retrieval performance between the seen and unseen set could indicate that the model overfitted the training set, or that the size of the dataset is too limited to generalize to unseen topics.

2.6.2 Generation Task

The second objective of our model is to use the past conversation and the retrieved knowledge to generate the next utterance.

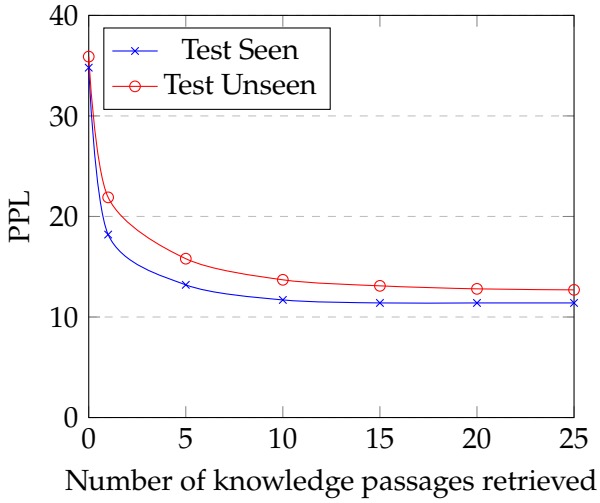

Figure 2.3: Perplexity results per number of passages retrieved. The inclusion of knowledge has a significant impact on perplexity, on the seen and unseen test set. The performance of the model gets better as more knowledge passages are retrieved. There is no trade-off between the number of included passages and the model's performance in terms of perplexity.

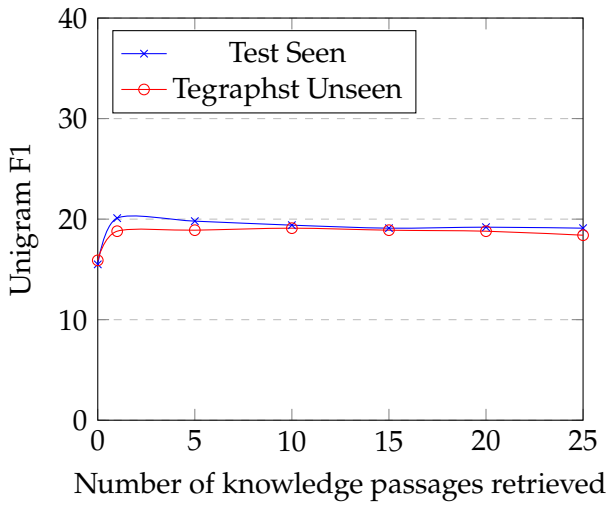

Figure 2.4: Unigram F1 results per number of passages retrieved. The inclusion of knowledge has a significant impact on the model's performance in terms of unigram F1. Contrary to Figure 2.3, the model's performance peaks at one passage retrieved on the seen test set.

Method	Seen Test		Unseen Test	
	PPL	F1	PPL	F1
Repeat last utterance		13.8		13.7
E2E Transformer MemNet (no auxiliary loss)	66.5	15.9	103.6	14.3
E2E Transformer MemNet (w/ auxiliary loss)	63.5	16.9	97.3	14.4
Two-Stage Transformer MemNet	54.8	18.6	88.5	17.4
Two-Stage Transformer MemNet (w/ K.D.)	46.5	18.9	84.8	17.3
KIF-Augmented Transformer*		**25.9**		**22.3**
BART	12.2	20.1	14.9	19.3

Table 2.2: Next utterance generation performance on the seen and unseen test set. The BART model outperforms the shared encoder methodologies (E2E) and non-shared encoder methodologies (Two-Stage) of (Dinan et al., 2019), but falls short of the KIF-Augmented Transformer (Fan et al., 2020) in terms of unigram F1. Perplexity number cannot be directly compared because of differences in vocabulary sizes. *Fan et al. (2020) did not report perplexity numbers.

Although BART was pre-trained on a denoising task, it quickly adapted to dialog generation.

Similar to (Dinan et al., 2019), we use the perplexity of the gold utterance and unigram F1 between the generated utterance and the gold utterance as evaluation metrics, see Table 2.2. The model achieves a better performance than (Dinan et al., 2019) in terms of unigram F1 but falls short of (Fan et al., 2020). The perplexity numbers cannot be directly compared between models because of differences in vocabulary size.

As our model is capable of handling more than one passage of knowledge, we also report the numbers for 0, 1, 5, 10, 15 and 20 knowledge passages retrieved, see Figures 2.3 and 2.4. In terms of perplexity, the more passages are retrieved, the better the performance. The higher the number of passages retrieved, the higher the probability of it containing the gold passage used by the Wizard to generate the next utterance (see Figure 2.2 for recall numbers). Hence, the model is less perplexed by this particular utterance. This phenomenon is true for the seen and unseen test set.

In terms of unigram F1, the performance reaches a maximum at 1 passage retrieved, while the unseen test reaches a maximum at 10. Unigram F1 and perplexity tell two different stories: perplexity says it is beneficial to include at least 10 passages, while unigram F1 says one is enough.

(Adiwardana et al., 2020a) show perplexity is correlated with SSA (Sensibleness and Specificity Average) and state that optimizing for perplexity is a good proxy for optimizing the human likeliness of a model. Using their result as hypothe-

sis, increasing the number of passages retrieved from memory results in more human-like models.

2.7 Future Work

An unsupervised pre-training of the BART model for simultaneous context retrieval and generation could help bridge the gap between seen and unseen performance.

The problem of knowledge selection is not one-to-one, but one-to-many. There are possibly many relevant passages for a single user query. The dataset could be updated to reflect that fact.

Unigram F1 has no semantic understanding of the generated text. Evaluating the model with USR (Mehri and Eskenazi, 2020b), a reference-free metric that trains unsupervised models to measure several desirable qualities of dialog, could help in the comparison of models.

2.8 Conclusion

In this work, we showed how a BART (Lewis et al., 2019) model can be extended to make use of an external memory. This model was successfully implemented in a knowledge grounded conversational setup using the Wizard of Wikipedia dataset (Dinan et al., 2019).

Current models retrieve only one sentence or vector from the memory (Dinan et al., 2019; Roller et al., 2020a). Our analysis showed that it is limiting the potential of current models as retrieving multiple sentences from the memory diminishes the model's perplexity to the gold utterance.

We also showed it is not necessary to have a separate encoder for knowledge retrieval and context encoding. A shared encoder can achieve competitive results in the knowledge retrieval task, limiting the model size and complexity.

Chapter 3

Open-Domain Dialog Evaluation using Follow-Ups Likelihood

This chapter was published in the Proceedings of the 29th International Conference on Computational Linguistics, - ISSN 2951-2093 - Gyeongju, International Committee on Computational Linguistics, 29(2022), p. 496-504

3.1 Introduction

Despite the recent progress in Natural Language Processing, the automatic evaluation of open-domain conversations remains an unsolved problem. It is difficult to establish criteria to measure the quality of a system. Task-oriented dialog systems use metrics such as task success or dialog efficiency. However, these do not apply to open-domain conversational agents (McTear, 2020).

Currently, there are two options for open-domain dialog evaluation: human evaluation and automated evaluation. Thanks to their understanding of natural language, humans are able to digest the entire dialog context in order to meaningfully evaluate a response (Mehri et al., 2022). Human evaluation also has its shortcomings: inconsistency in ratings (the same annotator may give two different scores depending on the mood), lack of reproducibility, and cost (Mehri et al., 2022).

The second option is to use automated evaluation metrics. Methods inherited from sequence-to-sequence machine translation such as BLEU (Papineni et al., 2002) evaluate the generated utterance by comparing it to the ground-truth. By doing so, these methods miss the one-to-many characteristic of conversation: a conversation may evolve in more than one valid direction.

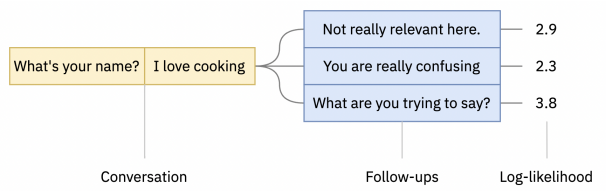

Figure 3.1: Illustration of our method. We measure the probability (log-likelihood) that a language model will continue the conversation with a set of predefined follow-ups. This paper shows that the sum of the individual log-likelihoods correlates strongly with human evaluations.

To tackle this problem, researchers came up with reference-free evaluation metrics: the generated utterance is not compared to a ground truth but evaluated on its own.

FED (Mehri and Eskenazi, 2020a) is an unsupervised reference-free evaluation metric. It uses the idea that one can use the next utterance in a conversation to rate the turn before it. When users speak to a system, their response to a given system may implicitly provide feedback for the system. FED uses a set of predefined follow-ups and the log-likelihood from a language model to measure 18 fine-grained attributes in a conversation.

Inspired by the FED metric, we propose a new evaluation method called FULL (Follow-Up Log-Likelihood). We start by explaining our method and how it departs from the original FED metric. Next, we explain our choice of language model and follow-ups. Finally, we demonstrate that our new method achieves the highest correlation with human evaluations compared to 12 automated metrics. We open-source our evaluation code[1] and publish FULL as a Python package[2] for easy usage.

3.2 Related Work

This section reviews the existing literature on evaluation metrics for open-domain conversations. In the interest of space, we limit ourselves to studying reference-free methods (methods that do not require a ground truth). The interested reader is encouraged to read Yeh et al. (2021) for a full review.

GRADE (Huang et al., 2020) and DynaEval (Zhang et al., 2021) use a graph-based structure to model the dialog-level interaction between a user and a system.

[1]https://github.com/maximedb/full
[2]https://pypi.org/project/full/

DynaEval distinguishes between well-formed dialogs from carefully constructed negative samples. MAUDE (Sinha et al., 2020) is also trained to distinguish a correct response from a randomly sampled negative response using a contrastive loss. FlowScore (Li et al., 2021b) evaluates the quality of a dialog using the dynamic information flow in the dialog history.

USR (Mehri and Eskenazi, 2020c) trains several models to measure different qualities of dialogs. A masked language modeling head measures the fluency of the conversation, a retrieval model determines the relevance of a response, and a fact-to-response model checks whether a response conditions on knowledge. USL-H (Phy et al., 2020) also has three internal models, although they measure different attributes: grammatical correctness, sensibleness, and the likelihood of a given response. Other notable evaluation methods include Ghazarian et al. (2020); See and Manning (2021); Ghazarian et al. (2022b,a)

FED (Mehri and Eskenazi, 2020a) and HolisticEval (Pang et al., 2020) both use GPT-like (Radford et al., 2019) models to evaluate conversation on several attributes. FED computes the likelihood of manually designed follow-up utterances to measure multiple dialog qualities without supervision. HolisticEval uses a GPT-2 model to measure coherence, fluency, diversity, and consistency.

3.3 Method

Our metric FULL (Follow-Up Log-Likelihood) is a reference-free evaluation method for dialogs inspired by FED (Mehri and Eskenazi, 2020a). Figure 3.1 provides an overview.

3.3.1 Follow-Up Utterance for Evaluation

Our method uses follow-up utterances to evaluate the quality of a conversation (Eskénazi et al., 2019). When interacting with a system, users may provide implicit feedback about the conversation in the semantics of their response. For example, if a user ends a conversation with *It was a pleasure talking to you*, we can reasonably assume it was a pleasant conversation. On the other hand, if a user ends a conversation with *What are you talking about?*, we could conclude that the user is confused about the state of the conversation.

3.3.2 Log-Likelihood of Follow-Ups

We do not have access to the next utterance in an interactive setting. Instead, we ask a language model to play the role of a human. We ask the model how likely it is to generate a fixed set of follow-ups. For example, if the language model is likely to continue a conversation with the follow-up *I don't understand what you are saying*, we could conclude that the utterance generated by the system does not make sense.

FULL analyzes the quality of a response r in the context of a dialog history h with a language model M and a set of n predefined follow-ups F. For each predefined follow-up, the language model computes the log-likelihood D of a follow-up utterance f_i given the dialog history.

$$\sum_{i=1}^{n} D(h, r, f_i) \tag{3.1}$$

The total score is equal to the sum of the individual log likelihoods. It is worth reminding that the metric does not mean anything. It is only useful to *compare* systems together.

3.3.3 Differences with FED

Our implementation differs from FED (Mehri and Eskenazi, 2020a) in multiple ways. First, we do not consider fine-grained attributes, only the overall quality of the turn or dialog.[3]

Second, FED computes the log-likelihood of the conversation history h, the response r, and the follow-up f_i. Whereas we only compute the conditional log-likelihood of the follow-up f_i. Computing the log-likelihood over the conversation introduces a bias towards the dataset used in training the language model, Reddit, in the case of FED. It also favors longer conversations over shorter ones. Our goal is to estimate the likelihood of the follow-up, not the conversation itself.

Third, FED did not justify its choice of follow-ups, while we studied each candidate and only took the most correlated ones making intuitive sense. Fourth, we also study multiple types of language models (conversational and general).

[3]Whereas FED considers 18 fine-grained attributes (overall quality included). Our initial experiments revealed that follow-ups assigned to a fine-grained attribute (e.g., engaging) often had a higher correlation with another unrelated attribute (e.g., correctness). For that reason, we choose to focus on a single attribute, the conversation's overall quality and leave the study of fine-grained attributes for future work.

3.4 Experimental Settings

This section explains our choices of follow-ups, language models, and conversational data. Our goal is to find the combination of language models and follow-ups correlating the most with human evaluations.

3.4.1 Follow-Ups

A follow-up is an utterance added after a conversation's last turn to evaluate the last turn or the entire dialog. FED defined 63 unique follow-ups in 16 categories (fine-grained attributes) at the turn level and the dialog level. Appendix A.2 list the entire list of follow-ups. The authors did not provide any justification for their choice of follow-ups. Instead of blindly using the list of follow-ups, we attempt to understand which of these follow-ups have the highest correlation with human evaluations.

3.4.2 Language Models

We experiment with several language models, both general and conversational. The goal of the language module is to compute the conditional log-likelihood of several follow-ups.

BlenderBot v1 is a conversational sequence-to-sequence model (Roller et al., 2020b) with three sizes: small, large, and extra-large. A distilled version is also available on HuggingFace.

DialoGPT is a conversational language model (Zhang et al., 2020) with three sizes: small, medium and large. The authors fine-tuned a GPT-2 model on a large corpus of Reddit conversations.

GPT-2 is a general language model (Radford et al., 2019). While it was not trained specifically on conversational data, our experiments revealed its potential to estimate a conversation's quality.

Follow-up	Correlation	
	Turn	Dialog
Not really relevant here.	0.48	0.65
You're really confusing.	0.46	0.67
I don't understand what you're saying.	0.46	0.58
That's not really relevant here.	0.45	0.70
You are so confusing.	0.45	0.64
You're really boring.	0.44	0.65
That's not very interesting.	0.44	0.60
That was a really boring response.	0.43	0.63
You don't seem interested.	0.43	0.61
I am so confused right now.	0.43	0.57

Table 3.1: Top 10 follow-ups ranked by Spearman correlation to human evaluations. All follow-ups exhibit a positive relationship, meaning that the likely presence of the follow-up (low log-likelihood) entails a low human evaluation and vice-versa.

3.4.3 Conversational Data

We use the FED dataset (Mehri and Eskenazi, 2020a) for evaluating the set of follow-ups. It consists of 372 turn-level (124 dialog-level), originally collected by Adiwardana et al. (2020b). The dataset consists of human-system conversations (Meena and Mitsuku) and human-human conversations. Mehri and Eskenazi (2020a) asked annotators to evaluate turn-level and dialog-level conversations on several attributes. In this work, we only use the evaluation of the overall quality of the turn or dialog.

3.5 Results

Our objective is to find the best combination of language models and follow-ups. We start by analyzing which language model correlates the most with human evaluation. In the second step, we look for the best set of follow-ups.

3.5.1 Choice of Language Model

We are looking for a language model whose log-likelihood of generating the follow-ups correlates highly with human evaluations. We do so both on a turn-level and dialog-level. We compare the average absolute correlation of each follow-up with human judgments. The results are displayed on Figure A.1 in

Annex A.1. The model standing out is the large Blender model (Roller et al., 2020b). It has the highest correlation with humans both on a turn-level and dialog-level. The difference in performance between Blender-3B and Blender-400M is small. For these reasons, we choose Blender-400M as our default language model.

3.5.2 Choice of Follow-ups

Now that we have identified our model of choice (Blender-400M), we wish to identify the follow-ups correlating the most with humans. We compute the Spearman correlation between each follow-up and human evaluation (turn-level and dialog-level). We present the top-10 follow-ups (by absolute correlation) in Table 3.1. The full table is available Appendix A.2.

The follow-up correlating the most on a turn-level basis is *Not really relevant here* with a Spearman correlation of 0.48. The least correlated follow-up is *Wow! That's really cool!* with correlations of 0.04. The follow-up correlating the most on a dialog-level basis is *That's not really relevant here* with a correlation of 0.70. The least correlated follow-up on a dialog level is *Cool! That sounds super interesting!* with a correlation of 0.01.

Most follow-ups exhibit a positive relationship, meaning that the likely generation of the follow-up by the language model (low log-likelihood) entails a low human rating and vice-versa. However, all the top follow-ups are *negative* follow-ups (e.g., *You're really confusing*), and their likely presence indicates a negative conversation. On the other hand, the *positive* follow-ups (e.g., *Great talking to you*) are not as highly correlated. On average, negative follow-ups correlate with 0.39, while positive follow-ups correlate with 0.24. These results indicate that the language model evaluates a good conversation by the likely absence of negative follow-ups.

Each follow-up brings another forward pass of the model, so ideally, we want to restrict the number of follow-ups in the final evaluation method. For the final selection of follow-ups, we combine the rank of the turn-level and dialog-level correlations and take the top 5.[4] The final selection of follow-ups is the following: *Not really relevant here. You're really confusing. You're really boring. What are you trying to say? You don't seem interested.*

3.5.3 Comparison

Yeh et al. (2021) compared 12 evaluation methods on the FED dataset (Mehri and Eskenazi, 2020a). We compare our method FULL against these 12 other methods

[4]We arbitrarily choose the number 5. We also removed close duplicates. For example *Not really relevant here.* and *That's not really relevant here.*

	Turn Level	Dialog Level
QuestEval	0.09	0.08
MAUDE	-0.09	-0.28
DEB	0.19	-0.01
GRADE	0.12	-0.06
DynaEval	0.32	0.55
USR	0.12	0.06
USL-H	0.19	0.15
DialoRPT	-0.09	-0.21
HolisticEval	0.12	-0.30
PredictiveEngage	0.09	0.15
FED	0.09	0.32
FlowScore	-0.05	-0.00
FULL (ours)	**0.51**	**0.69**

Table 3.2: Comparison of our evaluation method FULL with other automated methods. FULL achieves the highest correlation on turn-level and dialog-level, followed by DynaEval. Except for FULL, results are copied from Yeh et al. (2021).

in Table 3.2. The results are clear, FULL achieves the highest correlation both on a turn-level and dialog-level while being fully unsupervised (except in the choice of follow-ups). By combining the log-likelihood from 5 follow-ups, the average correlation on turn-level increases to 0.51, while the average of the individual correlation equals 0.45.

3.6 Conclusion

This short paper introduces a new automated evaluation method (FULL) for open-domain conversations. FULL measures the quality of a conversation by computing the probability that a language model will continue the conversation with a set of follow-ups (e.g., *Not really relevant here*, *What are you trying to say?*). FULL achieves the highest correlation with human evaluations compared to twelve other existing methods.

Our experiments revealed that negative follow-ups (e.g., *Not really relevant here*) have a higher correlation with human evaluations than positive follow-ups (e.g., *Wow, interesting to know*). It is easier for the model to evaluate a conversation from its bad angles rather than its good ones.

Future work is needed to know which fine-grained attribute can be measured using the same technique. Using ever-large models such as GPT-3 (Brown et al., 2020b) or OPT (Zhang et al., 2022b) could be a direction for future research,

although the resulting model will likely need to be distilled to be of practical use.

Chapter 4

ConveRT for FAQ Answering

This chapter was published in the Proceedings of BNAIC/BeneLearn 2021 : 33rd Benelux Conference on Artificial Intelligence and 30th Belgian-Dutch Conference on Machine Learning, November 10–12, 2021 Belval, Esch-sur-Alzette (Luxembourg) - Luxembourg, BnL, 2021, p. 312-319

4.1 Introduction

In this paper, we present a Dutch-based FAQ retrieval system trained using a limited amount of training data.

FAQ answering is the task of retrieving the right answer given a new user query. It is widely used in chatbots and has been studied for many years (Hammond et al., 1995a; Sneiders, 1999a; Jijkoun and de Rijke, 2005a; Riezler et al., 2007a; Karan and Šnajder, 2016b; Sakata et al., 2019a), although the attention has shifted towards extractive question answering more recently (Rogers et al., 2021a), probably because of a lack of dedicated datasets. FAQ answering systems typically use retrieval systems (Hammond et al., 1995a; Sneiders, 1999a; Jijkoun and de Rijke, 2005a; Riezler et al., 2007a; Karan and Šnajder, 2016b; Sakata et al., 2019a) rather than generative models grounded on external knowledge (Komeili et al., 2021a; De Bruyn et al., 2020; Lotfi et al., 2021). The generative approach is more flexible as it is able to generate new answers. However, these models suffer from knowledge hallucinations (Shuster et al., 2021c), limiting their usefulness in a corporate environment.

Most previous research focusing on FAQ retrieval and non-factoid question answering were developed for English. ConveRT (Henderson et al., 2019a), a response selection module available within Rasa (Bocklisch et al., 2017), caught our attention as it is effective and does not require a GPU at inference time.

Unfortunately, it is only available in English. Despite having significantly less conversational training data (400K pairs of utterances) than the original ConveRT model (727M pairs), we successfully trained the same model for Dutch.

Our contributions are the following:

- We show it is possible to train a ConveRT model for a non-English language using a limited number of conversation pairs by adopting a two-phase pre-training approach (general and conversational).
- We show that a Dutch ConveRT model performs better than the response selector module from Rasa, both in a low and high data regime.

4.2 Related Work

An FAQ dataset consists of pairs of questions and answers. The FAQ retrieval task involves ranking the available answers for a given user query. There are three methods available to solve this problem: matching a new user query on the available questions, the answers, or the concatenation of both. FAQ retrieval can be broadly divided into 4 categories: lexical, supervised, unsupervised, and conversational.

Lexical To our knowledge, FAQ-Finder (Hammond et al., 1995a) was the first to explicitly study the task of FAQ retrieval, it tries to do so by matching user queries to FAQ questions of the Usenet dataset with TF-IDF. FAQ-Finder was later improved by including the similarity to the answer (on top of the similarity to the question) (Tomuro and Lytinen, 2004a). Another improvement comes from adding a rule-based layer on top of the TF-IDF module (Sneiders, 1999a).

Unsupervised Another approach is to used unsupervised techniques to retrieve the right FAQ pair given a new user query. One possible way is to use Latent Semantic Analysis (LSA) to overcome the lexical mismatch between related queries (Kim and Seo, 2008a).

Supervised The first supervised methods were developed using tree kernels and SVMs (Moschitti et al., 2007a). BERT methods were later developed specifically for the task of FAQ retrieval (Sakata et al., 2019a).

Conversational In this paper, we propose a fourth type not yet explored in the literature: conversational. FAQ retrieval can be treated as a special case of conversational modeling: retrieving the answer is similar to retrieving the next utterance in a conversation.

Dual-encoder architectures, pre-trained on response selection, have become increasingly popular in the dialog community due to their simplicity and ease of control (Henderson et al., 2019b; Cer et al., 2018b). There are two options when it comes to retrieving the next utterance. One can either encode the two sentences separately (dual-encoder) (Henderson et al., 2019a), or simultaneously (cross-encoder) (Damani et al., 2020). Dual-encoders are faster than cross-encoders as they can cache the answer representations. ConveRT (Henderson et al., 2019a) is a dual-encoder pre-trained on a large-scale conversational dataset. Thanks to various design optimizations (such as using single-headed self-attention) ConveRT can vastly reduce the size of the model.

In this work, we choose to focus on ConveRT as it has a low computational cost and does not require a GPU for inference.

4.3 ConveRT

In this section, we give a brief overview of the ConveRT (Conversational Representations from Transformers) model (Henderson et al., 2019a). The objective of the model is to generate vector representations for utterances that are as similar as possible (in terms of dot-product) for a given pair. ConveRT takes as input the sequence of tokens of the two utterances. Both sequences are tokenized using the same byte pair encoding vocabulary.

4.3.1 Architecture

The ConveRT architecture (Fig. 4.1) is composed of three distinct parts: the embedding layer, the Transformer block and the feedforward layers.

4.3.1.1 Embedding

The first element stores the embeddings for the subwords and position tokens. Embeddings are shared for the input and response representations. Unlike the original Transformer architecture (Vaswani et al., 2017b), ConveRT uses two positional encoding matrices of different sizes to handle sequences larger than seen

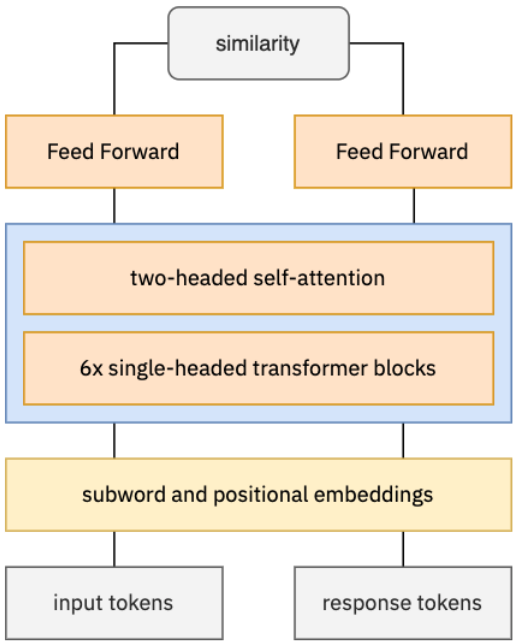

Figure 4.1: Illustration of the ConveRT model architecture. The model has three distinct parts. First, the subword and positional embeddings. Second, a shared Transformer block followed by a two-headed self-attention. Third, separate feed-forward networks (3 layers) for the input and responses.

during training. We refer the reader to the original paper for a detailed description (Henderson et al., 2019a).

4.3.1.2 Transformer Block

The next element is the Transformer block. It closely follows the original Transformer architecture (Vaswani et al., 2017b) with some notable differences. First, the model uses a single-headed self-attention using a 64-dimensional projection for computing the attention weights. Second, the model applies a two-headed self-attention after the six Transformer layers. The parameters of the Transformer block are fully shared for the input and response sides. ConveRT uses the square-root-of-N reduction (Cer et al., 2018b) to convert the embedding sequences to fixed-dimensional vectors.

4.3.1.3 Feed Forward

The last elements are a series of feed-forward hidden layers with skip connections. The parameters are not shared between the inputs and responses side, as there is a separate feed-forward for the inputs and responses.

4.3.2 Training Objective

The training objective of ConveRT is to select the right response given a question from a question-answer pair. The relevance of each response to a given input is quantified with a dot-product between the input and response representation. Training proceeds in a batch of K pairs of utterances. The objective is to distinguish between the true relevant responses and irrelevant negative examples (we use other responses from the batch as negative examples). ConveRT uses cross-entropy as the loss function. The model is optimized with Adam (Kingma and Ba, 2015) and L2 weight decay. The learning rate is warmed up over the first 10,000 steps to a peak value and then linearly decayed.

4.4 ConveRT for Dutch

In this section, we explain our approach to training a ConveRT model for Dutch. To overcome the limited supply of conversational data available in Dutch, we use a two-stage pre-training: general pre-training on a large open-domain corpus, and conversational pre-training using a smaller conversational dataset from Reddit.

4.4.1 Data

The original ConveRT model was developed for English using a large-scale conversational dataset from Reddit. We did not have access to such a dataset for Dutch. Instead, we chose to split the problem in two. First, we pre-train the model on a general Dutch corpus. Second, we use a smaller Dutch conversational corpus from Reddit.

4.4.1.1 General Dataset

We consider the same Dutch-language corpora as Bertje (de Vries et al., 2019), a successful Dutch BERT model:

- Books: a collection of contemporary and historical fiction novels
- TwNC (Ordelman et al., 2007): a Multifaceted Dutch News Corpus
- SoNaR-500 (Oostdijk et al., 2013): a multi-genre reference corpus
- Web news
- Wikipedia

In total, this is about 12GB of uncompressed text.

To match the setup expected by ConveRT (the tokens of a pair of utterances), we first split each paragraph into sentences. Next, we save pairs of sentences and treat them as pairs of input and response. To avoid small inputs, we filter out pairs with less than 64 characters. After transformation, the general corpus dataset for pre-training has 110M pairs.

4.4.1.2 Conversational Dataset

We also consider a Dutch conversational dataset for which we downloaded comments from around 200 Dutch subreddits. Non-Dutch comments were filtered out. After filtering for the language we arrive at a size of 400K pairs of utterances.

4.4.2 Pre-training

We followed the training procedure of ConveRT, except for the number of epochs and the batch size. For the general pre-training, we trained the model for 8 epochs. To facilitate the training, we used other examples from the batch as negative examples.

To increase the difficulty of the training, we doubled the batch size at every second epoch. The batch size increased from 128 at the first epoch to 2048 at the last epoch. The larger the batch size, the harder it is for the model as the model has to select the correct response amongst more negative examples.

For the conversational pre-training, we trained for 10 epochs with a fixed batch size of 2048.

model	split 1	split 2	split 4	split 6	split 8	split 10
RASA (baseline)	22%	42%	50%	55%	61%	65%
without pre-training	20%	25%	33%	45%	52%	65%
general pre-training	30%	36%	40%	55%	58%	43%
conversational pre-training	40%	44%	55%	63%	66%	69%
general + conv. pre-training	**46%**	**57%**	**68%**	**69%**	**75%**	**79%**

Table 4.1: Accuracy on the COVID-19 vaccination FAQ dataset per splits of increasing size. Split one has one training example per answer, while split ten has ten training examples. Pre-training ConveRT on both a general dataset, as well as a conversational dataset provides the best results on this task.

4.5 Experiments

In this section, we fine-tune our model on a corpus of FAQs related to the COVID-19 vaccine. We then perform an ablation study to analyze which part of the pre-training has the most impact on the downstream performance. To have a better understanding of how our model would perform in the real world, we study its performance as the number of training examples increases.

4.5.1 Data

We test the performance of our model on a proprietary dataset. The dataset was collected while running a COVID-19 vaccination FAQ bot with Rasa. It consists of 1,200 questions for 76 distinct answers.

4.5.2 Baseline

As our higher objective is to use this model in a Rasa chatbot, we compare our Dutch ConveRT model to a baseline response retrieval model developed by Rasa.[1] All models are trained using the same number of epochs and dropout probability.

4.5.3 Low Data Scenario

When starting out, FAQ bots usually have a one-on-one mapping between the number of questions and answers (one question for one answer). As the number of users increases, the number of available questions per answer also increases.

[1]Rasa does not have a published paper describing their model.

To evaluate the generalization capabilities of our model in a low data scenario, we artificially create datasets of increasing sizes, which we call splits. The first split has one training example per answer (the same as when someone starts a new FAQ chatbot), the second split has two training examples per answer, and so on until split ten. We also generate a test set by randomly selecting (and removing from the training set) one training example per answer.

4.5.4 Results

Results in Table 4.1 confirm our intuition that the baseline accuracy of the Rasa model radically improves with the number of training examples. In our analysis, the accuracy increases by a factor of 3 from split 1 to split 10. The results also show that a ConveRT model without any pre-training underperforms the baseline, on every split. General pre-training modestly improves the model's performance, but the results are not significantly different from the baseline. Conversational pre-training alone (without any general pre-training) shows a consistent improvement over the baseline. The gain is more visible in the low data regime than in the high data regime. The Dutch ConveRT model reveals its true power when pre-trained on a general corpus and a conversational corpus as it outperforms the baseline by a wide margin on every split.

4.6 Conclusion

We have successfully pre-trained, fine-tuned, and evaluated a Dutch ConveRT model. This model consistently outperforms a baseline response selector from Rasa on a COVID-19 vaccine FAQ dataset.

Conversational datasets for non-English languages are scarce. Our two-phase pre-training procedure bypasses this problem by first pre-training on a general corpus, then pre-training on a smaller conversational corpus.

In future work, we plan on extending the two-stage training to additional languages and additional domains.

Chapter 5

MFAQ: a Multilingual FAQ Dataset

This chapter was published in the Proceedings of the 3rd Workshop on Machine Reading for Question Answering - ISBN 978-1-954085-95-4 - Association for Computational Linguistics, (2021), p. 1-13

5.1 Introduction

Organizations create *Frequently Asked Questions* (FAQ) pages on their website to provide a better service to their users. FAQs are also useful to automatically answer the most frequent questions on different communication channels: email, chatbot, or search bar.

FAQ retrieval is the task of locating the right answer within a collection of candidate question and answer pairs. It is closely related to the tasks of *non-factoid QA* and *community QA*, although it has its own specificities. The total number of possible answers is generally small (the average FAQ page on the web has 6 answers), and only one is correct. Retrieval systems cannot rely on named entities, as they are typically shared by many possible answers. For example, three out of four answers in Table 5.1 share the *COVID-19* entity. Lastly, new user queries are matched against pairs of questions and answers, as opposed to passages for non-factoid QA.

Since FAQ-Finder (Hammond et al., 1995b), researchers applied different methods to the task of FAQ retrieval (Sneiders, 1999b; Jijkoun and de Rijke, 2005b; Riezler et al., 2007b; Karan and Šnajder, 2016a; Sakata et al., 2019b). However, since the advent of deep-learning and Transformers, the interest has somewhat faded compared to other areas of QA (Rogers et al., 2021b). One possible explanation is the lack of a dedicated large-scale dataset. The ones available are mostly limited to English, and domain-specific.

Example FAQs
Is it safe for my child to get a COVID-19 vaccine? Yes. Studies show that COVID-19 vaccines are safe and effective. [...]
If I am pregnant, can I get a COVID-19 vaccine? Yes, if you are pregnant, you can receive if COVID-19 vaccine.
What are the ingredients in COVID-19 vaccines? Vaccine ingredients can vary by manufacturer.
How long does protection from a COVID-19 vaccine last? We don't know how long protection lasts for those who are vaccinated. [...]

Table 5.1: Example FAQs about the COVID-19 vaccine from the CDC website.

On the other hand, the task of factoid question answering received the attention of many researchers. Recently, Transformers encoders such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) have been successfully applied to the retrieval part of factoid QA, overcoming strong baselines such as TF-IDF and BM25. However, we show that DPR's performance on passage retrieval is not directly transferable to FAQ retrieval. Lewis et al. (2021b) recently released PAQ, a dataset of 65M pairs of *Probably Asked Questions*. However, answers are typically short in PAQ (a few words), which differs from FAQs where answers are longer than questions.

Another way to answer users' questions is to use *Knowledge Grounded Conversation* models as it does not require the pre-generation of all possible pairs of questions and answers (Komeili et al., 2021b; De Bruyn et al., 2020). However, at the time of writing these models can hallucinate knowledge (Shuster et al., 2021b), which limits their attractiveness in a corporate environment.

In this paper, we provide the first multilingual dataset of FAQs. We collected around 6M FAQ pairs from the web in 21 different languages. This is significantly larger than existing datasets. However, collecting data from the web brings its own challenges: duplication of content and uneven distribution of topics. We also provide the first multilingual FAQ retriever. We show that models trained on all languages at once outperform monolingual models (except for English).

The remainder of the paper is organized as follows. We first review the existing models and datasets available for the task of FAQ retrieval. We then present our own dataset and apply different models to it. We finally perform some analysis on the results and conclude. Our dataset and model are available on the HuggingFace Hub[1].

[1]dataset, model and training script

5.2 Related Work

In this section, we review the existing literature on FAQ retrieval. We first start by reviewing available models and then look at the available datasets.

5.2.1 Models

Since the release of FAQ-Finder (Hammond et al., 1995b; Burke et al., 1997) and Auto-FAQ (Whitehead, 1995), several methods have been presented. We grouped them into three categories: lexical, unsupervised, and supervised.

Lexical FAQ-Finder (Hammond et al., 1995b; Burke et al., 1997) matches user queries to FAQ questions of the Usenet dataset using Term Frequency-Inverse Document Frequency (TF-IDF). The system tries to bridge the lexical gap between users' queries and FAQ pairs by using the semantic network *WordNet* (Miller, 1995a) to establish correlations between related terms. FAQ-Finder assumes that the question half of the QA pair is the most relevant for matching to a new query. Tomuro and Lytinen (2004b) improved upon FAQ-Finder by including the other half of the QA pair (the answer). Xie et al. (2020) uses a knowledge graph-based QA framework that considers entities and triples in texts as knowledge anchors. This approach requires the customization of a knowledge graph, which is labor-intensive and domain-specific.

Sneiders (1999b) used a rule-based technique called Prioritized Keyword Matching on top of a traditional TF-IDF approach. The use of shallow language understanding means that the matching is based on keyword comparison. Each FAQ entry must be manually annotated with a set of required and optional keywords. Sneiders (2002, 2009, 2010) brings further developments on the idea. Moreo et al. (2013) proposes an approach based on semi-automatic generation of regular expression for matching queries with answers. Yang (2009) integrates a domain ontology, user modeling, and a template-based approach to tackle this problem.

Unsupervised Kim and Seo (2008b, 2006) presented a clustering-based method of previous user queries to retrieve the right FAQ pair. The authors used a Latent Semantic Analysis (LSA) method to overcome the lexical mismatch between related queries. Jijkoun and de Rijke (2005b) experimented with several combinations of TF-IDF retrievers based on the indexing of different fields (question, answer, with or without stop words, the full text of the page). Riezler et al. (2007b) extended this method by incorporating a translation-based query expansion, as initially investigated in Berger et al. (2000).

Name	Size	Lang.	Domain	Source	Q>1	A>1
Usenet (Hammond et al., 1995b)	-	En	Multi-domain	Usenet	No	No
FAQIR (Karan and Šnajder, 2016a)	4,313	En	Maintenance	Yahoo! Answers	Yes	Yes
StackFAQ (Karan and Šnajder, 2018)	719	En	Web apps	StackExchange	Yes	Yes
InsuranceQA (Feng et al., 2015)	12,887	En	Insurance	Insurance Library	No	Yes
CQA-QL (Nakov et al., 2015)	2,600	En	Qatar	Qatar living forum	No	Yes
Fatwa corpus (Nakov et al., 2015)	1,300	Ar	Quran	Fatwa website	No	Yes
M-FAQ (ours)	6,134,533	Multi	Multi	Multi	No	No

Table 5.2: List of the common datasets used in FAQ retrieval. Size is the number of pairs available. Q>1 denotes if the dataset has multiple available questions for a single answer (i.e., does the dataset have paraphrases), while A>1 denotes if the dataset has multiple answers for a given question.

Supervised Moschitti et al. (2007b) proposed an approach based on tree kernels. Tree kernels can be defined as similarity metrics that compare a query to an FAQ pair by parsing both texts and calculating the similarity based on the resulting parse trees. Semantic word similarity can also be added to the computation. Filice et al. (2016) expanded on this method and achieved first place in the Community QA shared task at SemEval 2015 (Nakov et al., 2015).

Sakata et al. (2019b) were the first to use BERT-based models (Devlin et al., 2018) for the specific task of FAQ retrieval. The relevance between the query and the answers is learned with a fine-tuned BERT model which outputs probability scores for a pair of (query, answer). The scores are then combined using a specific method. Mass et al. (2020) also used a BERT model. Their method is based on an initial retrieval of FAQ candidates followed by three re-rankers. Bruyn et al. (2021) used a ConveRT (Henderson et al., 2019a) model to automatically answer FAQ questions in Dutch.

5.2.2 Datasets

In this section, we review the different datasets publicly available. FAQ retrieval datasets can be evaluated on four axes: source of data (community or organizational), the existence of user queries (paraphrases), domain, and language. See Table 5.2 for an overview.

Faq-Finder (Hammond et al., 1995b; Burke et al., 1997) used a dataset collected from Usenet news groups. FAQs were created on several topics so that newcomers do not ask the same questions again and again. This dataset is multi-domain. More recently, Karan and Šnajder (2016a) released the FAQIR dataset. It was collected from the "maintenance & repairs" section of the QA website *Yahoo! Answers*. The StackFAQ (Karan and Šnajder, 2018) dataset was collected from the "web apps" sections of *StackExchange*. Feng et al. (2015) collected a QA dataset from the insurancelibrary.com website where a community of insurance expert

reply to users' questions. Several authors (for example Filice et al., 2016) also rely on Sem-Eval 2015 Task 3 (Nakov et al., 2015) on Answer Selection in Community Question Answering. It contains pairs of questions and answers in English and Arabic.

There exist few publicly available datasets for organizational FAQs. OrgFAQ (Lev et al., 2020) is a notable exception. At the time of writing, it is not yet publicly available.

5.3 Multilingual FAQ dataset

In this section, we introduce our new multilingual FAQ dataset.

5.3.1 Data collection

Instead of implementing our own web crawler, we used the Common Crawl: a non-profit organization which provides an open repository of the web.[2] Common Crawl's complete web archive consists of petabytes of data collected over 10 years of web crawling (Ortiz Suárez et al., 2020). The repository is organized in monthly bucket of crawled data. Web pages are saved in three different formats: WARC files for the raw HTML data, WAT files for the metadata, and WET files for the plain text extracts.

For our purposes, we used WARC files as we are interested in the raw HTML data. Similar to Lev et al. (2020), we looked for *JSON-LD*[3] tags containing an *FAQPage* item. Web developers use this tag to make it easy for search engines to parse FAQs from a web page.[4] The language of each FAQ pair is determined with fastText (Joulin et al., 2016). We also apply some filtering to remove unwanted noise.[5] Using this method, we collected 155M FAQ pairs from 24M different pages.

[2]https://commoncrawl.org/about/

[3]JavaScript Object Notation for Linked Data

[4]More information on FAQPage markup

[5]Questions need to contain a question mark (including the Arabic question mark) to avoid keyword questions. Question and answer cannot start with a "<", "{", or "[" to remove "code like" data.

5.3.2 Deduplication

A common issue with datasets collected from the web is the redundancy of data (Lee et al., 2021). For example, hotel pages on *TripAdvisor* typically have an FAQ pair referring to shuttle services from the airport to the hotel.[6] The only changing term is the name of the hotel.

Algorithms such as SimHash (Charikar, 2002) and MinHash (Broder, 1997) can detect such duplicates. MinHash is an approximate matching algorithm widely used in large-scale deduplication tasks (Lee et al., 2021; Versley and Panchenko, 2012; Gabriel et al., 2018; Gyawali et al., 2020). The main idea of MinHash is to efficiently estimate the Jaccard similarity between two documents, represented by their set of n-grams. Because of the sparse nature of n-grams, computing the full Jaccard similarity between all documents is prohibitive. MinHash alleviates this issue by reducing each document to a fixed-length hash which can be used to efficiently approximate the Jaccard similarity between two documents. MinHash has the additional property that similar documents will have similar hashes, we can then use Locality Sensitive Hashing (LSH) (Leskovec et al., 2014) to efficiently retrieve similar documents.

In our experiments, we represented each page as a set of 3 consecutive tokens (n-grams). We worked with a document signature length of 100, and 20 bands with 5 rows as parameters for LSH. These parameters ensure a 99.6% probability that documents with a Jaccard similarity of 0.75 will be identified. We subsequently compute the true Jaccard similarity for all matches.

We follow the approach of *NearDup* (Lee et al., 2021) and subsequently create a graph of documents. Each node on the graph is an FAQ page, and they share an edge if their true Jaccard similarity is larger than 0.75. We then compute all the independent sub-graphs, each representing a graph of duplicated pages. We only keep one page per sub-graph.

Using this method, we trimmed the number of FAQ pages from 24M to 1M.

5.3.3 Description

After deduplication, our dataset contains around 6M FAQ pairs coming from 1M different web pages, spread on 26K root web domains.[7] This is significantly

[6] Does Ritz Paris have an airport shuttle? Does Four Seasons Hotel George V have an airport shuttle?

[7] We define a root web domain as the last substring before the extension (e.g. TripAdvisor is the root web domain in fr.tripadvisor.com). In other words, we strip the extension and any subdomain.

bigger than other FAQ datasets publicly available at the time of writing (see Table 5.2 for comparison).

Our dataset is composed of pairs of FAQs grouped by language and source page (URL). We collected data in 21 different languages.[8] The most common one is English, with 58% of the FAQ pairs, followed by German and Spanish with 13% and 8% respectively.

5.3.4 Training and validation sets

For a given language, the target size of the validation set is equal to 10% of the total number of pairs. However, two features of our dataset call for a more fine-grained approach.

5.3.4.1 Root domain distribution

Even though we deduplicated the dataset, FAQ pages tend to originate from the same root domain. As an example, kayak (*kayak.com*, *kayak.es*, etc.) is the largest contributor to the dataset. While this is not a problem for the training set (one can always restrict the number of pages per domain), it is an issue for the validation set, as we want to assess the quality of the model on a broad set of topics. Having several large root domain contributors skews the dataset to these topics. We make the simplifying assumption that different web domains have different topics of interest. Research on the true topic distribution is left for future work.

We artificially increased the topic breadth of the validation set by restricting the contribution of each root domain. In the validation set, a single root domain can only contribute up to 3 FAQ pages. This method reduces the contribution of the largest domain from 21% in the training set to 3% in the validation set. Furthermore, we make sure there is no overlap of root domain between the training and validation set.[9]

[8]We did not target specific languages, however, we removed languages with fewer than 250 pairs. Common languages such as Chinese, Hindi, Arabic and Japanese are missing. Although we do not have an official reason why, we think it may be because of our initial filtering or the fact ldjson markup is not widely used in these languages.

[9]We use the root domain instead of the regular domain name to avoid having *help.domain.com* in the training set and *domain.co.uk* in the validation set

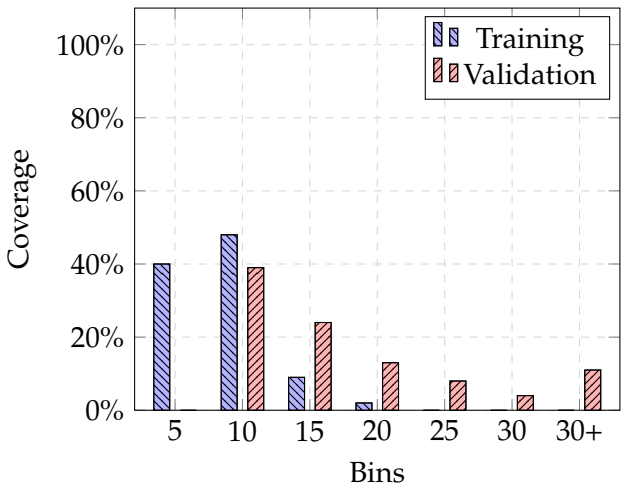

Figure 5.1: Bucketing of our dataset according to the number of FAQs per page. To make the validation set more challenging, we started by selecting pages with a higher number of pairs.

5.3.4.2 Pairs per page concentration

The distribution of the number of pairs per page is highly uneven (see Figure 5.1). Around 50% of the pages have 5 or fewer pairs per page. Intuitively, we prefer pages with a higher number of FAQs as it is harder to pick the right answers amongst 100 candidates than 5. We thus artificially increased the difficulty of the validation by first selecting pages with a higher number of FAQ pairs per page. See Figure 5.1 for a comparison between the training and validations set.

5.3.4.3 Cross-lingual leakage

The fact that our dataset is multilingual can lead to issues of cross-lingual leakage. Having pages from *expedia.fr* in the training set, and pages from *expedia.es* in the validation set can overstate the performance of the models. We avoid such problems by restricting root domains in the validation associated with only one language (e.g. *expedia* would be excluded from the validation set because it is associated with French and Spanish pages).

Language	**Pairs**	**Pages**	**Domains**
English	3,719,484	608,796	17,635
German	829,098	117,618	2,948
Spanish	482,818	75,489	1,610
French	351,458	56,317	1,795
Italian	155,296	24,562	685
Dutch	150,819	32,574	1,472
Portuguese	138,778	26,169	608
Turkish	102,373	19,002	580
Russian	91,771	22,643	953
Polish	65,182	10,695	445
Indonesian	45,839	7,910	309
Norwegian	37,711	5,143	198
Swedish	37,003	5,270	434
Danish	32,655	5,279	362
Vietnamese	27,157	5,261	469
Finnish	20,485	2,795	234
Romanian	17,066	3,554	152
Czech	16,675	2,568	182
Hebrew	11,212	1,921	205
Hungarian	8,598	1,264	150
Croatian	5,215	819	99
Total	6,346,693	1,035,649	31,525

Table 5.3: Summary statistics about our dataset.

5.4 Models

In this section, we describe the FAQ retrieval models used in our experiments. Let P be the set of all user queries and $F = \{(q_1, a_1), ..., (q_n, a_n)\}$ be the set of all FAQ pairs for a given domain. An FAQ retrieval model takes as input a user's query $p_i \in P$ and an FAQ pair $f_j \in F$, and outputs a relevance score $h(p_i, f_j)$ for f_j with respect to p_i. However, our dataset does not contain live user queries (or paraphrases) P, we thus use questions q as queries $P = \{q_1, ..., q_n\}$ and restrict the FAQ set to the answers $F = \{a_1, ..., a_n\}$. The task becomes to rank the answers A according to the questions Q.

5.4.1 Baselines

We experimented with several baselines: two unsupervised and one supervised.

5.4.1.1 TF-IDF

The traditional information retrieval method (Salton et al., 1975) uses a vector representation for q_i and a_i and computes a dot-product as similarity relevance score $h(q_i, a_i)$. We use n-grams of size (1, 3) and fit one model per FAQ page.

5.4.1.2 Universal Sentence Encoder

Encoding the semantics of a question q_i and an answer a_i can be achieved with the Universal Sentence Encoder (Cer et al., 2018a). The model works on monolingual and multilingual data. We encode each question and answer independently, and then perform a dot-product of the questions' and answers' representations.

5.4.1.3 Dense Passage Retrieval (DPR)

Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) is a state of the art method for passage retrieval. It uses a bi-encoder to encode questions and passages into a shared embedding space. We fine-tune DPR on our dataset using the same procedure described in Section 5.4.2.2.

5.4.2 XLM-Roberta as bi-encoders

Bi-encoders encode questions q_i and answers a_i independently and output a fixed d-dimensional representation for each query and answer. The encoder can be shared or independent to generate the representations.[10] At run-time, new queries are encoded using the encoder, and the top-k closest answers are returned. The representations for the answers can be computed once, and cached for later use. Similarity is typically computed using a dot product.

5.4.2.1 Multilingual

The state-of-the-art encoders such as RoBERTa (Liu et al., 2019a) and BERT (Devlin et al., 2018) are trained for English only. As our dataset is multilingual we opted for XLM-RoBERTa (Conneau et al., 2019), it was trained using masked language modeling on one hundred languages, using more than 2TB of filtered CommonCrawl data. This choice allows us to leverage the size of the English data for less represented languages.

5.4.2.2 Training

Given pairs of questions and answers, along with a list of non-relevant answers, the bi-encoder model is trained to minimize the negative log-likelihood of picking the positive answer amongst the non-relevant answers. Non-relevant answers can be divided into *in-batch* negatives and *hard* negatives.

In-batch negatives In-batch negatives are the other answers from the batch, including them into the set of non-relevant answers is extremely efficient as their representations are already computed.

Hard negatives Hard negatives are close but incorrect answers to the questions. Including them improves the performance of retrieval models (Karpukhin et al., 2020; Xiong et al., 2020). Hard negatives can either come from a standard retrieval system such as BM25, or an earlier iteration of the dense model (Xiong et al., 2020; Oğuz et al., 2021). The structure of our dataset, pages of FAQs, facilitates the search for hard negatives. As an example in Table 5.1, three out of four answers share the term *COVID-19*. The model now has to understand the semantic of sentences instead of matching on shared named entities. By including all the pairs

[10] We use a shared encoder, which means we use the same network to compute the representation for questions and answers. DPR uses independent encoders.

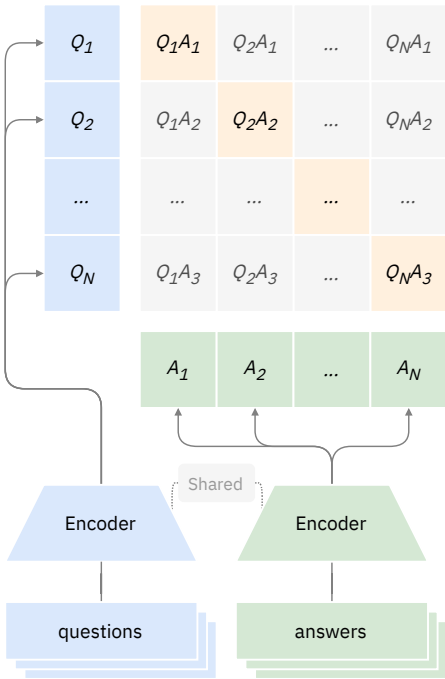

Figure 5.2: Diagram of our architecture. A shared encoder encodes the questions and the answers independently. Each question's representation (vector) is compared to each answer's representation from the same batch using a dot-product.

of the same page in the same training batch, we ensure that in-batch negatives act as hard negatives.[11]

Multilingual Although XLM-Roberta is multilingual, we do not expect the model to perform cross-lingual retrieval (i.e. using one language for the query and another for the answer). We make sure that each batch is composed of pairs from the same language. This increases the difficulty of the task. Otherwise, the model could rely on the language of answers as a differentiating factor.

5.5 Experiments

In this section, we evaluate the retrieval performance of our model on MFAQ. In all our experiments, we use three metrics to evaluate the performance: precision-at-one (P@1), mean reciprocal rank (MRR), and recall-at-5 (R@5). For space reasons,

[11]To create our batches of training data, we incrementally augment the batch with pairs of a given page. When the batch size reaches the desired size, we start over with the remaining pairs.

Language	**TF-IDF**	**USE**	**XLM-R**
English	63.8	64.2	**82.5**
German	58.0	61.8	**81.3**
Spanish	60.5	61.6	**81.7**
French	60.6	62.4	**80.7**
Italian	58.6	55.7	**74.7**
Dutch	62.9	59.6	**81.2**
Portuguese	55.8	56.2	**77.4**
Turkish	59.2	55.7	**78.8**
Russian	59.2	63.1	**82.1**
Polish	59.2	59.9	**85.2**
Indonesia	71.3	62.1	**88.5**
Norwegian	58.9	36.9	**83.1**
Swedish	59.3	36.7	**83.3**
Danish	64.0	42.1	**82.7**
Vietnamese	73.3	43.2	**81.2**
Finnish	53.5	33.2	**82.6**
Romanian	57.8	40.7	**83.2**
Czech	48.2	26.9	**69.0**
Hebrew	61.5	26.5	**83.6**
Hungarian	38.1	28.6	**69.7**
Croatian	58.1	41.4	**83.6**

Table 5.4: MRR on MFAQ using various methods. XLM-RoBERTa is a single model trained on all languages at once.

we only report on MRR in the main text, the full results are available in the annex. We used the same parameters for all experiments unless mentioned otherwise.[12] We insert a special token <question> before questions to let the shared encoder know it is encoding a question. Answers are respectively prepended with <answer>. All of our experiments use a subset of the training set: only one page per domain as this technique achieves higher results. Refer to Section 5.5.3 for more information.

We start by studying the performance of multilingual models, then compare it against monolingual models.

5.5.1 Multilingual

We present in Table 5.4 a summary of the results of our multilingual training. The model is trained concurrently on the 21 available languages. XLM-RoBERTa achieves a higher MRR on every language compared to the baselines. Low resource languages achieve a relatively high score which could indicate inter-language transfer learning.

5.5.2 Monolingual

Next, we attempt to study if a collection of monolingual models are better suited than a single monolingual model. We use language-specific BERT-like models for each language. The list of BERT models per language is available in the annex. We followed the same procedure as described in Section 5.4.2, except for the encoder which is language-specific.

We limited our study of monolingual models to the ten largest languages of MFAQ. We choose these languages as they have sufficient training examples, and pre-trained BERT-like models are readily available. To study the performance of monolingual models we train models using the same procedure as described in Section 5.4.2 except for the encoder.

The results in Table 5.5 indicates that a multilingual model outperforms monolingual models in all cases, except for English. These results indicate that leveraging additional languages is beneficial for the task of FAQ retrieval, especially for languages with fewer resources available. Interestingly, RoBERTa slightly beats DPR in English. This underperformance could be explained by the difference in batch

[12]We used a batch size of 800, sequences were limited to 128 tokens (capturing the entirety of 90% of the dataset), an Adam optimizer with a learning rate of 0,0001 (warmup of 1000 steps). Dropout of 25%.

Language	Monoligual	Multilingual
En. (DPR)	80.5	82.5
En. (RoBERTa)	**82.9**	82.5
German	81.1	**81.3**
Spanish	78.0	**81.7**
French	71.0	**80.7**
Italian	64.1	**74.7**
Dutch	70.4	**81.2**
Portuguese	68.4	**77.4**
Turkish	76.1	**78.8**
Russian	71.6	**82.1**
Polish	73.9	**85.2**

Table 5.5: MRR of monolingual models versus a single multilingual model. The multilingual model outperforms monolingual models in all languages, except for English.

Language	Cross-lingual	Multilingual
French	78.2	80.7
Hungarian	65.9	69.7
Croatian	71.2	83.6

Table 5.6: MRR results of our cross-lingual analysis. Questions were translated to English while answers remained in the original language.

size. Because of the dual encoder nature of DPR, we had to reduce the batch size to 320 compared to 800 for RoBERTa.

5.5.3 Cross-lingual

Our training procedure ensures that the model never has to use language as a cue to select the appropriate answer. Batches of training data all share the same language. We tested the cross-lingual retrieval capabilities of our multilingual model by translating the queries to English while keeping the answers in the original language. The French performance drops from 80.7 to 78.2, which is still better than the unsupervised baselines. The full results are presented in Table 5.6.

A subsectionSubset of training data We tested the effect of limiting the number of FAQ pages per domain by limiting the training set to one page per web domain. Using this technique, we achieved an average MRR of 80.8 while using all the training data to reach an average MRR of 76.7. Filtering the training set flattens the topic distribution and better matches the validation set. Another possible

approach is to randomly select a given page from a domain at each epoch. This technique would act as a natural regularization. This is left for future work.

5.6 Qualitative analysis

In this section, we dive into the model's predictions and try to understand why and where it goes wrong. We do so by focusing on a single FAQ page from the admission center of the *Tepper School of Business*.[13] The FAQs are displayed in Table B.2 in the annex. The multilingual model is correct on 74.07% of the pairs, with an MRR of 85.49. Our qualitative analysis reveals that the model is bad at coreference resolution and depends on keywords for query-answer matching.

Coreference Resolution The model makes a wrong prediction in question 4 *Can the GMAT or GRE requirement be waived? No, these test scores are required*. The model is unable to guess that *test scores* refer to GMAT or GRE. By changing the answer to *No, the GMAT or GRE scores are required*, the model correctly picks the right answer.

Paraphrase To study if the model is robust to paraphrasing, we change question 1 from « *Are the hours flexible enough for full-time working adults?* » to « *Is it manageable if I already have a full-time job?* » In this case, the model correctly identifies the right answer. However, if we remove the *full-time* cue, the right answer only arrives in the fourth position. Next, we look at question 15, the model makes a wrong prediction as *opportunities* is not mentioned in the answer. Changing the question to « *It's a part-time online program, but are there any on-campus [experiences | activities] for students?* » leads to a correct prediction.[14]

Keyword search We replace some questions with a single keyword. We reduced questions 12, 14, 16 and 20 to *cohort*, *payment plan*, *soldier veteran* and *technical requirements*. In all cases, the model guessed correctly, showing the model can do a keyword-based search.

Although it can cope with some synonyms (activities - experiences), this qualitative analysis shows our model is overly reliant on keywords for matching questions and answers. Further research on adversarial training of FAQ retrieval is needed.

[13] It was the first page with less than 25 pairs to end with a *.edu* extension.

[14] replacing *opportunities* with *events* does not work.

5.7 Future Work

Important non-Indo-European languages such as Chinese, Hindi, or Japanese are missing from this dataset. Future work is needed to improve data collection in these languages. Second, we did not evaluate the model on a real-life FAQ retrieval dataset (with user queries). Future work is needed to see if our model can perform question-to-question retrieval, or if it needs further training to do so. A linguistic study could analyze the model's strengths and weaknesses by studying the model's performance by type of questions, answers, and entities.

5.8 Conclusion

In this work, we presented the first multilingual dataset of FAQs publicly available. Its size and breadth of languages are significantly larger than other datasets available. While language-specific BERT-like models can be applied to the task of FAQ retrieval, we showed it is beneficial to use a multilingual model and train on all languages at once. This method of training outperforms all monolingual models, except for English. Our qualitative analysis reveals our model is overly reliant on keywords to match questions and answers.

Chapter 6

Machine Translation for Multilingual Intent Detection and Slots Filling

This chapter was published in the Proceedings of the Massively Multilingual Natural Language Understanding Workshop (MMNLU-22) - ISBN 978-1-959429-15-9 - Stroudsburg, Association for Computational Linguistics, (2022), p. 69-82

6.1 Introduction

Home assistants are omnipresent in everyday life. We expect to have an assistant at our disposal at any time using our phone, watch, or car — irrespective of our language.

Scaling home assistants to multiple languages brings additional challenges to NLU and ASR components. There are two options: a single model per language or a shared model for all languages. A single model per language works well for resource-rich languages such as English. However, lower resource languages benefit from the cross-lingual knowledge transfer of a single model dealing with all languages (Conneau et al., 2020). This trade-off applies to any multilingual system (Zhang et al., 2022a; De Bruyn et al., 2021).

While multilingual intent classification and slot filling datasets exist, their language coverage is limited, except for MASSIVE (FitzGerald et al., 2022), a new dataset focused on multilingual intent detection and slot filling. The authors translated and localized an English-only dataset in 50 topologically diverse languages. MASSIVE provides a good base to scale existing intent detection and slot filling methods to multiple languages.

The traditional way to tackle multilingual intents detection and slot filling is to use multilingual models such as XLM-R (Conneau et al., 2020), or mT5 (Xue

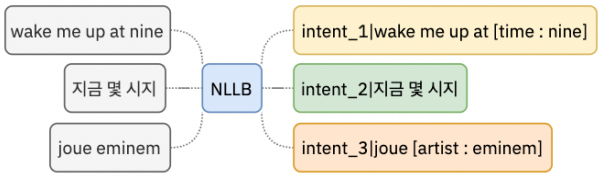

Figure 6.1: Illustration of our method. We repurpose a translation model for the task of multilingual intent classification and slot filling. We translate from utterances into annotated utterances.

et al., 2021). These models are similar to their monolingual counterparts (Liu et al., 2019b; Raffel et al., 2020a) except for the multilingual data used to train them.[1] This approach has been shown to work in multiple studies (FitzGerald et al., 2022; Li et al., 2021a). However, MASSIVE has an additional overlooked aspect: utterances are direct translations of one another.

In this work, we approach the task of intent classification and slot filling as a translation task: we translate the original utterance into the annotated utterance. For example, we translate the utterance `what is the temperature in new york?` into the annotated utterance `weather_query|what is the [weather_descriptor : temperature] in [place_name : new york]`.[2]

The typical use of translation models for intent detection and slot filling is to augment the size of an existing dataset (Zheng et al., 2021; Nicosia et al., 2021). However, we believe the inherent multilingual capabilities of these models make them excellent candidates for multilingual intent detection and slot filling.

To this end, we leverage the recently released translation model *No Language Left Behind* (NLLB) (NLLB Team et al., 2022) capable of translating between 202 pairs of languages simultaneously using a shared encoder-decoder. We anticipate that the wide range of languages covered by the model will help us deal with lower resources languages present in the MASSIVE dataset.

Better modeling is only half the story. Using more data also helps improve performance. For example, although the MASSIVE dataset displays a large training set of more than 500K training examples, the seed data is only around 10K training examples. Therefore, we used GPT-3 (Brown et al., 2020a) to generate additional training data using a dual-model approach. We also leveraged a dataset close to the seed dataset of MASSIVE. As a result, after translating our new training examples to the 50 remaining languages, our training set contains more than 2M training examples — 4x the size of the original training set.

[1] They also have larger vocabularies and may have special training tricks for cross-lingual training.

[2] We prepend the slot annotated utterance with the intent.

Our experiments reveal that translation models such as NLLB are a good fit for intent classification and slot filling. However, their performance sharply drops in languages that do not use spaces because of tokenization issues.

Unfortunately, the additional training data significantly overlaps with the MASSIVE test set. As a result, we propose two methods capable of dealing with overlaps: weighted exact match and logistic regression.

We conclude this introduction by summarizing our contributions:

- We showed that a translation model such as NLLB can complete the task of intent classification and slot filling
- We demonstrated a method to improve the training data with GPT-3
- We proposed two new evaluation methods taking the training/test set overlap into account

We release our model[3], utterance translation model[4], and generated data[5] on the HuggingFace hub.

6.2 Related Work

The problem of multilingual intent detection and slot filling is not new. (Razumovskaia et al., 2022) provides an excellent introduction to the subject. We divide our related work section into three parts. We start by reviewing the general problem of task-oriented semantic parsing (i.e., intent detection and slot filling). Next, we review the models commonly used, and lastly, we review the available multilingual datasets.

6.2.1 Task Oriented Semantic Parsing

Natural Language Understanding (NLU) systems aim to classify an utterance into a predefined set of intents and label the sequence with a predefined ontology of slots (McTear, 2020). Since the release of the ATIS dataset (Price, 1990), this problem has been studied in numerous previous studies (Mesnil et al., 2013; Liu and Lane, 2016; Zhu and Yu, 2017). However, it has recently been shown that the flat structure of sequence labeling falls short when a user issues sub-queries,

[3]maximedb/nllb_massive
[4]maximedb/massive_en_translation
[5]maximedb/massive_generated

or compositional queries, e.g., `set up a reminder to message mike tonight`[6] Gupta et al. (2018) solves that problem by using hierarchical representations instead.

6.2.2 Translation Models

Previous work tackling multilingual intent detection and slot filling uses multilingual versions of well-known Transformers such XLM-Roberta (Conneau et al., 2020), mT5 (Xue et al., 2021), or mBART (Liu et al., 2020). We diverge from existing research and use machine translation models instead. (Fan et al., 2021) released M2M100, a model capable of translating between pairs of 100 languages using a single shared encoder-decoder model. Instead of mainly going from and to English, the authors use a dataset that covers thousands of language pairs. M2M100 was later improved by the release of No Language Left Behind (NLLB) (NLLB Team et al., 2022), which follows the same architecture as M2M100 but covers 202 languages.

6.2.3 Cross-Lingual Task Oriented Semantic Parsing

Although the initial dataset for intent classification and slot filling targeted English, the number of non-English datasets is growing rapidly. Non-English datasets fall into two broad categories: non-English monolingual datasets (Meurs et al., 2008; Castellucci et al., 2019; Bellomaria et al., 2019; Zhang et al., 2017; Gong et al., 2019; He et al., 2013; Dao et al., 2021) and multilingual datasets. As we aim to study models capable of handling multiple languages simultaneously, we focus on the latter kind of datasets. We will now cover the existing multilingual datasets in greater detail. Upadhyay et al. (2018) translated an existing English dataset (Price, 1990) into Turkish and Hindi, while Susanto and Lu (2017) translated the same dataset in Vietnamese and Chinese. Schuster et al. (2019) released a multilingual dataset for task-oriented dialogues in English, Spanish, and Thai across three domains. (Li et al., 2021a) provides MTOP a new aligned task-oriented dataset in six languages. MASSIVE (FitzGerald et al., 2022) is the largest available dataset, covering 51 languages.

6.3 Data

There exist multiple alternative datasets to study multilingual intent detection and slot filling. However, in this work, we use the largest one available: the

[6]Two intents compose that query: create a reminder and send a message to mike.

MASSIVE dataset.

6.3.1 MASSIVE

MASSIVE (FitzGerald et al., 2022) is a dataset assembled by translating and localizing an existing English-only dataset in 50 topologically different languages.

English Seed MASSIVE is a translation of the English-centric SLURP dataset (Bastianelli et al., 2020). SLURP is a dataset of non-compositional queries directed at a home assistant. It covers 18 domains, 60 intents, and 55 slots.

Languages The authors of MASSIVE hired professional translators to translate the SLURP dataset into 50 topologically diverse languages from 29 genera. Furthermore, to complicate the task, the translators sometimes localized the queries instead of simply translating them.

6.3.2 English Data Augmentation

As the seed data of MASSIVE is limited in scale (10K training examples), we used two methods to increase the training set artificially.

6.3.2.1 Generated Data

Generator We first fine-tune a GPT-3 (Brown et al., 2020a) curie (13B) model on the task of generating an English utterance conditional on the given intent. For example, we train the model to generate `wake me up at nine am` given the prompt `alarm_set`.

Parser Next, we fine-tune a second GPT-3 curie model on intent detection and slot filling tasks. Given an utterance, the model must generate the concatenation of the intent and the annotated utterance. For example, given the prompt `what is the temperature in new york?` must generate `weather_query|what is the [weather_descriptor : temperature] in [place_name : new york]`.

Dataset We generate 30,000 utterances, equally distributed amongst the 60 intents. After removing duplicates and examples where the two models do not agree on the intent, we arrive at a final dataset of 22,276 annotated English utterances.

Intent & Slots Distribution Although we generated an equal amount of utterances per intent, removing duplicates skewed the distribution. However, comparing the entropy of both distributions with MASSIVE reveals that our generated dataset is more equally spread amongst the intents but less equally distributed relative to the slots.[7] See Annex C.1 for a detailed analysis and comparison with the MASSIVE dataset.

6.3.2.2 Synthetic Data

The SLURP dataset provides a synthetic dataset.[8] It is not part of the official training set, but as it shares the same ontology as MASSIVE, it provides an excellent extension to our training set. We compare the intent and slot distribution with MASSIVE in Annex C.1.

6.3.3 Non-English Data Augmentation

We explained in Section 6.3.2 our method to artificially increase the size of the (English) training set. This section reviews our method to scale this silver training set to the 50 remaining languages in the MASSIVE dataset.

Using commercial translation systems was not an option as this requires aligning the slots in the translated utterances — a complicated task. Instead, we fine-tune a translation model, NLLB (3B), on the task of translating *annotated* utterances directly. Using this method, we translate annotated utterances and reconstruct the utterances by removing the slot annotations from the text. Our translation model is available on the HuggingFace Hub.[9]

[7] Our generated dataset has an intent distribution entropy of 4.02 and a slot distribution entropy of 3.10 compared to 3.75 and 3.21 for MASSIVE.

[8] https://github.com/pswietojanski/slurp/tree/master/dataset/slurp

[9] https://huggingface.co/maximedb/mmnlu_full_v2

6.4 Model

This work uses a machine translation model for intent detection and slot filling. No Language Left Behind (NLLB) (NLLB Team et al., 2022) is a model specifically targeted at translating between 202 languages using a single encoder-decoder model based on the M2M100 architecture (Fan et al., 2021). It can translate text in 40,602 different directions.

Data NLLB uses FLORES-200 as training data, an extension of FLORES-100 (Goyal et al., 2022). The authors of FLORES-200 used LASER3 (Heffernan et al., 2022) to mine parallel data from the web, resulting in 1.1 billion sentence pairs.

Tokenization NLLB uses a sentencepiece tokenizer (Kudo and Richardson, 2018) with a vocabulary size of 256,000. To ensure low-resource languages are well-represented in the vocabulary, the authors downsample high-resource and upsample low-resource languages.

Architecture NLLB's architecture is based on the Transformer model (Vaswani et al., 2017a). NLLB is trained on several translation directions at once, utilizing the same shared model capacity. This architecture can lead to beneficial cross-lingual transfer between related languages at the risk of increasing interference between unrelated languages. The authors also present a Sparsely Gated Mixture of Experts (MoE) (Almahairi et al., 2016; Bengio et al., 2013). However, we did not experiment with this variant.

Distillation The authors distilled a 54 billion parameter model using MoE into smaller dense models of 1.3 billion and 615 million parameters using online distillation (Hinton et al., 2015). The student model is trained on the training data but with an additional objective: to minimize the cross-entropy to the word-level distribution of the teacher model. We use the distilled 615M parameter model as the base model for intent classification and slot filling.

6.5 Experiments

This section describes our experiments in applying NLLB to the task of intent classification and slot filling. NLLB is a translation model. While we could

Model	Train. Set	Intent Acc (%)			Slot F1 (%)			Exact Match (%)		
		High	Low	Avg	High	Low	Avg	High	Low	Avg
XLM-R	M	88.3	77.2	85.1	83.5	63.3	73.6	70.1	55.8	63.7
mT5 Enc.	M	89.0	79.1	86.1	85.7	64.5	75.4	72.3	57.8	65.9
mT5	M	87.9	79.0	85.3	**86.8**	67.6	76.8	73.4	58.3	66.6
NLLB	M+G	89.3	79.2	87.3	85.9	66.3	77.0	74.1	57.8	68.3
NLLB	M+G+S	**94.5**	**84.5**	**93.4**	82.9	**69.6**	**82.9**	**89.2**	**65.0**	**78.5**

Table 6.1: Modelling results on the MASSIVE test set. NLLB trained on the MASSIVE training set (M), our generated dataset (G) and the synthetic training set from SLURP (S) achieve the highest scores. However, as we show in a later section, this outperformance is due to a large overlap with the MASSIVE test set.

repurpose NLLB to the task of intent classification and slot filling directly, we choose to first pre-train it on a translation task.

6.5.1 Pre-training

As NLLB is, at its core, a translation model, we start by teaching it to translate between the aligned pairs of the MASSIVE dataset. Instead of translating between the utterances of two languages, we translate between the utterance and the annotated utterance. For example, the model must translate "tell me the time in moscow," to the French annotated utterance `datetime_query|donne moi l'heure à [place_name: moscou].` We take special care in avoiding localized utterances, as this would confuse the model. For example, we avoid predicting `datetime_query|donne moi l'heure à ~~moscou~~ bordeaux.`

6.5.2 Fine-tuning

In a second step, we fine-tune the model on the task of translating between the utterance and the annotated utterance in the same language. For example, we translate the utterance `what is the temperature in new york?` into the annotated utterance `weather_query|what is the [weather_descriptor : temperature] in [place_name : new york]`.

6.5.3 Technical Details

We use the NLLB-200 (600M) model for all experiments.[10] We wrap each encoder input according to the following formula: `<s>...</><language_code>`.

[10] facebook/nllb-200-distilled-600M

We prepend each decoder input with the target language code. We train for 50,000 steps during pre-training and fine-tuning with a learning rate of 1e−4 and 1e−5, respectively. We use Pytorch (Paszke et al., 2019), the HuggingFace Trainer (Wolf et al., 2020) and DeepSpeed (Rajbhandari et al., 2020).

6.6 Results

This section presents a high-level analysis of our results. Table 6.1 compares our results against the baselines provided by the authors of MASSIVE.

Our experiments reveal that NLLB performs similarly to mT5 on intent detection and slot filling tasks. Furthermore, our two data augmentation strategies improve the results on the MASSIVE test set. First, training with our generated training set improves the locale average exact match from 66.6 to 68.3. Second, training with the generated and synthetic data boosts the exact match as it improves from 68.3 to 78.5. As we show in the next section, this performance boost is mainly due to a large overlap between the training and test set.

6.7 Training & Test Set Overlap

This section analyses the similarity between the training sets and the MASSIVE. Next, we look for evaluation methods capable of correcting for the overlap between the training and test set.

Exact Duplicates An analysis of the data reveals problematic overlaps between the training sets and the MASSIVE test set. However, this overlap is unequal across the training sets and languages. Table 6.2 shows the percentage of examples in the MASSIVE test set, which are also present in our three training sets. The English subset of the MASSIVE test set overlaps highly with the synthetic training set described in Section 6.3.2.2. Localization and translation somewhat reduce the exact match overlap when looking at all languages, although it remains high. The MASSIVE and generated training sets also have a non-zero overlap with the MASSIVE test set.

Close Duplicates Some examples may not be exact duplicates but close duplicates. For example, `call the dentist` and `olly please call the dentist now`. We use character n-grams to measure the similarity between two utterances

Training Set	en-US (%)	All Locales (%)
MASSIVE	0.7	5.9
Generated	5.6	6.4
Synthetic	49.0	12.8

Table 6.2: Exact duplicate analysis. Percentage of examples in the MASSIVE test set, which are also present in the training set of MASSIVE, our generated training set, and the synthetic training set. Translation reduces the overlap of the synthetic dataset compared to the English-only figures. However, it is the opposite for the MASSIVE test set, where the overlap is higher for all locales compared to English only.

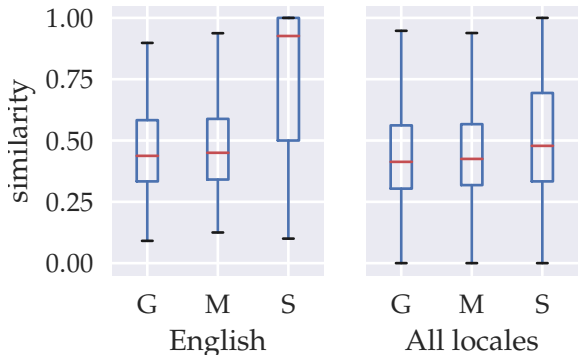

Figure 6.2: Box plot of the maximum similarity between examples in the MASSIVE test set with the training set of MASSIVE (M), Generated (G) and Synthetic (S), for the English part and the entire dataset (all locales). The English synthetic (S) training set overlaps highly with the MASSIVE test set. Translation and localization reduces this overlap in the all-locales dataset.

as similarity metric between two utterances. We search for the most similar training example for each example in the test and record their n-gram similarity.[11] Figure 6.2 shows the distribution of maximum similarity between the test set and our three training sets for the English subset and across all locales. It is clear from Figure 6.2 that the English synthetic dataset overlaps significantly with the English MASSIVE test set. However, as for the exact duplicates, the translation and localization process reduces this overlap but does not eliminate it.

A naive solution would be to remove training examples that overlap with the test set. However, how does one decide what is a close duplicate? Furthermore, as the training set grows, some overlap with the test is inevitable. We argue that the problem is not the training data but the evaluation metric. We need an evaluation metric capable of controlling for the overlap between the test and training sets.

[11]We do this search on a per-language basis.

Training S.	β_0	β_1	R^2
M+G	-0.96±0.03	3.31±0.06	0.07
M+G+S	-0.69±0.03	3.14±0.06	0.08

Table 6.3: We report the logistic regression results for two NLLB models fine-tuned on the training set of MASSIVE (M), generated (G), and synthetic (S). We report the point estimate and the 95% confidence interval for each parameter. After correcting for any overlap between the training and test set, the second is statistically better than the first.

6.7.1 Logistic Regression

Instead of looking at the simple exact match accuracy, we want to express the exact match accuracy as a function of the test/train similarity. One potential solution is to use logistic regression with similarity as the independent variable and exact match as the dependent variable.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{6.1}$$

Where $p(x)$ represents the probability of an exact match, β_0 represents the intercept and β_1 the slope. Using this method, we can compare both models at the same level of similarity.

Results Table 6.3 presents a summary of the logistic regression results. We report the point estimate and confidence interval for both β_0, β_1 and the pseudo R^2 given by `statsmodels` (Seabold and Perktold, 2010). Using Equation 6.1, we can estimate the performance of both models at multiple levels of similarity, as shown in Figure 6.3.

According to Table 6.3 and Figure 6.3, the model trained on the three training datasets is better than the one trained only on two — taking the overlap into account. However, these numbers also indicate that both models struggle with utterances dissimilar to the training set. Moreover, they achieve an exact match accuracy lower than random chance on dissimilar utterances — casting doubt on their abilities to generalize to unseen utterances.

6.7.2 Weighted Average

Another possibility is to give less importance to test examples similar to the training set.

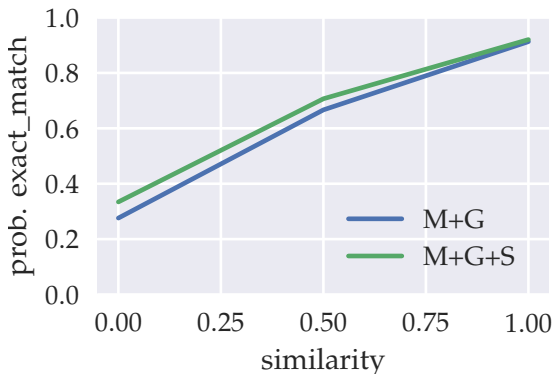

Figure 6.3: Exact match probability at three levels of similarity: 0, 0.5, and 1.0. We used Equation 6.1 with the estimated parameters from Table 6.3. Model two is better than model one on dissimilar utterances. However, the difference diminishes when the similarity increases.

Training Set	**Weighted Average (%)**
M+G	59.2
M+G+S	67.2

Table 6.4: We report the weighted average results for two NLLB models fine-tuned on the training set of MASSIVE (M), generated (G), and synthetic (S). The second model is better than the first even after correcting for its high overlap with the training set.

Language	Intercept	Num. Token Split	R-Squared
ja-JP	0.85*	-0.16*	0.013
zh-CN	0.58*	-0.15*	0.006
zh-TW	0.11	-0.03	0.000

Table 6.5: Logistic regression of exact match accuracy explained by the number of split token. The number of split token negatively influence the capability of the token to correctly parse the slots for ja-JA and zh-CN. The coefficient are not significantly different than zero for zh-TW. Starred numbers (*) are statistically different than zero with a p-value of 0.05

$$\sum_{i=1}^{n} \frac{w_i * exact\_match_i}{\sum_{i=1}^{n} w_i} \tag{6.2}$$

where $w_i = 1 - sim_i$.

Results Table 6.4 displays the results according to the weighted average metric. According to this metric, the second model outperforms the first one. This metric is easy to understand. However, it does not tell us anything about the performance of dissimilar queries.

6.7.3 Summary

According to our overlap-aware evaluation metrics, the model trained on the synthetic datasets is the most performant, even after correcting for its high overlap with the test.

6.8 Error Analysis

6.8.1 Tokenization

Our formatting of input and output consists of surrounding slots with brackets along with the slot name (e.g., `[place_name : new york]`. This method implies that slots' boundaries align with tokenization. Otherwise, the model cannot correctly place the opening or closing bracket — unless it uses a different token than the ones in the source utterance. See Figure 6.8.1 for an example.

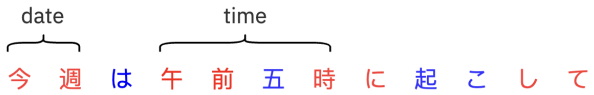

Figure 6.4: Our method does not scale well to non-space delimited languages. For example, in the utterance above, the time slot ends in the middle of a token. To correctly parse the utterance, the model must replace token 20202 (時に) by tokens 249229 (時) and 5954 (に).

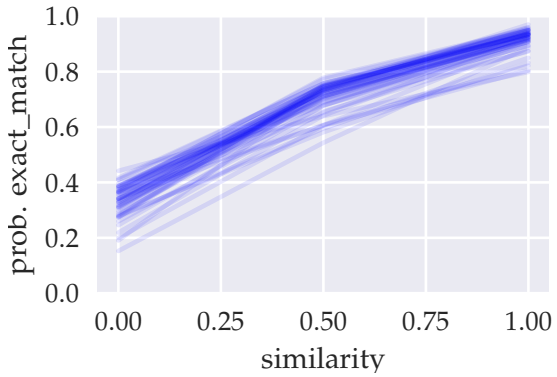

Figure 6.5: Exact match probability at three levels of similarity: 0, 0.5, and 1.0. We used Equation 6.1 with the estimated parameters from Table 6.3. Model two is better than model one on dissimilar utterances. However, the difference diminishes when the similarity increases.

We identified three languages for which this problem occurs: ja-JP in 66% of the test set, zh-CN in 66% of the test set, and zh-TW in 69% of the test set. These are three languages that do not use spaces between words.

Similar to Section 6.7.1, we ran a logistic regression to explain the exact match performance by the number of split tokens. Table 6.5 shows the results. We identified a statistically significant relationship between the number of split tokens and the exact match performance for ja-JP and zh-CN. The performance of zh-TW is low regardless of the number of split tokens.

6.8.2 Generalization

Section 6.7.1 demonstrated that models struggle to generalize to utterances dissimilar to the training set. In this section, we decompose this conclusion by languages. Figure 6.5 decomposes Figure 6.3 by languages. It shows the probability of an exact match on the test set by increasing levels of similarity to the training set. Figure 6.5 shows a wide distribution of probabilities for low similarity utterances (6% standard deviation), while the distribution for highly similar

utterances is more concentrated (3% standard deviation). Some languages do better than others. For example, km-KH achieves an exact match probability of 44% at a similarity of 0.0 while vi-VN only achieves a an exatch match probability of 15%. We list the full details of Figure 6.5 in Appendix C.2.

6.9 Future Work

In this work, we estimated the similarity between two utterances using character n-grams. However, while this captures lexically similar utterances, it fails to capture utterances semantically similar but lexically different. For example, these two utterances are highly similar, although they only share a single common token: `what time is it?` and `tell me the time`. Future work can tackle this by using multilingual sentence encoders such as LASER3 (Heffernan et al., 2022), Multilingual Universal Sentence Encoder (Yang et al., 2020), or multilingual models on Sentence Transformers (Reimers and Gurevych, 2020).

This work did not explicitly address cross-lingual training and instead relied on the cross-lingual pre-training of the translation model. Future work could combine a translation model with cross-lingual training methods such as xTune (Zheng et al., 2021), or X-Mixup (Yang et al., 2022).

Section 6.8.1 showed the limitation of subword tokenization methods. Future work could explore methods which do not uses subword tokenization such as byT5 (Xue et al., 2022).

6.10 Conclusion

In this work, we showed that a translation model such as NLLB can perform the task of intent classification and slot filling. Because of tokenization issues, it is, however, suboptimal with non-spaced languages.

Moreover, we showed that artificially increasing the training sets' size leads to improved performance. Unfortunately, we also show that this added data can overlap with the existing test set, distorting the true evaluation of these models. The normal way to overcome this problem is to remove the overlap from the training set. However, deciding on what constitutes an overlap remains an open question. Therefore, we argued that the data overlap is not the problem — the evaluation metric is. As a result, we proposed two evaluation metrics that control the training/test overlap. Both metrics reveal that the model trained on overlapped data improves the results on non-overlapped data. However, our

analysis also reveals that these models struggle to beat random chance when evaluated on utterances dissimilar to the training set.

Chapter 7

Is It Smaller Than a Tennis Ball? Language Models Play the Game of Twenty Questions

This chapter was published in the Proceedings of the Fifth BlackboxNLP Analyzing and Interpreting Neural Networks for NLP - ISBN 978-1-959429-05-0 - Stroudsburg, Association for Computational Linguistics, 2022, p. 80-90

7.1 Introduction

Generative language models achieve strong performance on multiple NLP tasks by using an unsupervised training objective: predicting the next token in a string of text (Brown et al., 2020a; Chowdhery et al., 2022b; Zhang et al., 2022b).

Despite the simple training objective, these models capture a significant amount of world knowledge (Roberts et al., 2020; Jiang et al., 2020; Talmor et al., 2020). However, we can quickly uncover some limitations by asking simple questions. For example, GPT-3 (Brown et al., 2020a) is more likely to complete the following sentence *question: is a kettle smaller than a tennis ball? answer:* ___ with *yes* than *no*. While trivial for a human, GPT-3 has trouble comparing the size of a kettle and a tennis ball.

We can use the *let's think step by step* method to look into the chain of reasoning of GPT-3 (Kojima et al., 2022): *question: is a kettle smaller than a tennis ball? answer: let's think step by step. [...] a tennis ball is about 6 inches in diameter [...] a typical kettle is about 8-10 inches tall and has a diameter of about 4-5 inches. So, a kettle is smaller than a tennis ball*. According to this example, GPT-3 predicts that a tennis ball is twice its actual size, leading to the wrong conclusion that a kettle is smaller than a tennis ball.

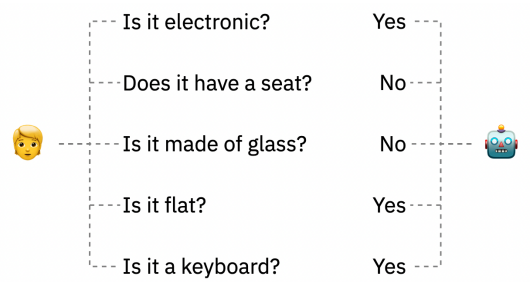

Figure 7.1: Example Twenty Questions game: a human must discover the hidden entity (a keyboard) by asking yes/no questions to the language model. In this case, the model needs to know about the shape, composition, and purpose of a keyboard to correctly answer all questions. While trivial for humans, our results show that this is not the case for most language models, except for GPT-3, which displays fantastic world knowledge on all questions except size-related questions.

In this work, we try to analyze the world knowledge of language models through the game of Twenty Questions. We collected a dataset of 2000+ questions and tried to understand the strength and weaknesses of language models by classifying questions into nine categories of knowledge.

Our results show that GPT-3, a 175 billion parameters language model, can play Twenty Questions thanks to a consistent world knowledge on all categories identified, except for size & shape questions (e.g., *is it bigger than a foot*). Unfortunately, we also show that smaller models do not display the same consistency. However, leveraging the web improved the knowledgeability of T0 by 10% and brought it to a level competitive with GPT-3, despite having 16 times fewer parameters.

Our contributions are the following:

- We release the first dataset consisting of Twenty Questions games.
- We show that very large language models have a consistent world knowledge, while smaller models do not.
- We provide a method to improve the knowledgeability of smaller models using background information from the web.

We publicly release our dataset on HuggingFace (Wolf et al., 2020).[1] We also present *Twentle*, a website to interactively test the world knowledge of language model by playing the game of Twenty Questions.

7.2 Related Work

Although analyzing the capabilities of language models through the game of Twenty Questions is new, researching the amount of general knowledge and common sense of language models is not.

Unfortunately, the knowledge stored by language models is not symbolic. Therefore, we cannot look into the model and inspect its knowledge. Instead, previous work relied on multiple proxy tasks.

One option is to use regular reading comprehension datasets in a closed-book format. Roberts et al. (2020) follow this approach. They evaluate how much knowledge can be stored inside the weights of a text-to-text T5 model (Raffel et al., 2020b). The authors repurposed three reading comprehension datasets to closed-book question answering: Web Questions (Berant et al., 2013), Trivia QA (Joshi et al., 2017) and Natural Questions (Kwiatkowski et al., 2019). They concluded that T5 performs on par with specialized machine comprehension models. GPT-3 (Brown et al., 2020a) was also evaluated on the same closed-book question-answering datasets. The largest model (175B parameters) achieved state-of-the-art results on TriviaQA despite not being trained for the task.

Unfortunately, it has been demonstrated later by Lewis et al. (2021a) that the datasets used by Roberts et al. (2020) and Brown et al. (2020a) suffer from a considerable overlap between the training and test set, invalidating the authors' conclusion based on these datasets. Furthermore, when the overlap between the training and test set is removed, the performance of BART (Lewis et al., 2020a)

[1] https://huggingface.co/datasets/maximedb/twentle

diminishes from 26.7% to 0.8% on TriviaQA (Joshi et al., 2017), suggesting that the model is unable to generalize to previously unseen questions.

To overcome the previously mentioned overlap problem, Wang et al. (2021) repurposed SQuAD (Rajpurkar et al., 2016), a popular reading comprehension dataset, as a closed-book question answering dataset. They evaluated the performance of BART on this new dataset and concluded that it was still challenging for generative models to perform closed-book question answering.

Another approach is to look at how a language model fills in blanks (i.e., masking). One can estimate what the language model knows by carefully analyzing the model's suggestion. This is the approach followed by Petroni et al. (2019). The authors introduce a new dataset LAMA to test the factual and commonsense knowledge in language models. It provides a set of cloze tasks, e.g., *ravens can* _____ with the associated answer *fly*.

The *oLMpic Games* (Talmor et al., 2020) tests the symbolic reasoning of language models through eight synthetic tasks. While very similar to our work, the dataset uses masking to probe the language model. Mask tokens are only applicable to encoder language models, while we are interested in generative language models.

Previous studies have shown that providing generative language models with background information improves their performance. (Borgeaud et al., 2021; Lewis et al., 2020c; Komeili et al., 2022; De Bruyn et al., 2020; Lazaridou et al., 2022) Similar to Lazaridou et al. (2022), we find that including external knowledge improves the language model's performance, however, we obtain better results by restricting the source of knowledge to Wikipedia instead of the entire Internet.

To summarize, we are the first to analyze the world knowledge of generative language models through the game of Twenty Questions. We depart from the work of Roberts et al. (2020) and Wang et al. (2021) in several ways. First, we only have yes/no answers, which simplifies the evaluation and removes the surface-form problem (Holtzman et al., 2021). Second, using generic questions allows disentangling the understanding of the object and the question.

7.3 Data

This section presents our dataset based on the Twenty Questions game — the first boolean closed-book question answering dataset regarding world and commonsense knowledge. We start this section by introducing the Twenty Questions game. We then explain our data collection process. Finally, we analyze the type of knowledge required to perform well on this dataset.

	Twenty Questions
Questions	2,832
Generic questions	915
Entities	126
Words (per question)	6.8
Yes	35%
No	65%

Table 7.1: Summary of the Twenty Questions dataset. We collected 2,832 questions from 126 different entities. We make the distinction between generic and regular questions. Generic questions refer to the entity as "it" (e.g. does it [a rake] have a seat). Generic questions are asked multiple times over different entities (on average 3). We use this unique feature to disentangle the understanding of the question and the entity.

7.3.1 Twenty Questions Game

Wikipedia describes Twenty Questions as a spoken parlor game that encourages deductive reasoning and creativity. In the traditional game, one player (the answerer) chooses a subject and does not reveal it. The other players are questioners and must find the hidden entity by asking yes/no questions.

Previous research focused on playing the questioner (Hu et al., 2018; Chen et al., 2018), however, we are interested in the role of the answerer — the player responsible for answering the yes/no questions using his knowledge of the world. According to our research, this is the first attempt at playing the role of the answerer.

7.3.2 Akinator

Instead of organizing games using Amazon Mechanical Turk, we used Akinator[2] to collect many questions. Akinator is an online game where users can play games of Twenty Questions against a probabilistic model.

Users first pick an entity (without revealing it), and Akinator will then ask yes/no questions to find the hidden entity. It can guess animals, objects, or characters. The player can answer with 5 possible options: *yes*, *no*, *probably yes*, *probably not*, and *don't know*. Although the original Twenty Questions game used a maximum of 20 questions, Akinator will ask questions until it finds the correct entity. We provide examples of questions and entities in Table 7.2. We were pleasantly surprised by the quality of the Akinator model. It was able to find our hidden

[2]https://akinator.com/

Generic Question	Entity	Answer
Is it bigger than a foot?	Padlock	No
Does it work with electricity?	Magnifying glass	No
Does it have a seat?	Forklift	Yes
Does it work with the feet?	Lawn mowner	No
Can it be made of wood?	Rake	Yes
Is it mostly for girls?	Belt	No
Does it have a relationship with school?	Wallet	No
Can it be read?	Worldmap	Yes
Is it made of rubber?	Balloon	Yes
Is it bigger than a foot?	Saw	Yes

Table 7.2: Example questions in our dataset. Akinator does not know the entity when asking the question, and refers to the entity using "it". To avoid any bias toward a specific culture we only used well-known objects as hidden entities. We did not use animals or characters.

entities in most instances. We removed questions from the few instances where it was not capable of finding the correct entity.

Question	Entities
Is it bigger than a foot?	68
Does it go into the mouth?	67
Is it something we wear?	56
Can we buy it?	55
Is it a toy?	50
Is it made of metal?	48
Is it soft?	45
Can it be opened or closed?	42
Is it electronic?	34
Can it be found in a kitchen?	31

Table 7.3: Most common generic questions in the dataset.

7.3.2.1 Generic Questions

Akinator does not know the entity when asking the question and refers to the entity using *"it"*. Because of its probabilistic nature, Akinator will likely ask the same generic question for multiple entities. We list the most common generic questions in Table 7.3. For example *is a rake bigger than a foot* and *is a tennis ball bigger than a foot* are two different questions but share the same generic question *is it bigger than a foot*. The average generic question (e.g., *is it bigger than a foot*) is asked for three different entities. However, the distribution is highly skewed,

with many specific questions asked only once.

7.3.2.2 Choice of Entities

We restricted our choice of entities to objects, as we think characters and animals are too culture-dependent to be deemed general knowledge. As much as possible, we tried to choose objects which are not specific to a particular place or culture.

7.3.2.3 Post-processing

As we are interested in yes/no questions, we remove all questions with *probably yes*, *probably not*, or *don't know* as answer. We use simple regex rules to inject entities into generic questions. We removed all questions about sex or the user's personal experience (e.g., *do you have one at home?*) as these require personal knowledge.

7.3.3 Knowledge Category

In order to understand the reasoning abilities of the language model, we need to understand the type of knowledge required to answer each question correctly.

After carefully reviewing the questions in our dataset, we classified each question into one of the following nine categories: usage, size & shape, location, composition, description, relatedness, appearance, functioning, and purpose. Finally, we provide an overview with examples in Table 7.4.

Shape and Size To answer this kind of question, the model should understand an object's shape and be able to compare it with others. For example, *is it bigger than a foot?*

Usage The model should know how an object is used in everyday life to answer these questions. For example, the model should know that a question like *is it something we wear?* applies to a pair of sunglasses, but not a forklift.

Location The model must know in which place or circumstances an object is used. For example, *can we find it in a bathroom* or, *is it outside*.

Composition These questions require knowing the composition of an object. For example, *is it liquid*, or *is it made of glass*.

Description The model should know how humans describe this object with adjectives. For example, *is it heavy*, or *is it sticky*.

Relatedness To answer these questions, the model must be able to relate two categories of objects or concepts together. For example, *does it have a relation with water*, or *is it a toy*.

Functioning These questions require knowing how an object works. This category is broad and includes questions such as *can it be opened or closed*, or *does it work with electricity*.

Appearance This category is related to the description category but focuses on how an object looks. For example, it includes questions such as *does it have a seat*, or *does it have eyes*.

Purpose This kind of question focuses on the purpose of objects. It is related to the usage category but focuses on why we use objects instead of how. It includes questions like *is it useful to sleep*, or *do we use it for travel*.

Object Knowledge	Example Question	Percentage
Shape and Size	Is it bigger than a foot? Is it flat?	12.7
Usage	Is it something we wear? Do we use it for a sport?	15.5
Location	Can it be found in houses? Is it outside?	10.9
Composition	Is it liquid? Is it made of glass?	7.8
Description	Is it heavy? Is it sticky?	7.1
Relatedness	Does it have a relation with water? Is it a toy?	14.5
Functioning	Does it work with electricity? Can it be opened or closed?	14.8
Appearance	Does it have eyes? Does it have a seat?	6.9
Purpose	Is it useful to sleep? Do we use it for travel?	7.4

Table 7.4: We classified each question of the dataset into nine categories depending on the type of knowledge required to answer the question.

7.3.4 Human Agreement

Answering yes/no question is not always straightforward. A single question can be approached in multiple ways. For example, some people answer the question,

" *is a DVD smaller than a tennis ball* with yes because the height of a DVD is smaller than that of a tennis ball, while others look at the diameter and answer *no*. We asked four annotators to answer 100 randomly sampled questions. On average, they share the same answer as the one in the dataset 94% of the time. The inter-annotator agreement is good, with a Cohen's Kappa score of 0.76 (Cohen, 1968).

7.4 Language Models

In this section, we review the subjects of this work: generative language models. Language models come in all forms and shapes. However, we focus on two types: encoder-decoder and decoder-only models.

7.4.1 Encoder-Decoder Models

Encoder-decoder models treat every NLP task as a text-to-text problem using an encoder-decoder Transformer. When this framework is applied to question answering, the model is trained to generate the literal text of the answer in a free-form fashion (Roberts et al., 2020).

T5 is a text-to-text model pre-trained on multiple tasks simultaneously: translation, summarization, classification, reading comprehension, and an unsupervised span corruption task (Raffel et al., 2020b). We experiment with the 11 billion parameters version.

T0 further trains T5 on 1700 English datasets (Sanh et al., 2022). The resulting model outperforms GPT-3 (Brown et al., 2020a) on several tasks despite being 16x smaller. We use the T0pp version with 11 billion parameters. Conveniently, T0 has already been pre-trained on BoolQ (Clark et al., 2019), a reading comprehension dataset with boolean answers.

7.4.2 Decoder Models

Decoder models use the decoder part of the original Transformer (Vaswani et al., 2017a) model. These models were not trained for a specific task but with an unsupervised objective: predict the next token in a piece of text. Due to their

extensive training corpora, these models have already seen many examples of Trivia style questions.

GPT-3 is an auto-regressive language model (Brown et al., 2020a). The largest version has 175 billion parameters. The model weights are not publicly available, although the model's predictions are available through a paid API.[3]

GPT-J is a 6 billion parameters autoregressive language model (Wang and Komatsuzaki, 2021) trained on the Pile (Gao et al., 2021).

GPT-Neo-X is a 20 billion parameters autoregressive language model (Black et al., 2022) trained on the Pile (Gao et al., 2021).

OPT is a similar model to GPT-3, but the models' weights were publicly released (Zhang et al., 2022b), except for the largest version (175 billion parameters), which is available upon request. Similar to GPT-J, it was trained on the Pile along with data from Reddit. We experiment with the 30 billion parameters version.

7.5 Experiments

In this section, we report on our experiments using our dataset of Twenty Questions. We experimented with three setups: zero-shot, few-shot, and zero-shot with knowledge augmentation. We use these results in the section to understand the scale of the world knowledge stored by language models.

7.5.1 Experimental Settings

Our experiments do not require any training, we use language models as-is without fine-tuning. We use the entirety of our dataset for evaluation. We measure the probability of the *yes* answer by summing the probability of the *yes*, *Yes*, *true*, and *True* tokens. The same is done for the *no* answer with *no*, *No*, *false* and *False*. Our dataset contains 65% of *no* answers, we use F1 (binary) as primary evaluation metric and also report accuracy.

[3]https://openai.com/api/

Model	Size	F1	Accuracy
Majority	-	-	65.0
GPT-J	6B	48.6	49.0
T5	11B	24.6	68.4
T0	11B	**68.5**	**81.9**
GPT-Neo-X	20B	51.8	34.9
OPT	30B	52.8	38.2
GPT-3	13 B	59.4	60.2
GPT-3	175B	66.4	81.3

Table 7.5: Result of the zero-shot evaluation. Best performance is achieved by GPT-3 and T0. The other models struggle to reach the majority vote baseline.

7.5.2 Zero-shot

In the zero-shot setting, models answer the question with only a textual description of the task. We expect T5 and T0 to perform well in this setup as they were pre-trained using the same setup, while this is not the case for decoder-only models.

Prompt We use the same prompt for both encoder-decoders and decoder-only models.

```
You are playing a game of 20 questions.
Answer the following question with yes or no.
Question: {{ question }}
Answer:
```

Results We report the results of our zero-shot experiment in Table 7.5. As expected, T0 achieves the best results with an F1 of 68.5% and an accuracy of 81.9%. GPT-3 also performs nicely in this setup, with 16x more parameters than T0. However, all the other models show an accuracy lower than the majority vote baseline.

7.5.3 Few-shot

In the few-shot setup, models receive identical instructions as in the zero-shot setup, in addition to a few examples. This setup benefits decoder-only models as they can now learn the task on the fly using in-context learning (Beltagy et al., 2022).

Model	Size	F1	Accuracy
Majority	-	-	65.0
GPT-J	6B	57.7	57.7
T5	11B	-	65.8
T0	11B	6.7	65.8
GPT-Neo-X	20B	58.4	58.3
OPT	30B	60.4	71.6
GPT-3	13B	58.2	60.2
GPT-3	175B	**83.0**	**87.9**

Table 7.6: Result of the few-shot evaluation. GPT-3's F1 improves by 9% to reach 83%. The performance of OPT barely improves compared to the zero-shot reasoning, while as expected the performance of encoder-decoder models plummets.

Prompt We augment the zero-shot prompt with four examples. There are two examples with *yes* and two with *no*. We randomly select examples from different entities and generic questions.[4]

```
You are playing a game of 20 questions.
Answer the following question with yes or no.
Question: {{ question_example_1 }}
Answer: {{ answer_example_1 }}
...
Question: {{ question_example_n }}
Answer: {{ answer_example_n }}
Question: {{ question }}
Answer:
```

Results We provide an overview of the few-shots results in Table 7.6. As expected, the performance of decoder-only models increases, while the performance of encoder-decoder decreases[5]. For example, GPT-3's F1 increased from 66.4% to a record 83.0%. Unfortunately, these results also show that (relatively) smaller decoder-only models do not reach T0's performance in a zero-shot setup.

7.5.4 Zero-shot with Knowledge Augmentation

The performance of GPT-3 is exceptional. However, it comes at a steep computational and environmental cost. Moreover, as T0 has fewer parameters than GPT-3,

[4]This setup is similar to the start of a Twenty Questions game where the model does not have previous examples for the same entity.

[5]These models were zero-shot inference, not few-shot.

Model	Size	F1	Accuracy
T0 (ZS)	11B	68.5	81.9
T0 (Bing)	11B	69.7	75.7
T0 (Wiki)	11B	**79.3**	**86.0**
GPT-3 (FS)	175B	**83.0**	**87.9**

Table 7.7: Augmenting T0 with background information improves its F1 score by 10% and brings it to a competitive level with GPT-3.

it has less "space" to store world knowledge. In this section, we try to augment T0 with external knowledge to help it bridge the performance gap with GPT-3. We use two sources of background knowledge: the entire Internet using Bing search and the Wikipedia page of the entity.

Prompt We follow the same prompt as in the zero-shot analysis. In addition, we augment it with a space for background knowledge.

```
Text: {{ background_knowledge }}
You are playing a game of 20 questions.
Answer the following question with yes or no.
Question: {{ question }}
Answer:
```

Bing We run a bing search for every question and only keep the text snippet returned by Bing. We compare each text snippet to the question using a cross-encoder from Sentence Transformers (Reimers and Gurevych, 2019). We then keep the snippet with the highest score. We do not restrict Bing, so it can also choose to return pages from Wikipedia.

Wikipedia We chunk the Wikipedia page of each entity into passages of around 256 tokens. Then, we re-rank the passages using the same cross-encoder.

Results We provide an overview of the few-shots results in Table 7.7. The Bing search results are disappointing. The F1 score barely improves by 1%. On the other hand, the Wikipedia search results are outstanding: F1 improves by over 10% and accuracy by 4%.

This section concludes that GPT-3 (few-shot) is the best model for playing the answerer in a game of Twenty Questions. However, GPT-3 is computationally and environmentally costly. We showed that incorporating background knowledge

from Wikipedia can improve T0's performance to a competitive level with GPT-3 despite having 16 times fewer parameters.

7.6 World Knowledge Analysis

We now use the results of the previous section to analyze the world knowledge of the three best models: GPT-3, T0, and T0 Knowledge Grounded (KG).

7.6.1 Knowledge Category

We list the accuracy by category of knowledge in Table 7.8. The most striking result is the low performance of the three models in the Shape & Size category. For example, GPT-3 has a difference of 20% between the worst category (Shape & Size) and the second-worst category (Usage).

On the other hand, GPT-3 and T0 can answer questions relating to two objects or concepts exceptionally well (e.g., *is it related to water* or *is it a toy*). Intriguingly, incorporating knowledge into the prompt diminishes the score on relatedness for T0-KG.

We now dig deeper into *size & shape* questions and try to understand if there are specific kinds of questions mishandled by the language models. We list the average accuracy by questions in the Shape & Size category in Table 7.9. We notice that questions 1, 3 & 4 are not specific enough. On which dimension should we compare the size of the tennis ball? [6] The inter-annotator score on Shape & Size question is 0.75, almost equivalent to the global inter-annotator score of 0.76. We believe humans have enough common sense to decide on which dimension to evaluate the size of objects.

7.6.2 Entities

Inspired by previous research (Razeghi et al., 2022), we look for a correlation between the average accuracy of an entity and its frequency in the pre-training data.[7] We do not find any significant correlation, except a small 0.05 correlation for T0. We believe the conclusion would be different with lesser-known objects.

[6] Is a DVD smaller than a tennis ball because of its thickness?

[7] We use the first 10 billion tokens of the C4 dataset (Raffel et al., 2020b) to estimate the frequency of entities in the pre-training data.

Knowledge Type	GPT-3	T0	T0-KG	OPT
Shape & Size	66	56	69	60
Usage	86	82	86	75
Location	88	74	89	60
Composition	90	78	78	69
Description	81	69	73	65
Relatedness	95	94	88	79
Functioning	87	79	74	71
Appearance	91	83	83	89
Purpose	91	88	82	75

Table 7.8: Accuracy (%) by category of knowledge. GPT-3 outperforms T0 on every knowledge type. Shape & Size questions stand out as a weak spot for GPT-3 and T0.

Question	GPT-3	T0	T0-KG
Is it smaller than a tennis ball?	50	55	60
Is it globe-shaped?	55	77	77
Is it bigger than a foot?	60	47	67
Can we transport it in a pocket?	62	50	50
Is it flat?	66	55	61
Is it round?	68	43	69
Is it long?	71	28	57
Is it rectangular?	72	81	72
Is it taller than a man?	78	78	71
Does it have a square shape?	80	80	100
Is it pointed?	85	71	71
Is it bigger than a bus?	100	100	100

Table 7.9: Accuracy (%) of GPT-3, T0, and T0-KG on Shape & Size questions. GPT-3 struggles with comparing the size of entities with the size of a tennis ball.

We notice that ambiguous entities such as *a rule*[8] and *a racket*[9] are not well managed by all models for understandable reasons.

7.6.3 Knowledge Augmentation

In this section, we try to understand why Wikipedia is a much better source of background knowledge than Bing's search over the Internet.

Knowledge Source We manually reviewed and compared the background knowledge provided by Bing and Wikipedia. We found that the knowledge returned by Bing can be specific, whereas the game of Twenty Questions requires general knowledge. For example, when asked *does a printer have a seat*, the obvious answer is no. However, Bing returns a text saying *[...] each used printer takes one license seat. [...]* confusing the model into thinking printers do have seats. Another example is the question *is a litter box a weapon*. The correct answer is no. Bing, however, returns a text saying *[...] cat litter box used as a weapon in fight over prescription drugs [...]* confusing the model into thinking a litter box is a weapon. In both instances, the knowledge returned by Wikipedia is the introductory paragraph describing the entity.

Knowledge Category According to Table 7.8, incorporating background knowledge helps in Location (+15%) and Usage (+13%) questions. On the other hand, it hurts performance on Relatedness questions (-6%).

This section concludes that GPT-3 performs consistently on all categories of questions, except Shape and Size. Although competitive, T0 does not show the same consistency as GPT-3, even when augmented with background information.

7.7 Twentle

We present an interactive website to let anyone test the world knowledge of T0-KG by playing the game of Twenty Questions. Inspired by Wordle, we named our website Twentle, available at twentle.com.

[8] As in a 30 cm rule/ruler
[9] As in a tennis racket

7.8 Future Work

Reducing the world to yes/no questions is not an easy task. Our human agreement section demonstrates that humans do not agree on all answers. Future work is needed to compare the agreement of humans and language models by category of question. In this study, we limited ourselves to the study of the answerer. However, GPT-3 could potentially also play the role of the questioner. Future work is needed to study the knowledgeability of language models on lesser-known objects. In this case, we anticipate that large models will also need to leverage the web for information.

7.9 Conclusion

In this work, we analyzed the world knowledge of language models through the game of Twenty Questions. Our analysis reveals that most language models do not have the world knowledge required to play this game. GPT-3 is a notable exception. It displays impressive world knowledge on all categories of questions identified, except for shape & size questions — *is it smaller than a tennis ball*. Furthermore, we showed how grounding smaller models on information from the web improves their knowledgeability. Through this work, we demonstrated the need for more clarity on which model architecture and pre-training method best captures world knowledge.

7.10 Limitations

We intentionally limited our analysis to well-known objects. We anticipate a lower performance on lesser-known objects. Furthermore, our work uses well-defined questions with little noise, whereas real-world questions by humans could be more challenging for language models to understand. The dataset we collected could contain biases already present in our society. Unfortunately, the same is true for the answers given by the language model.

Chapter 8

20Q: Overlap-Free World Knowledge Benchmark for Language Models

This chapter was published in the Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM) - ISBN 978-1-959429-12-8 - 2022, p. 494-508

8.1 Introduction

Transformers are omnipresent in today's Natural Language Processing. Using a simple training and inference procedure, they reach human-level performance on numerous benchmarks.

The scale of these models is hard to grasp. The most recent one, PaLM (Chowdhery et al., 2022c), has 540 billion parameters. It has sixteen times more parameters than all words on Wikipedia, or sixty-eight times more parameters than the total population on Earth (Roser et al., 2013).

Much previous work focused on what these models can do: question-answering, mathematics, translation, or code generation (Wei et al., 2022; Chen et al., 2021b; Cobbe et al., 2021b; NLLB Team et al., 2022; Lewkowycz et al., 2022). Another exciting area of research is to focus on what these models know: common sense, world knowledge, or biases (Kejriwal et al., 2022; Kadavath et al., 2022b; Lucy and Bamman, 2021; Abid et al., 2021).

Transformers (Vaswani et al., 2017a) models do not store knowledge symbolically — they distribute the knowledge within their weights. As a result, researchers have to use proxy tasks to study it. Previous research used closed-book question-answering datasets to study how much knowledge language models can store

Topic	Question	Answer
Gorilla	Is it alive?	Yes
Ball	Can we eat it?	No
Anchor	Is it heavy?	Yes
Pen	Can it fly?	No
Car	Can you drive it?	Yes
Satellite	It it furniture?	No

Table 8.1: Example questions and answers in our 20Q benchmark. We use simple questions to compare the amount of world knowledge between different language models. Despite its apparent simplicity, this benchmark is challenging for even the largest language models — GPT-3 makes a wrong prediction about 20% of the time.

(Roberts et al., 2020). They concluded that language models perform similarly with or without external information, thanks to a broad embedded knowledge.

Unfortunately, Lewis et al. (2021a) later demonstrated that these datasets suffer from a significant overlap between the training and test set. For example, *who has scored more goals in the premier league* shares the same answer with *most goals scored by a premier league player*. Training on the first and evaluating on the second does not make sense. As a result, T5's (Raffel et al., 2020b) performance dramatically dropped when Lewis et al. (2021a) removed the overlap – invalidating the conclusion that these models performed equally with or without external knowledge. Our analysis reveals commonsense reasoning benchmarks also display major overlap between the training and test sets. Commonsense QA 2.0 Talmor et al. (2022) and Com2sense (Singh et al., 2021) have exact or close-to-exact duplicates between the training and test set.

In this work, we propose a new benchmark, free of any lexical and semantic overlap between the training and test set, to evaluate the world knowledge of large language models using the game of Twenty Questions – a popular yes/no guessing game. See Table 8.1 for example questions and answers.

We test two hypotheses using this benchmark. First, we test whether large models possess more world knowledge that smaller models. Second, we test our intuition that world knowledge is correlated with the frequency of the topic in language models' pre-training data.

Despite the massive size of GPT-3, it only reaches an F1 score of 82% on our benchmark. It is however much better than its smaller variants, which validates our first hypothesis that larger models possess more world knowledge than smaller models.

Our dataset's unique feature — a generic question and a topic — is ideal for testing

our second hypothesis: does world knowledge correlate with topic frequency. Again, the results show our hypothesis is true as the bottom quartile of topics is associated with higher variability, whereas the other quartiles are not.

We conclude this introduction by summarizing our main contributions:

- We release a new benchmark to study the world knowledge of language models. It is free of any overlap between the training and test set.
- We show that large models possess more knowledge than smaller ones. However, the relationship is not linear.
- We show that the knowledgeability of language models on a specific topic depends on the relative frequency of the topic in the pre-training data.

We release our benchmark on the HuggingFace dataset hub (Lhoest et al., 2021) for anyone to use.[1]

8.2 Related Work

Before the rise of deep learning, NLP stored commonsense and world knowledge using semantic networks such as WordNet (Miller, 1995b) and later ConceptNet (Speer et al., 2017). These graphs have the advantage of using symbolic representations, facilitating their analysis. Contrary to Transformers-based models, they perform equally well on lower-frequency topics.

Commonsense and world knowledge of Transformers' based models is harder to evaluate, researchers resort to using proxy tasks to evaluate it. Several previous works studied the commonsense abilities of language models in multiple areas: pronoun resolution (Levesque et al., 2012; Sakaguchi et al., 2021), natural language generation (Lin et al., 2020), story understanding (Mostafazadeh et al., 2016), reading comprehension (Zhang et al., 2018; Huang et al., 2019; Ning et al., 2020), physical and social intelligence (Bisk et al., 2020; Sap et al., 2019), temporal reasoning (Zhou et al., 2019), numerical knowledge (Dua et al., 2019; Ravichander et al., 2019), and global commonsense reasoning (Singh et al., 2021; Talmor et al., 2022, 2019).

The remainder of this section focuses on two datasets evaluation the commonsense knowledge of language models using yes/no questions: Commonsense QA 2.0 (Talmor et al., 2022) and Com2Sense (Singh et al., 2021). For both of these datasets, we review the overlap between the training and test set and find troubling examples.

[1] https://huggingface.co/datasets/clips/20Q

8.2.1 Commonsense QA 2.0

Talmor et al. (2022) provide a dataset of 14,343 yes/no questions on several commonsense skills: numerical reasoning, causal reasoning, world knowledge, temporal understanding. The authors used a human-in-the-loop approach to create a challenging benchmark for language models. We partially share the same seed data (AllenAI, 2018) as Commonsense QA 2.0, however we follow a stricter pre-processing and split formation procedure.

Overlap Analysis The authors split the training and test sets according to the topic of questions.[2] Our qualitative review of the overlap between the training and test reveals problematic examples. Some examples are almost duplicates: « *an electron holds a positive charge* » and, « *an electron holds a positive charge and [sic]* », while others are lexically different but semantically similar: « *most happy meals include a toy* » and, « *happy meals almost always come with a toy* ». We provide more examples in Appendix E.1.

8.2.2 Com2sense

Com2sense (Singh et al., 2021) provides a comprehensive commonsense benchmark to test language models' understanding of everyday events and entities by answering yes/no questions. The authors classify their dataset on three axes: knowledge domain (physical, social, or temporal), reasoning scenario (comparative or causal) and numeracy.

Overlap Analysis The authors do not take any special care in the division of the data. However, a key feature of the dataset introduces a high overlap between the two. The authors use a simple technique to double the size of the dataset: edit a few words of each sentence to flip the answer: *to ~~read books~~ see stars at night, one should turn on the lights*. Our qualitative review of the overlap between the training and test reveals highly problematic examples. First, we found exact duplicates between the training and test sets. Second, some examples in the test set are simple negations of examples in the training set. For example « *[...] opening the blinds will help you see* » and, « *[...] opening the blinds will not help you see* ». Third, some examples only change one term between the test and training set, but are semantically similar. We provide more examples in Apppendix E.1.

[2]For example the question « *an uncle has to have a brother or sister* » has the topic *uncle* even though it also is about the *brother* topic.

Dataset	Train	Valid.	Test	No Overlap	Focus	Example
CQA2.0	9,264	2,541	2,473	✗	Multiple	*A bus has at least two steering wheels.*
Com2sense	804	402	2,779	✗	Multiple	*As the weather was very cold he put on his jacket to protect himself.*
20Q (ours)	815	-	2,500	✓	World Knowledge	*Can [an acquittal] cheer you up?*

Table 8.2: Comparison of 20Q with other similar benchmarks. 20Q focuses solely on world-knowledge and is free of any overlap between the training and test set.

8.2.3 Overlap Analysis Summary

Our qualitative review reveals both of these benchmarks do not properly check for training and test set overlap.

Unfortunately, Lewis et al. (2021a) demonstrated that a high overlap between the training and test set can inflate the true performance of language models.

To summarize, we provide the first commonsense reasoning benchmark focused exclusively on world knowledge. Contrary to existing benchmarks, we take extensive measures to ensure there is no overlap between the training and test set. We compare 20Q against alternative benchmarks in Table 8.2.

8.3 Data

Data is a double-edged sword. On the one hand, more data is usually good. However, on the other hand, more data can also complicate the study of the generalization abilities of the model as it gets harder to find uncorrelated validation data.

Regarding world knowledge and common sense, two factors can contaminate the validation data: the training and pre-training data. Large language models can memorize their pre-training data. The bigger the model, the larger the probability of memorization (Chowdhery et al., 2022c).

In this work, we take a novel approach and analyze the inner knowledge of large transformers models through the game of Twenty Questions — a popular yes/no guessing game. We take extra care to avoid lexical and semantic overlap between the training and validation sets.

8.3.1 Twenty Questions Game

Wikipedia describes Twenty Questions as a game that encourages deductive reasoning and creativity. In the traditional game, the answerer chooses a topic and does not reveal it to the questioners, whom themselves must find the hidden entity by asking yes/no questions to the answerer. Humans can play this game (or a variant of it like Guess Who) from a young age.

8.3.2 Twenty Questions Dataset

We do not generate a dataset ourselves. Instead, we rely on an existing dataset of Twenty Questions games developed by AllenAI, where they had humans play the game of Twenty Questions on Amazon Mechanical Turk. In total, they collected 78,890 questions in the style of Twenty Questions. The dataset is available on Github (AllenAI, 2018).[3]

8.3.2.1 Generic Questions

As the questioner does not know the topic, he mainly refers to the entity using "it". Therefore, we term these "generic questions." This disentangling of question and topic is helpful in two regards. First, we can use it to ensure no semantic and lexical overlap between the training and validation sets for both topics and questions. Second, we can measure the topic's knowledge by type of word, domain, or relative frequency in the pre-training data.

8.3.2.2 Fine-grained Answers

Reducing the world to yes and no can be challenging, even impossible. Instead of answering with yes or no, annotators[4] must answer with fine-grained answers: *never*, *rarely*, *sometimes*, *usually*, or *always*. Three annotators answer each question. With a Kappa score of 57%, the disagreement between annotators is high. However, converting the answers to *yes* or *no* instead of fine-grained answers resolves any disagreement between annotators. Using a binary answer also facilitates the analysis.

[3] https://github.com/allenai/twentyquestions

[4] We want to stress that we are referring to the annotation of the original dataset (AllenAI, 2018).

8.3.2.3 Quality Score

Annotators provide a quality score for each question and flag potential problems: questions that are not answerable by yes or no, questions that are not playing the game, or questions that refer to another turn. We only retain questions with the highest quality score (85% of the dataset).

8.3.3 Pre-processing

As with all data generated by humans, it can be noisy. The original dataset contains many sentences with orthographic errors, or even questions unrelated to the Twenty Questions game. Our goal is to understand the knowledge stored inside the language models, not their capacity to deal with noise. Therefore, we take extensive pre-processing steps to clean the dataset. We give further insight into our pre-processing in Annex E.2. First, we remove all questions below the maximum score of three (-15%). Next, we remove all questions which do not use "it" (-12%). Finally, we remove all duplicate questions (-3%) and answers where the topic is not in WordNet (-3%). Our pre-processing removes 34% of the initial dataset.

8.3.4 Training Set

The original authors performed a random split of questions into training, validation, and test set. The authors deal with training/test overlap by flagging questions where the topic is also present in the training set. We take a much stronger stance on train/test overlap and include the semantic overlap between topics and questions.

Our objective is to test the existing knowledge of language models — not to provide new knowledge. Therefore, the priority should be the size of the test set, not the training set. Our training set consists of 815 questions (500 generic questions) on 707 different topics.

8.3.5 Similarity Metrics

Before removing the overlap between the training and test set, we must first decide which similarity metric to use.

We use three methods to compute the similarity between two topics (words) or questions (sequence of words).

Bag-of-words The simplest method to compare two words or sequences of words is their bag-of-words representations. We first tokenize, remove stop-words, and finally stem the words. This method typically identifies close lexical duplicates such as *is it animal* & *is it an animal*.

WordNet Our second method uses the semantic graph WordNet (Miller, 1995b). WordNet excels at identifying synonyms. For example, it will identify that *bike* is a synonym of *bicycle*.

Sentence Transformers Our last method uses Sentence Transformers (Reimers and Gurevych, 2019). It uses pre-trained encoder networks to compute vector representations of sentences (it also works for single words). We can compare the similarity of two sentences (resp. words) by looking at the cosine similarity of their vector representations. We use three different models.

8.3.6 Test Set

We follow three steps before including an example in the test set:

1. We ensure that the bag-of-words representation of the question and the topic is not present in the training set.
2. We check if the topic of the question is not a synonym of any topic in the training set.
3. Our last step removes any example with a cosine similarity larger than 0.8 with any topic or question in the training set.

After all these steps, we arrive at a test set of 4,201 examples. Given the high cost of evaluating very large language models, we only keep the first 2,500 examples. Given the limited size of the validation set, we did not implement a test set. Additional statistics about the dataset are available in Table 8.3. Our validation consists of only 4% of the clean dataset. However, as there is no overlap between the training and validation set, we can make safe conclusions on the generalization abilities of language models.

8.4 Overlap Exploration

Lewis et al. (2021a) demonstrated the devastating effect of an uncontrolled overlap between the training and validation set. Therefore, this section uses different

	Training	**Test**
Questions (total)	815	2,500
Generic Questions	500	1,250
Topics	707	1,436
Words	5.3	5.2
Yes	46%	42%
No	54%	58%

Table 8.3: Descriptive statistics. Our goal is not to learn new knowledge but to test existing knowledge. As a result, the training set is small compared to the validation set.

techniques to inspect the most similar items between the training and validation set.

8.4.1 Topic Overlap in 20Q

We start by analyzing the overlap in topics. For example, we want to avoid having questions about *cars* in the training set and about *automobiles* in the validation set.

N-grams Character n-grams are a good way to retrieve words sharing almost the same lexical form.[5] We show the five most similar pairs of topics between the training and validation set in Table E.2 in Annex E.3. The most similar topics according to this method are *account* and *accountant*. This technique does not reveal problematic overlap between the two sets.

WordNet We use WordNet to compute the distance between two topics by following the hypernym or hyponym chain. Table E.3 in Annex E.3 shows this technique's most similar pair of topics. None of the retrieved pairs show a significant semantical or lexical overlap.

Sentence Transformers We finish our qualitative review of the topic overlap using Sentence Transformers. Table E.4 in Annex E.3 shows the five most similar pairs of topics. The most similar pairs are *costume* with *halloween*, *chlorophyll* and *chrysanthemum*, *bracelet* and *pendant*. All of these words are related, but none are synonyms of one another.

[5]We use a character tri-grams

Train	Topic	Validation	Topic
Does it have a one time function?	knocker	Does it need to be one student at a time?	lettering
Would a parent want their child to do it?	soloist	Is it a category response, like parent or child?	cornea
Can the human population fit on it?	earth	Would it fit in the palm of a human hand?	keyboard
Does it rock?	brim	Is it some sort of precious, rare stone or rock?	emerald
Is it a turn?	heron	Is it something you turn on?	dice

Table 8.4: Qualitative review of the most similar pair of questions computed using BM25. Questions usually share a similar word (e.g., *child* or *rock*), however, it is used in a different context each time. Moreover, the topics are completely unrelated, reducing the risk of overlap even more.

8.4.2 Question Overlap in 20Q

An overlap in terms of topics is only part of the story. We also want to avoid evaluating models on the same kind of answers used to train them. Therefore, we perform the same procedure to avoid lexical and semantic overlap between the questions in the training and validation set. The task is trickier than for topics. For example, *Does it make you cry* and *Does it make you laugh* only differ in a single token, but their meaning is opposite.

BM25 We use BM25 to retrieve similar questions between the two sets. The two most similar questions are *Can the human population fit on it?* and *Would it fit in the palm of a human hand?*. These questions share two important tokens: *fit* and *human*, but they do not have the same meaning. See table 8.4 for more examples. This clearly shows how semantically inequivalent even the most similar sentences in the train and validation set are.

Sentence Transformers Next, we perform the same analysis with Sentence Transformers. The most similar questions between the two sets are *does it have a steering wheel?* and *does it have gears or screws?*, indicating a sufficient amount of dissimilarity between the questions in the training and test set.

8.4.3 Comparison with Existing Benchmarks

We finish this section by comparing the train/test overlap of 20Q with two existing benchmarks presented in Section 8.2: Commonsense QA 2.0 and Com2sense. For each question in the test set, we look for the most similar one in the training set using Sentence Transformers. We summarize the results in Figure 8.1. The results are striking, 20Q has significantly less overlap with the training set than Com2sense and Commonsense QA 2.0. Our qualitative analysis of these results reveal dangerously close duplicates between the training and test of these

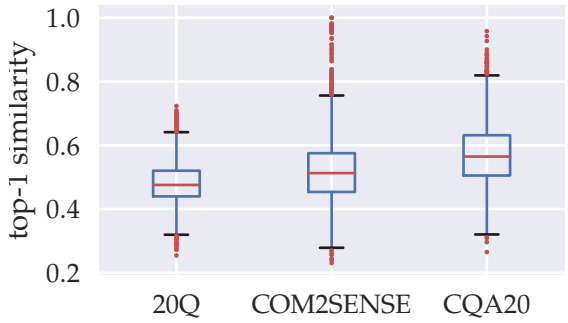

Figure 8.1: Distribution of top-1 similarity between examples in the training and test set. 20Q has the lowest similarity between the two (by design).

		F1			NLL		
Model	**Size**	**Z-S**	**F-S**	**F-T**	**Z-S**	**F-S**	**F-T**
GPT-3	2.7B	58.77	58.02	58.04	112.9	82.64	66.46
GPT-3	6.7B	58.45	54.53	66.35	140.5	80.56	55.41
GPT-3	13B	59.65	48.88	74.48	79.87	65.52	55.63
GPT-3	175B	61.10	67.14	**82.50**	69.86	62.23	**41.16**

Table 8.5: Results per model size and inference method: zero-shot (Z-S), few-shot (F-S), and fine-tune (F-T). According to F1 and NLL, the best method is the largest GPT-3 fine-tuned on our training set.

two benchmarks. Even less expected, we uncover exact duplicates between the training and test of Com2sense. We provide a more detailed analysis in Annex E.1.

To summarize, our benchmark is free of any semantic and lexical overlap between the training and validation set regarding topics and questions. Moreover, despite the strict separation constraints, both sets stay semantically diverse.

8.5 Language Model

After reviewing that data, we review the language models. Although previous work used text-to-text models such as T5 (Raffel et al., 2020b), T0 (Sanh et al., 2022), and BART (Lewis et al., 2020a), in this work, we stick to GPT-3 (Brown et al., 2020a), a general-purpose decoder-only Transformers language model. By sticking to a single model, we can ensure that the only differentiating factor between the models is the network size, not the pre-training data or model architecture.

8.5.1 GPT-3

GPT-3 (Brown et al., 2020a) is an auto-regressive language model developed by OpenAI. The model weights are not publicly available, although the model's predictions are available through a paid API.

Size GPT-3 comes in four sizes: 2.7B, 6.7B, 13B and 175B. We use this feature to understand how the size of a model influences the amount of world knowledge it can store.

Pre-training Data The authors of GPT-3 did not release the pre-training data used to train the model. So instead, we use C4, the dataset used to train T5 (Raffel et al., 2019), as a proxy to estimate the frequency of each topic in our benchmark.

Prompting GPT-3 was never trained to answer yes/no questions. Instead, its objective is to predict the next token in a piece of text. The standard way to query a large language model is to use in-context learning, where one provides a few examples of the task in the prompt and asks the language model to complete the last example.

8.6 Experiments

Our experiments aim at understanding which models possess the best world knowledge. We believe large language models are ineffective at querying their internal knowledge using in-context learning. For this reason, we also fine-tune each model on the training set for a single epoch. The goal is not to teach new knowledge but to guide the model into learning the task. As we meticulously assembled our training and validation splits, we are sure any performance gain will not come from the knowledge acquired during fine-tuning.

8.6.1 Zero-shot

The zero-shot approach is the simplest way to evaluate the knowledge of the language model. The model must predict the next token without any prior examples. We record the probability of the yes token and no token.

Prompt

```
You are playing a game of 20 questions.
Answer the following question
about with yes or no.

Topic: {{ question_topic_1 }}
Question: {{ question_example_1 }}
Answer:
```

8.6.2 Few-shot

This approach improves upon the previous one by providing multiple examples to steer the model in the right direction. The model learns the task *on the fly* using examples from the training set. We record the probability of the yes token and no token.

Prompt

```
Topic: {{ topic_example_1 }}
Question: {{ question_example_1 }}
Answer: {{ answer_example_1 }}
...
Topic: {{ topic_example_n }}
Question: {{ question_example_n }}
Answer:
```

Settings We provide four examples in a random order (two positives and two negatives) from the training set.

8.6.3 Fine-tuning

Understanding the task of answering yes/no questions using on the fly examples is hard. Therefore, we also tested another approach where we fine-tuned models on our training set.

Prompt

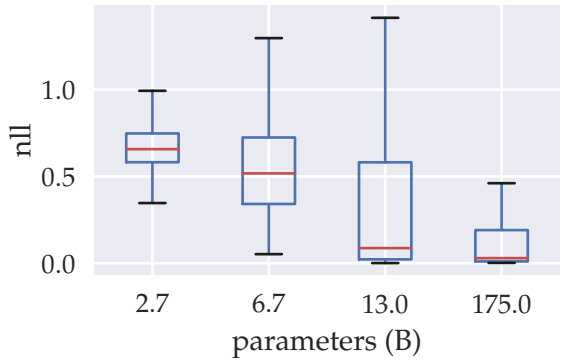

Figure 8.2: Box-plot of negative-likelihood (NLL) per model size.

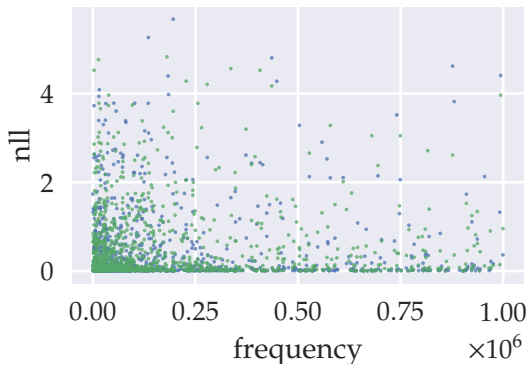

Figure 8.3: Scatter plot of NLL by topic frequency for the 13B (blue) and 175B (green) models.

```
Topic: {{ topic_example }}
Question: {{ question_example }}
Answer:
```

Settings Each model is trained on a single epoch of the training set.

8.7 Results

We run all experiments and report binary-F1 and Negative Log-Likelihood (NLL) to the ground-truth answers in Table 8.5. We start by reviewing the effect of fine-tuning and then analyze our two hypotheses.

8.7.1 Fine-tuning

The benefit of fine-tuning is clear: fine-tuned models are systematically better than few-shot and zero-shot across model size and evaluation metrics. Moreover, thanks to our detailed review of the overlap, we can safely assume the out-performance does not come from learning any new knowledge but is due to better use of the world knowledge already present in the language models.

8.7.2 Size Effect

In theory, the larger the model, the more space it has to store world knowledge. Therefore, we expect to see better performance for large models. Figure 8.2 shows a box-plot of the negative log-likelihood of the fine-tuned results by the model size.

The results are somewhat unexpected. Although the median negative log-likelihood is steadily declining with the model size, the variability also increases with the model size, except for the largest one, which breaks the trend with a low median loss and low variability. In other words, the model's ability to know what it does not know diminishes with model size.

8.7.3 Frequency Effect

Previous research showed that the frequency of tokens in the pre-training data influences the ability of large language models to do numeric reasoning (Razeghi et al., 2022). We hypothesize that the same is true when it comes to world knowledge. Language models should have a harder time answering questions on topics they have rarely encountered during pre-training. Therefore, we collected the frequency count of each topic in a large pre-training corpus: C4 (Raffel et al., 2020b). Our experiments revealed the high correlation of topic frequency with the perplexity of GPT-2 (XL) to generate the word. We use this metric as it scales to different word forms and is easier to collect. [6]

Figure 8.3 clearly shows the frequency effect. Topics associated with a lower frequency quartile have more variability in negative log-likelihood than higher quartiles. This effect is especially strong on the 13B model.

[6]We use the cross-entropy loss (using a sum reduction) from a GPT-2 XL model as a measure of frequency

8.7.4 Question Bias

In this section, we try to uncover whether language models use statistical cues in the question rather than their internal knowledge to answer questions. To this end, we run the fine-tuned model (explained in Section 8.6.3) without the topic in the prompt. If language models use statistical patterns in questions, it should not matter whether the subject is present or not. The F1 score of GPT-3 (175B) drops from 82.50% to 59.40%, just over the performance of the smallest GPT-3 model. We conclude that language models use their internal knowledge rather than statistical cues in the questions.

8.8 Conclusion

Previous research (Lewis et al., 2021a) showed that language models do not have enough world knowledge to rival open-domain question-answering systems. We update this claim using larger models and a novel benchmark, 20Q. We find two factors influencing the world knowledge of language models: the model's size and the topic's frequency in the pre-training data. Thanks to careful attention to the overlap between the training and validation set, we can safely conclude that fine-tuning provides a better picture of the world knowledge possessed by language models. Our benchmark shows that even the largest language models (175 billion parameters) have room for improvement regarding world knowledge. We propose several areas of improvement for coping with a rapidly changing world as future work.

Chapter 9

Conclusion

This thesis investigated several interdisciplinary domains to enhance knowledge-grounded conversations using large language models. The ultimate goal was to improve conversational agents, emphasizing better access to external knowledge and world knowledge, enhancing capabilities in non-English languages, and developing more effective evaluation metrics. This final chapter synthesizes our findings and proposes directions for future research.

9.1 External Knowledge

In Chapter 2, we tackled the challenge of integrating external knowledge into pre-trained language models without incurring prohibitive retraining costs. Our approach merged a non-parametric external database of Wikipedia with a modified BART model, tailored for knowledge-infused dialogues using the Wizard of Wikipedia dataset. This departure from the traditional single-piece knowledge retrieval method of BART encourages the retrieval and fusion of multiple knowledge pieces.

Our system's capability for autonomous sourcing and integration of knowledge enhances the model's efficacy and sets it apart from other Retrieval Augmented Generation (RAG) systems like FiD. The distinguishing feature lies in the incorporation of knowledge pieces within the encoder rather than the decoder. Future work could shed light on the performance differences between these two approaches.

With the increasing interest in large-scale language models such as GPT-3/4, it becomes evident that the incorporation of external knowledge in language models is essential. This necessity is underscored by the high costs of fine-tuning, making the embedding of knowledge within prompts a critical approach to minimizing instances of hallucination.

However, increased computational and latency costs go hand in hand with the integration of more external knowledge pieces into our system. Future research could investigate the use of lightweight adapters, like QLoRA (Dettmers et al., 2023), for incorporating external knowledge. While this is feasible in a supervised fine-tuning setup, how this could be achieved with Reinforcement Learning from Human Feedback (RLHF) trained models remains unclear.

9.2 Open-Domain & Task-Oriented Chatbots

The inherent unpredictability of open-domain dialogues poses significant evaluation challenges. Traditional evaluation methods, such as BLEU or semantic overlap, are often insufficient due to the one-to-many nature of these dialogues.

We proposed a novel reference-free evaluation method, the Follow-Ups Log-Likelihood (FULL), in response to the shortcomings of gold-standard evaluation methodologies. This method measures the likelihood of a large language model responding with a fixed set of negative utterances. FULL provides a more accurate assessment of open-domain dialogues and correlates more closely with human evaluations than twelve other existing methods.

Nevertheless, FULL does not provide a rationale for why one utterance outperforms another – it merely indicates the superior response. There is a striking resemblance to reward models in Reinforcement Learning from Human Feedback (RLHF). Future work could explore using FULL as a reward model to train conversational models with RLHF. This could improve the quality of smaller language models and decrease the costs associated with collecting human feedback.

Recognizing a lack of conversational search datasets for non-English languages, we extended the ConveRT model to Dutch by pre-training it on Dutch conversations from Reddit. ConveRT is an efficient, lightweight retriever model using a dual-tower approach. Thanks to several optimizations, this lightweight ConveRT model can easily run on a CPU. One of its limitations, however, is its inability to mix information bits from multiple answers.

Instead of pre-training on Reddit, we also collected a large number of questions and answers from FAQ pages on the web. This resulted in a unique dataset of its kind. The rationale was to pre-train our model on all of these questions and answers before applying them to a more specific use case, such as answering questions on COVID-19 vaccination. In our experiments, we found that simultaneously training an XLM-RoBERTa across all languages yielded better results compared to individual language models. There were, however, two problems with this approach. First, we only mapped questions to FAQ answers, whereas we could also leverage the similarity of the user query with the FAQ question.

Second, the questions in the dataset were clean and did not reflect the noise of real-life user queries.

We also explored multilingual task-oriented models. Here, we successfully repurposed a translation model for intent detection and slot filling, enabling us to target more languages than traditional encoder-decoder models. However, we identified the need for improved tokenization methods for non-spaced languages like Japanese, where slot boundaries and tokenization do not always align. This misalignment makes the task harder for the model as it has to break tokens apart to be able to insert slot boundaries. One of the main challenges in multilingual task-oriented chatbots is coping with the linguistic nuances of each language. For instance, a Chinese user may not ask for the same artist or restaurant as a Dutch user. Future work could investigate novel ways to automatically localize English datasets to other languages using the latest large language models.

9.3 Common Sense & Background Knowledge

Users of large language models expect a certain level of common sense. A lack of it can quickly lead users to perceive the agent as "dumb." However, measuring common sense is a challenging task. We evaluated the world knowledge of language models using the game of Twenty Questions. We found that language models of all sorts do possess some world knowledge. However, the larger the model, the more world knowledge it has. Our experiments revealed it is possible to enhance their world knowledge by leveraging an external search engine. We also found that language models, especially GPT-3, struggle to compare the size of objects.

In several common sense and world knowledge datasets, there is an overlap between the training set and the test set, which artificially increases the performance of models on these tasks. To overcome this, we created a new benchmark that is free of any lexical and semantic overlap between the training and test sets. Our findings indicated that a model's size and the topic frequency in pre-training data significantly impact its performance on world knowledge tasks.

A potential way to improve the world knowledge of a language model is to incorporate a vision component. We also noticed that language models struggle with infrequent topics. Upsampling these topics in the pre-training data could potentially help improve the world's knowledge of language models.

9.4 Final Thoughts

The challenge of discerning when a language model knows or does not the answer to a query is the missing piece of our research. Interestingly, this issue is not unique to machines, as humans often exhibit overconfidence in their knowledge as well. Identifying the bounds of a model's knowledge has several potential benefits, including:

- Efficient deployment of larger models only when necessary, leading to significant energy savings.
- Enhanced trustworthiness of model outputs by detecting and mitigating hallucinations.
- Improvement in training datasets by pinpointing knowledge gaps.

While in classical machine learning, such as classification, the logit probabilities can act as a confidence score, the scenario is more intricate with generative models. Although generative models effectively employ a form of classification during each token generation, the uncertainty associated with token generation does not necessarily reflect the model's overall uncertainty. This might be because such uncertainties capture both the model's semantic and lexical uncertainty (Lin et al., 2022).

A common approach to gauge a model's semantic certainty is by generating multiple samples and assessing the coherence among them (Manakul et al., 2023; Kadavath et al., 2022a; Agrawal et al., 2023). The thinking is that if the model has semantic uncertainty, not only lexical uncertainty the answers will diverge a lot, however, if the model only has lexical uncertainty they won't diverge semantically. However, this method has limited applicability in practice, mainly because it requires invoking expensive language models multiple times, increasing latency and cost.

The fundamental problem lies with the traditional way of training language models with teacher forcing and cross-entropy. This method is not well suited to teaching models to express uncertainty as it would require knowing in advance what a model knows or does not know. This is not an easy endeavour and remains an open area of research. Recent developments with Reinforcement Learning from Human Feedback (Ouyang et al., 2022) could help in the pursuit of having language models express uncertainty. One could align the model with the simple idea that an uncertain answer is *better* (i.e., has a higher reward) than a wrong or hallucinated answer. This method could potentially teach the language model to express uncertainty without access to its internal knowledge. Since the reward and language model often share the same base model (Touvron et al.,

2023), in theory, the reward could "know" what the language model knows or not (i.e., is it hallucinating or not).

Another potential approach is to use the latest mechanistic analysis of Transformers models and highlight from within when a language model knows or does not know the answer to a particular query.

In conclusion, while large language models like Transformers continue to revolutionize diverse domains, there remains an ever-pressing need for deeper insight into their underlying mechanisms. This insight could be pivotal in addressing the longstanding challenge of discerning the boundaries of a model's knowledge.

Bibliography

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 298–306, New York, NY, USA. Association for Computing Machinery. 95

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020a. Towards a human-like open-domain chatbot. 15, 23

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020b. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977. 30

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Tauman Kalai. 2023. Do language models know when they're hallucinating references? 6, 114

AllenAI. 2018. A web application for playing 20 questions to crowdsource common sense. 98, 100

Amjad Almahairi, Nicolas Ballas, Tim Cooijmans, Yin Zheng, Hugo Larochelle, and Aaron Courville. 2016. Dynamic capacity networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2091–2100. JMLR.org. 67

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7252–7262, Online. Association for Computational Linguistics. 65

Valentina Bellomaria, Giuseppe Castellucci, Andrea Favalli, and Raniero Romagnoli. 2019. Almawave-slu: A new dataset for slu in italian. *arXiv*, abs/1907.07526. 64

Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, and Sameer Singh. 2022. Zero- and few-shot NLP with pretrained language models. In *Proceedings of*

the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pages 32–37, Dublin, Ireland. Association for Computational Linguistics. 87

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432. 67

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics. 79

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199. 45

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439. 97

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics. 86

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *CoRR*, abs/1712.05181. 35

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2021. Improving language models by retrieving from trillions of tokens. *ArXiv*, abs/2112.04426. 80

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones,

Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR. 7

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html. 5

A.Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 21–29. 48

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc. 4, 62, 65, 77, 79, 85, 86, 105, 106

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165. 32

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2021. Convert for FAQ answering. *CoRR*, abs/2108.00719. 46

Marc Brysbaert. 2019. How many words do we read per minute? a review and meta-analysis of reading rate. *Journal of memory and language*, 109:104047. 4

Pawel Budzianowski and Ivan Vulic. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. *CoRR*, abs/1907.05774. 16

Robin D. Burke, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. *AI magazine*, 18(2):57–57. 45, 46

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *arXiv*, abs/1907.02884. 64

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018a. Universal sentence encoder. *CoRR*, abs/1803.11175. 52

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018b. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics. 37, 38

Tyler A. Chang and Benjamin K. Bergen. 2023. Language model behavior: A comprehensive survey. 5

Moses S. Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 380–388, New York, NY, USA. Association for Computing Machinery. 48

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. Evaluating large language models trained on code. 10

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael

Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021b. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374. 95

Yihong Chen, Bei Chen, Xuguang Duan, Jian-Guang Lou, Yue Wang, Wenwu Zhu, and Yong Cao. 2018. Learning-to-ask: Knowledge acquisition via 20 questions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery, Data Mining*, KDD '18, page 1216–1225, New York, NY, USA. Association for Computing Machinery. 81

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics. 4

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022a. Palm: Scaling language modeling with pathways. 10

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar

Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022b. Palm: Scaling language modeling with pathways. 77

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek B Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022c. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311. 95, 99

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics. 85

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*. 10

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168. 95

J. Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin*, 70 4:213–20. 85

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116. 53

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics. 11, 61, 64

Justine Coupland. 2003. Small talk: Social functions. *Research on Language and Social Interaction*, 36:1 – 6. 8

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In *Fourth Workshop on Very Large Corpora*, Herstmonceux Castle, Sussex, UK. Association for Computational Linguistics. 3

Sonam Damani, Kedhar Nath Narahari, Ankush Chatterjee, Manish Gupta, and Puneet Agrawal. 2020. Optimized transformer models for faq answering. In *Advances in Knowledge Discovery and Data Mining*, pages 235–248, Cham. Springer International Publishing. 37

Mai Hoang Dao, Thinh Hung Truong, and Dat Quoc Nguyen. 2021. Intent detection and slot filling for vietnamese. *arXiv*, abs/2104.02021. 64

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2020. Bart for knowledge grounded conversations. In *Converse@ KDD*. 35, 44, 80

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2021. MFAQ: a multilingual FAQ dataset. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 1–13, Punta Cana, Dominican Republic. Association for Computational Linguistics. 61

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. arXiv:1912.09582. 11, 39

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. 112

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805. 46, 53

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 4

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*. 15, 16, 18, 19, 20, 21, 23, 24

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics. 97

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211. 3

Maxine Eskénazi, Shikib Mehri, Evgeniia Razumovskaia, and Tiancheng Zhao. 2019. Beyond turing: Intelligent agents centered on the user. *CoRR*, abs/1901.06613. 27

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1). 64, 67

Angela Fan, Claire Gardent, Chloe Braud, and Antoine Bordes. 2020. Augmenting transformers with knn-based composite memory for dialogue. 15, 16, 18, 23

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE. 46

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123. Association for Computational Linguistics. 46, 47

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv*, abs/2204.08582. 61, 62, 64, 65

Rodney A Gabriel, Tsung-Ting Kuo, Julian McAuley, and Chun-Nan Hsu. 2018. Identifying and characterizing highly similar notes in big clinical note datasets. *Journal of biomedical informatics*, 82:63–69. 48

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. 86

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 5

Sarik Ghazarian, Behnam Hedayatnia, Alexandros Papangelis, Yang Liu, and Dilek Hakkani-Tur. 2022a. What is wrong with you?: Leveraging user sentiment for automatic dialog evaluation. *arXiv*, abs/2203.13927. 27

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7789–7796. 27

Sarik Ghazarian, Nuan Wen, Aram Galstyan, and Nanyun Peng. 2022b. Deam: Dialogue coherence evaluation using amr-based semantic manipulations. *arXiv*, abs/2203.09711. 27

Yu Gong, Xusheng Luo, Yu Zhu, Wenwu Ou, Zhao Li, Muhua Zhu, Kenny Q. Zhu, Lu Duan, and Xi Chen. 2019. Deep cascade multi-task learning for slot filling in online shopping assistant. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. 64

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538. 67

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: An automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA. Association for Computing Machinery. 3

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions. 10

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics. 64

Bikash Gyawali, Lucas Anastasiou, and Petr Knoth. 2020. Deduplication of scholarly documents using locality sensitive hashing and word embeddings. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 901–910. European Language Resources Association. 48

Kristian Hammond, Robin Burke, Charles Martin, and Steven Lytinen. 1995a. Faq finder: a case-based approach to knowledge navigation. In *Proceedings the 11th Conference on Artificial Intelligence for Applications*, pages 80–86. IEEE. 35, 36

Kristian Hammond, Robin Burke, Charles Martin, and Steven Lytinen. 1995b. FAQ finder: a case-based approach to knowledge navigation. In *Proceedings the 11th Conference on Artificial Intelligence for Applications*, pages 80–86. IEEE. 43, 45, 46

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. 11

Xiaodong He, Li Deng, Dilek Hakkani-Tur, and Gokhan Tur. 2013. Multi-style adaptive training for robust cross-lingual spoken language understanding. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8342–8346. 64

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *arXiv*, arxiv.2205.12654. 67, 75

Matthew Henderson, Iñigo Casanueva, Nikola Mrksic, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulic. 2019a. Convert: Efficient and accurate conversational representations from transformers. *CoRR*, abs/1911.03688. 35, 37, 38, 46

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. Convert: Efficient and accurate conversational representations from transformers. 7

Matthew Henderson, Ivan Vulic, Daniela Gerz, Iñigo Casanueva, Pawel Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksic, and Pei-Hao Su. 2019b. Training neural response selection for task-oriented dialogue systems. *CoRR*, abs/1906.01543. 37

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*. 10

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv*, abs/1503.02531. 67

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. 3

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 80

Huang Hu, Xianchao Wu, Bingfeng Luo, Chongyang Tao, Can Xu, Wei Wu, and Zhan Chen. 2018. Playing 20 question game with policy-based reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3233–3242, Brussels, Belgium. Association for Computational Linguistics. 81

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics. 97

Lishan Huang, Zheng Ye, Jinghui Qin, Liang Lin, and Xiaodan Liang. 2020. GRADE: Automatic graph-enhanced coherence metric for evaluating open-domain dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9230–9240, Online. Association for Computational Linguistics. 26

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Real-time inference in multi-sentence tasks with deep pretrained transformers. *CoRR*, abs/1905.01969. 16

Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Najat Drawel, Gaith Rjoub, and Witold Pedrycz. 2023. A comprehensive survey on applications of transformers for deep learning tasks. 4

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. 7

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. 7

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438. 77

Valentin Jijkoun and Maarten de Rijke. 2005a. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83. 35

Valentin Jijkoun and Maarten de Rijke. 2005b. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83. 43, 45

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics. 79, 80

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759. 47

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022a. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221. 114

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022b. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221. 95

Mladen Karan and Jan Šnajder. 2016a. Faqir – a frequently asked questions retrieval test collection. In *Text, Speech, and Dialogue*, pages 74–81. Springer International Publishing. 43, 46

Mladen Karan and Jan Šnajder. 2016b. Faqir–a frequently asked questions retrieval test collection. In *International Conference on Text, Speech, and Dialogue*, pages 74–81. Springer. 35

Mladen Karan and Jan Šnajder. 2018. Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval. *Expert Systems with Applications*, 91:418–433. 46

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics. 44, 52, 53

Mayank Kejriwal, Henrique Santos, Alice M Mulvehill, and Deborah L McGuinness. 2022. Designing a strong test for measuring true common-sense reasoning. *Nature Machine Intelligence*, 4(4):318–322. 95

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization - nearest neighbor language models. DBLP's bibliographic metadata records provided through http://dblp.org/search/publ/api are distributed under a Creative Commons CC0 1.0 Universal Public Domain Dedication. Although the bibliographic metadata records are provided consistent with CC0 1.0 Dedication, the content described by the metadata records is not. Content may be subject to copyright, rights of privacy, rights of publicity and other restrictions.; 8th International Conference on Learning Representations, ICLR 2020 ; Conference date: 26-04-2020 Through 01-05-2020. 7

Harksoo Kim and Jungyun Seo. 2006. High-performance FAQ retrieval using an automatic clustering method of query logs. *Information Processing & Management*, 42(3):650–661. 45

Harksoo Kim and Jungyun Seo. 2008a. Cluster-based faq retrieval using latent term weights. *IEEE Intelligent Systems*, 23(02):58–65. 36

Harksoo Kim and Jungyun Seo. 2008b. Cluster-based FAQ retrieval using latent term weights. *IEEE Intelligent Systems*, 23(2):58–65. Conference Name: IEEE Intelligent Systems. 45

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 39

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. 77

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021a. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*. 35

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021b. Internet-augmented dialogue generation. *CoRR*, abs/2107.07566. 44

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics. 80

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. 67

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466. 79

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc. 4

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. 80

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *CoRR*, abs/2107.06499. 48

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets*, 2nd edition. Cambridge University Press, USA. 48

Guy Lev, Michal Shmueli-Scheuer, Achiya Jerbi, and David Konopnicki. 2020. orgfaq: A new dataset and analysis on organizational faqs and user questions. *CoRR*, abs/2009.01460. 47

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press. 97

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 16, 18, 19, 24

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. 79, 105

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. 15

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc. 7, 80

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021a. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics. 10, 79, 96, 99, 102, 110

Patrick S. H. Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021b. PAQ: 65 million probably-asked questions and what you can do with them. *CoRR*, abs/2102.07033. 44

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. 95

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 97

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021a. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics. 62, 64

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021b. Conversations are not flat: Modeling the dynamic information flow across dialogue utterances. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 128–138, Online. Association for Computational Linguistics. 27

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics. 97

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*. 114

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689. 63

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. 16

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv*, abs/2001.08210. 64

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692. 53

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv*, abs/1907.11692. 62

Ehsan Lotfi, Maxime De Bruyn, Jeska Buhmann, and Walter Daelemans. 2021. Teach me what to say and i will learn what to pick: Unsupervised knowledge selection through response generation with pretrained generative models. 35

Li Lucy and David Bamman. 2021. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, Virtual. Association for Computational Linguistics. 5, 95

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. 114

Christopher D. Manning. 2022. Human Language Understanding amp; Reasoning. *Daedalus*, 151(2):127–138. 3

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics. 11

Yosi Mass, Boaz Carmeli, Haggai Roitman, and David Konopnicki. 2020. Unsupervised FAQ retrieval with question generation and BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 807–812, Online. Association for Computational Linguistics. 46

Michael McTear. 2020. Conversational ai: dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies*, 13(3):1–251. 25, 63

Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, Samira Shaikh, David Traum, Yi-Ting Yeh, Zhou Yu, Yizhe Zhang, and Chen Zhang. 2022. Report from the nsf future directions workshop on automatic evaluation of dialog: Research directions and challenges. *arXiv*, abs/2203.10012. 25

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 1st virtual meeting. Association for Computational Linguistics. 26, 27, 28, 30, 31, 148

Shikib Mehri and Maxine Eskenazi. 2020b. Usr: An unsupervised and reference free evaluation metric for dialog generation. 24

Shikib Mehri and Maxine Eskenazi. 2020c. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics. 27

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*. 63

Marie-Jean Meurs, Frédéric Duvert, Frédéric Béchet, Fabrice Lefèvre, and Renato de Mori. 2008. Semantic frame annotation on the French MEDIA corpus. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). 64

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. 4

George A Miller. 1995a. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41. 45

George A. Miller. 1995b. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41. 97, 102

A. Moreo, E. M. Eisman, J. L. Castro, and J. M. Zurita. 2013. Learning regular expressions to template-based FAQ retrieval systems. *Knowledge-Based Systems*, 53:108–128. 45

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007a. Exploiting syntactic and shallow semantic kernels for question answer

classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 776–783. 36

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007b. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783. Association for Computational Linguistics. 46

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics. 97

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 269–281, Denver, Colorado. Association for Computational Linguistics. 46, 47

Massimo Nicosia, Zhongdi Qu, and Yasemin Altun. 2021. Translate & Fill: Improving zero-shot multilingual semantic parsing with synthetic data. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3272–3284, Punta Cana, Dominican Republic. Association for Computational Linguistics. 62

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online. Association for Computational Linguistics. 97

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672. 62, 64, 67, 95

Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. 2013. *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*, pages 219–247. Springer Berlin Heidelberg, Berlin, Heidelberg. 40

OpenAI. 2023. Gpt-4 technical report. 4

Roeland J.F. Ordelman, Franciska M.G. de Jong, Adrianus J. van Hessen, and G.H.W. Hondorp. 2007. Twnc: a multifaceted dutch news corpus. *ELRA Newsletter*, 12(3-4). 40

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics. 47

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. 114

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval. 53

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3619–3629, Online. Association for Computational Linguistics. 27

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. 25

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 69

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. 6

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics. 80

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. 11

Vitou Phy, Yang Zhao, and Akiko Aizawa. 2020. Deconstruct to reconstruct a configurable evaluation metric for open-domain dialogue systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4164–4178, Barcelona, Spain (Online). International Committee on Computational Linguistics. 27

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*. 63, 64

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. In *OpenAI Blog*. 4

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. 16, 27, 29

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*. 106

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. 62

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. 79, 85, 90, 96, 105, 109

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press. 69

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics. 80

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics. 97

Yasaman Razeghi, Robert L. Logan, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. 90, 109

Evgeniia Razumovskaia, Goran Glavas, Olga Majewska, Edoardo M Ponti, Anna Korhonen, and Ivan Vulic. 2022. Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems. *Journal of Artificial Intelligence Research*, 74:1351–1402. 63

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. 89, 102, 159

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics. 75

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O Mittal, and Yi Liu. 2007a. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471. 35

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. 2007b. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471. 43, 45

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics. 77, 79, 80, 85, 96

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021a. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*. 35

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021b. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *CoRR*, abs/2107.12708. 43

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. 2020a. Recipes for building an open-domain chatbot. 16, 24

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2020b. Recipes for building an open-domain chatbot. *CoRR*, abs/2004.13637. 29, 31

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics. 15

Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. 2013. World population growth. *Our World in Data*. 95

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106. 97

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019a. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116. 35, 36

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019b. FAQ retrieval using query-question similarity and bert-based query-answer relevance. *CoRR*, abs/1905.02851. 43, 46

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620. 52

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al.

2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations*. 85, 105

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics. 97

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics. 64

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*. 71

Abigail See and Christopher Manning. 2021. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12, Singapore and Online. Association for Computational Linguistics. 27

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics. 4

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021a. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics. 6

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021b. Retrieval augmentation reduces hallucination in conversation. *CoRR*, abs/2104.07567. 8, 44

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021c. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*. 35

Shikhar Singh, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. COM2SENSE: A commonsense reasoning benchmark with complementary sentences. In *Findings of the Association for Compu-*

tational Linguistics: ACL-IJCNLP 2021, pages 883–898, Online. Association for Computational Linguistics. 96, 97, 98

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L. Hamilton, and Joelle Pineau. 2020. Learning an unreferenced metric for online dialogue evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2430–2441, Online. Association for Computational Linguistics. 27

Eriks Sneiders. 1999a. Automated faq answering: Continued experience with shallow language understanding. In *Question Answering Systems. Papers from the 1999 AAAI Fall Symposium*, pages 97–107. 35, 36

Eriks Sneiders. 1999b. Automated faq answering: Continued experience with shallow language understanding. In *Question Answering Systems. Papers from the 1999 AAAI Fall Symposium*, pages 97–107. 43, 45

Eriks Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database. In *International Conference on Application of Natural Language to Information Systems*, pages 235–239. Springer. 45

Eriks Sneiders. 2009. Automated FAQ answering with question-specific knowledge representation for web self-service. In *2009 2nd Conference on Human System Interactions*, pages 298–305. IEEE. 45

Eriks Sneiders. 2010. Automated email answering by text pattern matching. In *International Conference on Natural Language Processing*, pages 381–392. Springer. 45

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press. 97

Raymond Hendy Susanto and Wei Lu. 2017. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44, Vancouver, Canada. Association for Computational Linguistics. 64

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014a. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press. 4

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014b. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215. 16

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758. 77, 80

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics. 97

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2022. Commonsenseqa 2.0: Exposing the limits of AI through gamification. *ArXiv*, abs/2201.05320. 96, 97, 98

Noriko Tomuro and Steven L Lytinen. 2004a. Retrieval models and q and a learning with faq files. In *New Directions in Question Answering*, pages 183–202. 36

Noriko Tomuro and Steven L. Lytinen. 2004b. Retrieval models and q and a learning with FAQ files. In *New Directions in Question Answering*, pages 183–202. 45

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. 4, 5, 10, 114

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. 64

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is

all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 67, 85, 95

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. *CoRR*, abs/1706.03762. 3, 18, 37, 38

Yannick Versley and Yana Panchenko. 2012. Not just bigger: Towards better-quality web corpora. In *Proceedings of the seventh Web as Corpus Workshop (WAC7)*, pages 44–52. 48

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. 5

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`. 86

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021. Can generative pre-trained language models serve as knowledge bases for closed-book QA? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3241–3251, Online. Association for Computational Linguistics. 80

Xinyi Wang, Jason Weston, Michael Auli, and Yacine Jernite. 2019. Improving conditioning in context-aware sequence to sequence models. 16

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903. 95

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics. 16

Steven D. Whitehead. 1995. Auto-FAQ: an experiment in cyberspace leveraging. *Selected Papers from the Second World-Wide Web Conference*, 28(1):137–146. 45

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771. 19

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz,

Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. 69, 79

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149. 16, 18

W.A. Woods. 1978. Semantics and quantification in natural language question answering. volume 17 of *Advances in Computers*, pages 1–87. Elsevier. 3

Ruobing Xie, Yanan Lu, Fen Lin, and Leyu Lin. 2020. FAQ-based question answering via knowledge anchors. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 3–15. Springer. 45

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808. 53

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306. 75

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics. 61, 64

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2022. Enhancing cross-lingual transfer by manifold mixup. *arXiv*, abs/2205.04182. 75

Sheng-Yuan Yang. 2009. Developing of an ontological interface agent with template-based linguistic processing technique for FAQ services. *Expert Systems with Applications*, 36(2):4049–4060. Publisher: Elsevier. 45

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics. 75

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, Online. Association for Computational Linguistics. 26, 31, 32

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. 2021. DynaEval: Unifying turn and dialogue level evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5676–5689, Online. Association for Computational Linguistics. 26

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball. 6

Qingyu Zhang, Xiaoyu Shen, Ernie Chang, Jidong Ge, and Pengke Chen. 2022a. Mdia: A benchmark for multilingual dialogue generation in 46 languages. *arXiv*, abs/2208.13078. 61

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *ArXiv*, abs/1810.12885. 97

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022b. Opt: Open pre-trained transformer language models. 32, 77, 86

Weinan Zhang, Zhigang Chen, Wanxiang Che, Guoping Hu, and Ting Liu. 2017. The first evaluation of chinese human-computer dialogue technology. *CoRR*, abs/1709.10217. 64

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. 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*, pages 270–278, Online. Association for Computational Linguistics. 29

Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Consistency regularization for cross-lingual fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3403–3417, Online. Association for Computational Linguistics. 62, 75

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics. 97

Su Zhu and Kai Yu. 2017. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5675–5679. IEEE. 63

Appendix A

Appendix Chapter 3

A.1 Appendix: Comparison of Models

We present in Figure A.1 the average absolute correlation to human evaluations per model.

A.2 Appendix: List of Candidate Follow-ups

Table A.1 list the entire list of follow-ups considered.

	Follow-up	Category	Level	Type	Level	Dialog
X	Not really relevant here.	specific	turn	neg	0.48	0.65
X	You're really confusing.	error recovery	dialog	neg	0.46	0.67
	I don't understand what you're saying.	correct	turn	neg	0.46	0.58
	That's not really relevant here.	specific	turn	neg	0.45	0.70
	You are so confusing.	coherent	dialog	neg	0.45	0.64
X	You're really boring.	informative	dialog	neg	0.44	0.65
	That's not very interesting.	interesting	turn	neg	0.44	0.60
	That was a really boring response.	interesting	turn	neg	0.43	0.63
X	You don't seem interested.	inquisitive	dialog	neg	0.43	0.61
	I am so confused right now.	error recovery	dialog	neg	0.43	0.60
	I'm so confused!	understandable	turn	neg	0.43	0.59
	I don't really care. That's pretty boring.	engaging	turn	neg	0.43	0.61
	I want to talk about something else.	engaging	turn	neg	0.43	0.65
	That's not even related to what I said.	relevant	turn	neg	0.42	0.58
X	What are you trying to say?	understanding	dialog	neg	0.42	0.68
	I am so confused right now!	correct	turn	neg	0.42	0.57
	That makes no sense!	appropriate	turn	neg	0.42	0.56
	I don't understand at all!	understandable	turn	neg	0.41	0.54
	That's really boring.	interesting	turn	neg	0.41	0.54
	I don't like you.	likeable	dialog	neg	0.40	0.58
	I'm so confused right now!	fluent	turn	neg	0.40	0.56
	Don't change the topic!	relevant	turn	neg	0.40	0.58
	You're not understanding me!	correct	turn	neg	0.40	0.62
	That's a very generic response.	specific	turn	neg	0.39	0.50
	You don't really know much.	informative	dialog	neg	0.39	0.52
	You're not very nice.	likeable	dialog	neg	0.38	0.56
	You're not very fun to talk to.	likeable	dialog	neg	0.37	0.55
	Is that real English?	fluent	turn	neg	0.37	0.49
	That's a lot of questions!	inquisitive	dialog	pos	0.36	0.52
	Why are you repeating yourself?	diverse	dialog	neg	0.35	0.50
	You're making no sense at all.	coherent	dialog	neg	0.35	0.43
	You ask a lot of questions!	inquisitive	dialog	pos	0.35	0.54
	Let's change the topic.	engaging	turn	neg	0.35	0.45

Table A.1 continued from previous page

Follow-up	Category	Level	Type	Level	Dialog
You don't ask many questions.	inquisitive	dialog	neg	0.35	0.54
Why are you changing the topic?	relevant	turn	neg	0.34	0.51
Stop saying the same thing repeatedly.	diverse	dialog	neg	0.34	0.50
Do you know how to talk about something else?	flexible	dialog	neg	0.33	0.49
You're changing the topic so much!	coherent	dialog	neg	0.33	0.47
You know a lot of facts!	informative	dialog	pos	0.32	0.48
Tell me more!	engaging	turn	pos	0.32	0.34
I like you!	likeable	dialog	pos	0.31	0.43
Wow that's a lot of information.	informative	dialog	pos	0.31	0.38
Stop changing the topic so much.	depth	dialog	neg	0.31	0.44
What does that even mean?	understandable	turn	neg	0.30	0.35
I don't want to talk about that!	flexible	dialog	neg	0.29	0.50
That's not what you said earlier!	consistent	dialog	neg	0.29	0.37
You have a good point.	appropriate	turn	pos	0.29	0.43
I see, that's interesting.	specific	turn	pos	0.28	0.31
Stop contradicting yourself!	consistent	dialog	neg	0.28	0.36
You're very easy to talk to!	flexible	dialog	pos	0.28	0.40
Stop repeating yourself!	diverse	dialog	neg	0.27	0.40
That's good to know. Cool!	specific	turn	pos	0.25	0.30
That's a good point.	specific	turn	pos	0.25	0.34
Wow you can talk about a lot of things!	flexible	dialog	pos	0.23	0.27
I'm really interested in learning more about this.	engaging	turn	pos	0.22	0.26
That makes sense!	appropriate	turn	pos	0.21	0.21
Thanks for all the information!	informative	dialog	pos	0.21	0.15
You're super polite and fun to talk to	likeable	dialog	pos	0.17	0.23
Wow that is really interesting.	interesting	turn	pos	0.17	0.14
That's really interesting!	interesting	turn	pos	0.16	0.11
Great talking to you.	likeable	dialog	pos	0.15	0.10
Cool! That sounds super interesting.	interesting	turn	pos	0.08	- 0.01
Wow! That's really cool!	engaging	turn	pos	0.04	- 0.08

Table A.1: List of candidate follow-ups along with their category (fine-grained attribute), positivity (negative of positive follow-up) and correlation with a human evaluation of the overall quality of the turn/dialog. All follow-ups and static data is from Mehri and Eskenazi (2020a).

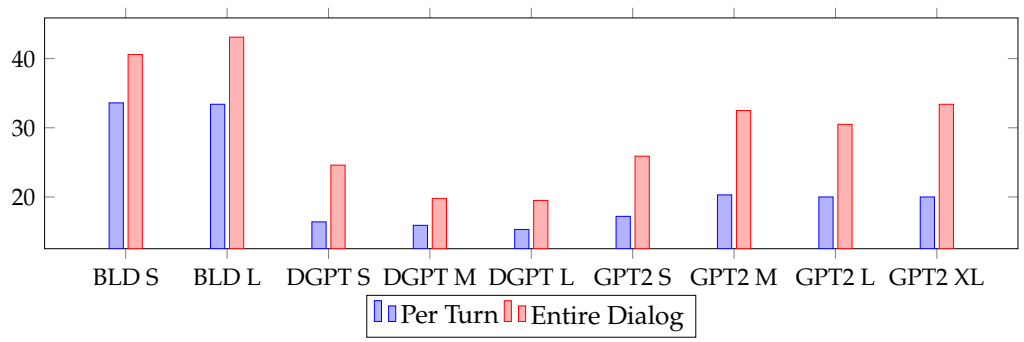

Figure A.1: Average absolute correlation with human evaluations for several language models. We use Blender-400 (BLD S) as language model because of its high correlation with human evaluations. For space reasons, Blender is abbreviated as BLD and DialoGPT as DGPT.

Appendix B

Appendix Chapter 5

Language	Random			TF-IDF			USE			XLM-Roberta (1 page per domain)			XLM-RoBERTa (full training set)			Monolingual		
	P@1	MRR	R@5	P@1	MRR	R@5	P@1	MRR	R@5	P@1	MRR	R@5	P@1	MRR	R@5	P@1	MRR	R@5
English	5.9	18.9	29.7	53.9	63.8	79.8	52.6	64.2	83.0	74.9	82.5	93.5	72.5	80.7	92.5	75.6	**82.9**	93.5
German	5.8	18.3	28.8	48.0	58.0	74.8	49.8	61.8	81.6	73.0	**81.3**	93.7	69.0	78.2	92.0	72.5	81.1	93.6
Spanish	7.3	22.3	36.7	49.2	60.5	78.9	49.3	61.6	82.0	72.8	**81.7**	94.5	68.9	78.6	93.2	68.6	78.0	92.3
French	6.1	19.4	30.3	49.9	60.6	78.0	50.2	62.4	82.5	72.2	**80.7**	93.3	68.9	78.0	91.8	60.5	71.0	87.4
Italian	5.2	16.8	26.2	49.0	58.6	74.4	44.1	55.7	75.2	65.9	**74.7**	88.6	60.2	70.2	85.6	53.3	64.1	81.6
Dutch	5.3	17.3	26.5	53.2	62.9	78.6	47.7	59.6	79.2	73.0	**81.2**	93.2	69.8	78.6	91.7	60.1	70.4	86.8
Portuguese	5.3	17.0	26.6	45.5	55.8	72.7	44.1	56.2	75.8	68.5	**77.4**	90.3	65.6	74.8	88.8	58.3	68.4	84.6
Turkish	6.2	19.0	31.0	49.3	59.2	75.2	43.5	55.7	76.4	70.2	**78.8**	91.6	64.5	74.5	89.6	65.7	76.1	91.4
Russian	7.1	21.7	35.7	48.5	59.2	76.7	49.8	63.1	83.8	73.5	**82.1**	94.4	68.9	78.7	93.1	61.0	71.6	88.3
Polish	6.1	19.4	30.4	49.9	59.2	74.9	47.4	59.9	81.2	77.6	**85.2**	96.0	73.2	81.6	94.6	64.0	73.9	89.8
Indonesian	8.0	23.8	40.1	61.8	71.3	86.4	49.3	62.1	83.5	82.2	**88.5**	97.2	76.6	84.3	95.4	-	-	-
Norwegian	5.5	17.8	27.5	48.8	58.9	75.9	25.3	36.9	57.4	76.5	**83.1**	93.7	70.6	79.4	93.3	-	-	-
Swedish	5.0	16.1	25.0	49.1	59.3	76.1	25.7	36.7	56.3	75.6	**83.3**	94.6	72.0	80.5	93.1	-	-	-
Danish	5.6	18.1	27.8	54.3	64.0	80.6	30.4	42.1	63.3	75.4	**82.7**	92.9	71.5	79.5	91.8	-	-	-
Vietnamese	11.3	30.6	56.6	62.7	73.3	90.6	28.4	43.2	70.7	73.9	**81.2**	92.5	69.8	78.3	91.6	-	-	-
Finnish	5.6	18.3	28.2	43.9	53.5	70.0	21.8	33.2	53.7	74.9	**82.6**	93.9	68.5	76.8	89.7	-	-	-
Romanian	6.4	20.3	32.1	48.6	57.8	73.0	29.3	40.7	62.0	76.9	**83.2**	91.5	66.6	75.4	88.2	-	-	-
Czech	3.8	12.9	19.0	38.3	48.2	64.0	18.3	26.9	42.4	59.6	**69.0**	83.5	50.1	60.2	77.2	-	-	-
Hebrew	8.6	25.1	42.8	49.3	61.5	81.3	14.5	26.5	50.7	75.3	**83.6**	95.5	68.8	78.7	93.7	-	-	-
Hungarian	4.1	13.3	20.4	30.3	38.1	52.1	21.0	28.6	41.4	60.6	**69.7**	83.7	54.1	64.1	80.5	-	-	-
Croatian	4.9	15.9	24.5	49.4	58.1	73.0	32.8	41.4	56.7	78.2	**83.6**	92.6	71.8	79.4	91.4	-	-	-

Table B.1: Results of our experiments on MFAQ. XLM-RoBERTa (1 page per domain) is consistently better than the rest, except for English where a RoBERTa model achieves a higher MRR. P@1 = Precision-at-1 (accuracy), MRR = Mean Reciprocal Rank, R@5 = Recall-at-5, One page per domain = subset of the training set.

ID	Question	Answer
0	Are international students eligible for the MSBA program?	Yes, international students are eligible for the MSBA program. Please review the International Applicants page for specific requirements.
1	Are the hours flexible enough for full-time working adults?	Yes, the MSBA program accommodates students working full-time. Required weekly live sessions, lasting 75 minutes, are held in the evening, and the three residential components, two strongly recommended and one optional, take place over weekends. Students complete all other coursework on their own schedule, but must adhere to deadlines and be prepared to participate in weekly live sessions.
2	Can I take a course from a third-party provider, like Lynda or Coursera, to prepare for the programming requirements of this program?	Our goal is to make sure that everyone entering the program has the necessary background to be successful. We strongly recommend that applicants who feel they need additional preparation in programming languages take a for-credit course from an accredited two- or four-year institution.
3	Can I transfer credits into the program?	No, the Tepper School does not accept transfer credits.
4	Can the GMAT or GRE requirement be waived?	No, these test scores are required.
5	Do I have to maintain a certain GPA in the program to graduate?	Yes, MSBA degree candidates must maintain a minimum cumulative GPA of 3.0 to graduate.
6	Do you offer the opportunity to preview courses in your program to get a feel for what they are like?	Yes we do. To preview one of our courses, please visit our Virtual Class Visit page. You'll be able to register to virtually participate in a course of your choosing.
7	How do I learn more about the online learning environment?	To preview one of our courses, please visit our Virtual Class Visit page. You'll be able to view upcoming courses and register to virtually attend a course of your choosing.
8	How many hours per week should be dedicated to coursework?	Students take two classes at a time and should expect to spend at least 10 hours on each course, or 20 hours total for the week. Coursework includes live synchronous meetings, assignments, projects, readings, and quizzes.
9	If I need to withdraw from the program, will I get a refund?	If I need to withdraw from the program, will I get a refund?
10	If I'm already proficient in basic programming and probability/statistics, do I have to take these courses?	Yes, the 46-880 Introduction to Probability and Statistics and 46-881 Programming in R and Python courses are required for all MSBA students. These courses ensure that all students have the necessary skills and knowledge to succeed in courses that follow. For more information, visit the Curriculum page on our website.
11	Is the MSBA offered exclusively on campus?	No, the MSBA degree is offered only online, with three optional on-campus experiences. Though they all are optional, we strongly recommend that students attend the BaseCamp and Capstone Project experiences, which occur at the beginning and end of the degree program.
12	Is the MSBA program structured in cohorts?	Yes, the part-time, online MSBA is structured in cohorts to optimize student interaction and success in the program.
13	Is the Tepper School participating in the Yellow Ribbon Program?	Yes, the Tepper School is participating in the Yellow Ribbon Program. For more information, please visit the Tuition page or contact Mike Danko at uro-vaedbenefits@andrew.cmu.edu.
14	Is there a Tuition Payment Plan available?	Yes, for more information about a monthly payment plan and debt minimization services, please review our payment options.
15	It's a part-time online program, but are there any on-campus opportunities for students?	We have three on-campus experiences. The first is an orientation basecamp, where the students are introduced to the program, interact with faculty, and learn about their cohort. The second, an immersive analytics experience led by top CMU faculty, takes place mid-program. [...]
16	I'm an active duty soldier/veteran. Am I eligible for an application fee waiver?	Yes, as a GMAC military-friendly business school, we waive the $125 application fee for active duty U.S. military personnel, veterans and retirees. Please contact Mike Danko at uro-vaedbenefits@andrew.cmu.edu to discuss the fee waiver.
17	Must international students come to campus?	We recommend attendance at the on-campus experiences, but students who are unable to attend may participate remotely and still meet the requirements of the program. Please note that because the program is delivered online, enrollment in the MSBA will not qualify students for a student visa to enter the United States.
18	What are some examples of roles a graduate could pursue after the program?	Business analytics professionals hold a range of positions across sectors and industries. They have titles such as business intelligence analyst, operations research analyst, market research analyst and statistician. Other job titles for these professionals are available here.
19	What are the programming languages that I should have experience in before applying to the program?	Basic programming knowledge in a modern language is required for admission. You do not need to be familiar with any specific language or build advanced programming skills before applying to the MSBA program. Your courses in the program will introduce you to relevant languages and provide hands-on experience.
20	What are the technical requirements for the MSBA program?	All students must have access to the following technologies in order to participate in the program: Laptop with the following requirements: - Windows – Intel Core i5 processor or higher; 8GB RAM, 256+ hard drive capacity - Macintosh [...]
21	What career resources are available for MSBA students and alumni?	The Master's Career Center helps students develop strategies focused on their career needs through a variety of services. For example, the career center hosts workshops and webinars in job search fundamentals, such as resume writing, interviewing, and networking. [...]
22	What happens if I need to defer starting or withdraw from the program?	Deferrals are granted only if an applicant must complete military service or has an extreme emergency. Deposits are refunded in these instances. Students are re-admitted the following year and must submit their deposit before the deadline for their start date. [...]
23	What is the average Quant and Verbal scores for the GRE and GMAT?	There is no average score expectation. The test scores are simply one component of the multifaceted admissions process that we consider when making an admissions decision.
24	What separates the Tepper School of Business' online MSBA program from other MSBA programs, either online or on-campus?	The Tepper School of Business is globally renowned for its analytical approach to business problem solving. It is an integral part of Carnegie Mellon University, a top-tier research university that has become the center for disciplines including data science, robotics, business intelligence and additive manufacturing. [...]
25	What time(s) do the synchronous sessions take place?	The weekly live sessions are in the evening (U.S. Eastern Time) and typically last 75 minutes.
26	What types of financial aid or scholarships are available to online students?	Students may be eligible to take out federal and/or private education loans to cover tuition and other education-related costs. Please view our Tuition page for details. At this time, the Tepper School does not provide scholarships for the MSBA program.

Table B.2: FAQ pairs from the Tepper School of Business.

Appendix C

Appendix Chapter 6

C.1 Distribution of Intents & Slots

We list in Table C.1 the distribution of intents across the three datasets. Table C.2 shows the distribution of slots across the three datasets.

C.2 Logistic Regression by Languages

We list the results of the logistic regression by language in Table C.3.

Intent	MASSIVE	Generated	Synthetic
calendar_set	7,0%	2,6%	3,6%
play_music	5,5%	2,2%	3,3%
weather_query	5,0%	2,0%	3,0%
calendar_query	4,9%	2,2%	2,4%
general_quirky	4,8%	1,7%	5,2%
qa_factoid	4,7%	2,0%	8,3%
news_query	4,4%	2,1%	2,5%
email_query	3,6%	2,2%	12,0%
email_sendemail	3,1%	2,4%	11,1%
datetime_query	3,0%	1,5%	1,4%
calendar_remove	2,7%	2,1%	1,1%
play_radio	2,5%	2,1%	1,5%
social_post	2,5%	2,4%	8,7%
qa_definition	2,3%	2,2%	3,7%
transport_query	2,0%	2,3%	1,1%
cooking_recipe	1,8%	2,2%	1,2%
lists_query	1,7%	1,5%	1,0%
play_podcasts	1,7%	1,5%	1,0%
recommendation_events	1,7%	2,0%	0,9%
alarm_set	1,6%	1,8%	0,6%
lists_createoradd	1,5%	1,7%	0,6%
recommendation_locations	1,5%	2,3%	0,9%
lists_remove	1,4%	1,7%	0,9%
music_query	1,3%	1,3%	0,6%
iot_hue_lightoff	1,3%	1,3%	0,6%
qa_stock	1,3%	2,5%	2,7%
play_audiobook	1,3%	2,0%	0,3%
qa_currency	1,2%	2,2%	3,3%
takeaway_order	1,2%	2,1%	0,4%
alarm_query	1,1%	1,3%	0,2%
email_querycontact	1,1%	2,0%	3,3%
transport_ticket	1,1%	1,8%	0,6%
iot_hue_lightchange	1,1%	2,1%	0,7%
iot_coffee	1,1%	1,2%	0,5%
takeaway_query	1,1%	1,8%	0,5%
transport_traffic	1,0%	1,8%	0,4%
music_likeness	1,0%	1,5%	0,5%
play_game	1,0%	1,7%	0,7%
audio_volume_up	1,0%	1,2%	0,1%
audio_volume_mute	1,0%	1,5%	0,3%
social_query	0,9%	2,0%	2,8%
transport_taxi	0,9%	1,9%	0,5%
iot_cleaning	0,8%	1,4%	0,4%
alarm_remove	0,7%	1,8%	0,2%
qa_maths	0,7%	1,7%	0,8%
iot_hue_lightup	0,7%	1,3%	0,4%
iot_hue_lightdim	0,7%	1,4%	0,4%
general_joke	0,6%	1,3%	0,3%
recommendation_movies	0,6%	2,0%	0,4%
email_addcontact	0,5%	1,3%	1,4%
iot_wemo_off	0,5%	0,8%	0,2%
datetime_convert	0,5%	1,6%	0,2%
audio_volume_down	0,5%	1,1%	0,1%
music_settings	0,4%	0,9%	0,2%
iot_wemo_on	0,4%	1,0%	0,2%
general_greet	0,2%	0,2%	
iot_hue_lighton	0,2%	1,0%	0,1%
audio_volume_other	0,2%	0,6%	0,0%
music_dislikeness	0,1%	0,9%	0,1%
cooking_query	0,0%	0,0%	0,0%

Table C.1: Distribution of intents across the three datasets. Generated represents the utterances generated by GPT-3, while synthetic represents the synthetic training set of SLURP.

Intent	MASSIVE	Generated	Synthetic
date	16,0%	10,8%	10,7%
place_name	9,6%	10,6%	8,0%
event_name	8,8%	4,3%	5,5%
person	7,6%	5,4%	17,2%
time	7,0%	5,8%	4,1%
media_type	4,2%	5,4%	9,5%
business_name	3,4%	5,7%	7,6%
weather_descriptor	2,8%	1,1%	1,5%
transport_type	2,8%	5,0%	1,2%
food_type	2,6%	4,2%	1,4%
relation	2,2%	2,3%	4,8%
timeofday	2,1%	2,0%	1,3%
artist_name	2,0%	0,8%	1,2%
device_type	2,0%	3,4%	1,1%
definition_word	2,0%	2,0%	3,5%
currency_name	1,9%	3,8%	5,7%
house_place	1,7%	3,8%	0,8%
list_name	1,7%	1,8%	0,9%
business_type	1,7%	2,8%	0,8%
news_topic	1,6%	0,7%	1,1%
music_genre	1,6%	0,9%	1,0%
player_setting	1,4%	2,1%	0,5%
radio_name	1,2%	1,1%	0,9%
song_name	1,1%	0,3%	0,7%
order_type	0,9%	1,6%	0,3%
color_type	0,9%	1,7%	0,4%
game_name	0,8%	1,3%	0,6%
general_frequency	0,7%	0,3%	0,4%
personal_info	0,7%	1,2%	2,0%
audiobook_name	0,6%	0,9%	0,2%
podcast_descriptor	0,6%	0,6%	0,3%
meal_type	0,6%	0,4%	0,4%
playlist_name	0,5%	0,1%	0,3%
podcast_name	0,5%	0,4%	0,3%
time_zone	0,5%	1,1%	0,2%
app_name	0,4%	0,3%	0,1%
change_amount	0,4%	0,9%	0,1%
music_descriptor	0,4%	0,2%	0,2%
joke_type	0,3%	0,8%	0,2%
email_folder	0,3%	0,2%	0,9%
email_address	0,3%	0,4%	1,4%
transport_agency	0,3%	0,5%	0,2%
coffee_type	0,2%	0,2%	0,1%
ingredient	0,2%	0,1%	0,1%
cooking_type	0,1%	0,1%	0,1%
movie_name	0,1%	0,1%	0,1%
movie_type	0,1%	0,2%	0,0%
transport_name	0,1%	0,1%	0,1%
drink_type	0,1%	0,1%	0,0%
alarm_type	0,1%	0,1%	0,0%
transport_descriptor	0,1%	0,0%	0,0%
audiobook_author	0,1%	0,2%	0,0%
sport_type	0,0%	0,0%	0,0%
music_album	0,0%		0,0%
game_type	0,0%	0,0%	0,0%

Table C.2: Distribution of slots across the three datasets. Generated represents the utterances generated by GPT-3, while synthetic represents the synthetic training set of SLURP.

language	β_0	β_1	R_2	$f(x=0)$	$f(x=0.5)$	$f(x=1)$
all	-0.69	3.14	0.08	0.33	0.71	0.92
af-ZA	-0.98	4.01	0.11	0.27	0.74	0.95
am-ET	-0.46	3.09	0.06	0.39	0.75	0.93
ar-SA	-0.58	3.01	0.07	0.36	0.72	0.92
az-AZ	-0.55	3.24	0.08	0.37	0.75	0.94
bn-BD	-1.27	3.71	0.10	0.22	0.64	0.92
cy-GB	-0.66	3.37	0.08	0.34	0.74	0.94
da-DK	-0.95	4.13	0.12	0.28	0.75	0.96
de-DE	-0.65	3.58	0.09	0.34	0.76	0.95
el-GR	-0.92	3.64	0.09	0.28	0.71	0.94
en-US	-1.45	4.93	0.21	0.19	0.73	0.97
es-ES	-0.60	2.99	0.07	0.36	0.71	0.92
fa-IR	-0.96	2.70	0.06	0.28	0.60	0.85
fi-FI	-0.86	3.80	0.10	0.30	0.74	0.95
fr-FR	-0.37	2.65	0.05	0.41	0.72	0.91
he-IL	-0.72	3.44	0.08	0.33	0.73	0.94
hi-IN	-0.76	3.10	0.08	0.32	0.69	0.91
hu-HU	-0.55	3.25	0.08	0.37	0.75	0.94
hy-AM	-1.05	3.35	0.08	0.26	0.65	0.91
id-ID	-0.67	3.33	0.08	0.34	0.73	0.93
is-IS	-0.56	3.19	0.07	0.36	0.74	0.93
it-IT	-0.46	2.82	0.06	0.39	0.72	0.91
ja-JP	-0.48	2.77	0.06	0.38	0.71	0.91
jv-ID	-0.34	2.95	0.06	0.42	0.76	0.93
ka-GE	-0.46	2.59	0.06	0.39	0.70	0.89
km-KH	-0.23	1.62	0.03	0.44	0.64	0.80
kn-IN	-0.94	2.55	0.05	0.28	0.58	0.83
ko-KR	-0.49	3.42	0.08	0.38	0.77	0.95
lv-LV	-0.81	3.62	0.09	0.31	0.73	0.94
ml-IN	-1.39	3.64	0.10	0.20	0.61	0.90
mn-MN	-0.79	3.32	0.07	0.31	0.70	0.93
ms-MY	-0.77	3.55	0.08	0.32	0.73	0.94
my-MM	-0.97	4.12	0.08	0.27	0.75	0.96
nb-NO	-0.72	3.65	0.09	0.33	0.75	0.95
nl-NL	-0.80	3.71	0.10	0.31	0.74	0.95
pl-PL	-0.52	2.65	0.06	0.37	0.69	0.89
pt-PT	-0.56	3.05	0.07	0.36	0.72	0.92
ro-RO	-0.36	3.00	0.06	0.41	0.76	0.93
ru-RU	-0.47	3.12	0.07	0.38	0.75	0.93
sl-SL	-0.63	3.25	0.08	0.35	0.73	0.93
sq-AL	-0.54	3.04	0.07	0.37	0.73	0.92
sv-SE	-0.51	3.53	0.09	0.37	0.78	0.95
sw-KE	-0.89	3.26	0.08	0.29	0.68	0.91
ta-IN	-0.70	3.20	0.07	0.33	0.71	0.92
te-IN	-0.65	2.18	0.04	0.34	0.61	0.82
th-TH	-0.66	2.61	0.06	0.34	0.66	0.88
tl-PH	-1.12	3.72	0.09	0.25	0.68	0.93
tr-TR	-0.71	3.53	0.09	0.33	0.74	0.94
ur-PK	-0.80	3.30	0.08	0.31	0.70	0.92
vi-VN	-1.72	3.78	0.10	0.15	0.54	0.89
zh-CN	-0.42	2.35	0.06	0.40	0.68	0.87
zh-TW	-0.56	1.97	0.05	0.36	0.61	0.80

Table C.3: Logistic regression results by language

Appendix D

Appendix Chapter 7

D.1 Computing Infrastructure

We ran all our experiments on a server running 8 NVIDIA GPU (12GB) with 128GB of RAM and 24 CPU. All models ran in parallel using the device_map argument of the from_pretained method.

D.2 Hyperparameter Search

We did not engage in a hyperparameter search. Future research could look for the optimal prompt, and the balance of yes and no examples.

D.3 Correlation With Token Frequency

We display the correlation between the average accuracy of an entity and its relative frequency in the pre-training data in Table D.1.

Model	Correlation	P-value
GPT-3	-0.02	0.35
T0	0.05	0.01
T0-KG	-0.01	0.45

Table D.1: Spearman correlation of the average accuracy of an entity with its frequency in the pre-training data.

Appendix E

Appendix Chapter 8

E.1 Detailed Overlap Analysis

In this section, we review the most similar pairs of questions between the training and test for Commonsense QA 2.0, Com2sense, and 20Q (our benchmark). We use Sentence Transformers (Reimers and Gurevych, 2019) to compute the similarity between all pairs of questions in the training and test set.

E.1.1 Commonsense QA 2.0

The authors of Commonsense QA 2.0 used a topical split to divide the training and test set. We list the top 15 most overlapping questions between the training and test set in Table E.6. A quick analysis of the table reveals a number of problematic pairs such as « *an electron holds a positive charge and* » is an almost duplicate to « *an electron hold a positive charge* ».

E.1.2 Com2sense

Our overlap analysis of com2sense reveals three *exact duplicates* between the training and test set of Com2sense. A number of examples are close duplicates and only differ in one word or punctuation. For example « *if it is dark outside, opening the blinds will not help you see* » and « *if it is dark outside opening the blinds will help you see* ». We list the top fifteen overlapping pairs in Table E.7.

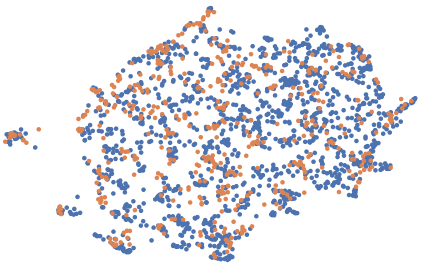

Figure E.1: UMAP projection of the Sentence Transformers representation of the questions. Blue dots belong to the training set, red dots belong to the validation set.

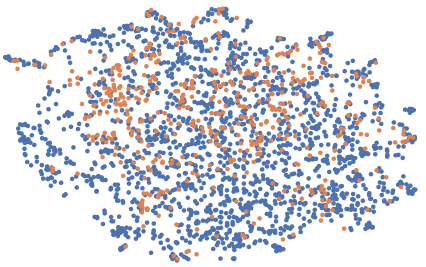

Figure E.2: UMAP projection of the Sentence Transformers representation of the topics. Blue dots belong to the training set. Red dots belong to the validation set.

E.1.3 20Q

Our overlap analysis of 20Q does not reveal any overlap thanks to our strict pre-processing pipeline. We list the top fifteen overlap pairs in Table E.5.

E.1.3.1 UMAP

Figure E.1 and E.2 provide a 2 dimension projection of the semantic of questions and subject in 20Q.

E.2 Pre-processing

The original Twenty Questions dataset is generated by humans, and is thus extremely noisy. In this section, we expand upon Section 8.3.3 and go into the details of our pre-processing steps. We detail our pre-processing steps and the percentage of questions removed in Table E.1.

Step	Size (abs)	Size (%)
Initial dataset	78,890	100
Low scores	-12,396	-15.7
Do not use "it"	-9,665	-12.3
Duplicates	-2,708	-3.4
WordNet	-2,312	-2.9
Clean dataset	51,809	65.7

Table E.1: Pre-processing of the original dataset. We are aggressive in our pre-processing as we prefer a small dataset of high quality to the reverse. First, we remove all questions with a score of 2 (the maximum is 3). We then remove all sentences that do not use "it." Next, we use a stemmed bag-of-words representation to remove close duplicates. Finally, we remove all questions where the answer is not in WordNet.

E.2.1 Quality Score

We start our pre-processing by removing all sentences with a score below three. These are questions which are not answerable with *yes* or *no*, or questions which are not playing the game of Twenty Questions. For example, questions such as « *so not an object, but tangible. is it edible* » which references the previous turn, or simple one word questions such as « *mountain?* »

E.2.2 Use of *it*

Our goal is to understand the world knowledge of language models. For some models such as T0 or T5, it may be easier to answer the question if the topic is part of the question, instead of having two separated parts. For example it is easier to answer: « *does a rock float* » than « *subject: rock, question: does it float* ». To make sure all questions are equally easy or difficult in terms of lexical information, we only keep questions of the latter format.

E.2.3 Duplicate Questions

Some questions may be close, but not exact, duplicates. We want to avoid such questions in the training or test set as these add very little information while artificially inflating the size of the dataset. We use a stemmed bag-of-words approach to detect these questions. For example, questions such as « *is it animal* » and « *is it an animal* ».

Train	Validation	Sim.
Account	Accountant	0.84
Thinking	Thing	0.79
Constitution	Institution	0.78
Extraction	Traction	0.78
Attraction	Traction	0.78

Table E.2: Most similar pairs of topics between the training and validation set using a character tri-gram method.

E.2.4 WordNet Filtering

We want to avoid having questions where the subject is not orthographically correct. We remove all questions where the subject is not present within WordNet. In effect, this will remove words such as *trex*, *chldren*, *voiceing*, or acronym words such as *potus* or *49ers*.

E.3 Topic Overlap Exploration

In this section, we show the list the overlapping topics according to three different metrics.

E.3.1 N-grams

We show the five most similar pairs of topics between the training and validation set in Table E.2.

E.3.2 WordNet

We use WordNet to compute the distance between two topics by following the hypernym or hyponym chain. Table E.3 shows this technique's most similar pairs of topics.

E.3.3 Sentence Transformers

We finish our qualitative review of the topic overlap using Sentence Transformers. Table E.4 shows the five most similar pairs of topics.

Train	Validation	Sim.
Vegetation	Galaxy	0.33
Purifier	Pendulum	0.33
Lambskin	Squirrel	0.33
Foil	Steel	0.33
Repellent	Menthol	0.33

Table E.3: Most similar pairs of topics between the training and validation set using the WordNet method.

Train	Validation	Sim.
Costume	Halloween	0.60
Chlorophyll	Chrysanthemum	0.60
Housekeeper	Groomsman	0.60
Bracelet	Pendant	0.60
Forearm	Ankle	0.60

Table E.4: Most similar pairs of topics between the training and validation set using the Sentence Transformers method.

Test Set	Training Set
would it [a granite] be of rock material?	can it [a rock] be molded?
is it [a window] see through?	does it [a curtain] cover a window?
is it [a sweat] produced by the human body?	does it [an exercise] involve sweating?
does it [a hyacinth] have red flowers?	does it [a chrysanthemum] have a long stem?
is it [a ring] jewlery?	does it [a treasure] go on engagement rings?
is it [a bridge] larger than a car?	is it [a bumper] a bridge?
is it [a refuge] a type of campsite?	is it [a campground] the mountains?
is it [an ant] bigger than a honeybee?	does it [a honeybee] collect nectar?
is it [a marsupial] a kind of bear?	is it [a bear] long?
does it [a hyacinth] have white flowers?	does it [a chrysanthemum] have a long stem?
is it [a pendant] jeweled?	does it [a treasure] go on engagement rings?
does it [a hyacinth] have yellow flowers?	does it [a chrysanthemum] have a long stem?
is it [a ship] larger than a whale?	does it [a whale] have fins?
is it [a hurdle] made of stone or rock?	can it [a rock] be molded?
is it [a fly] a bug?	does it [an insect] have antennae?

Table E.5: Top fifteen most similar pairs of questions between the training and test set of 20Q.

Test Set	**Training Set**
an electron holds a positive charge and	an electron holds a positive charge.
happy meals almost always come with a toy.	most happy meals include a toy.
april is larger than february	april is smaller than march
sunlight on the skin causes eye cancer	sunlight causes almost all skin cancer
thunder sounds before lightning strikes	noise of thunder is heard before the lightning.
the beginning of a story is part of the end	a story has a beginning and an end.
is there a feminine french word for a city hall?	in french is it true that there are feminine and masculine words for a city hall?
europe is considered to be the most wealthy and richest continent.	europe has the richest countries in the world
a grapefruit is a fruit larger than a watermelon?	is a watermelon smaller than an apple?
tree is always part of forest	trees are never part of forests
someone of the male gender cannot give birth.	an adult male cannot give birth
if you add two plus two you will always get four.	two plus two unfortunately cannot ever add up to anything but four.
you can return items to a store only if you have a receipt.	an item can be returned from a store only if it is sold by that store.
private is another way to say public	private almost never means public.
a letter can be written with invisible ink.	writing cannot be read if you use invisible ink.

Table E.6: Top fifteen most similar pairs of questions between the training and test set of Commonsense QA 2.0.

Test Set	Training Set
john leaves work at 6 pm so that he is an unlikely suspect for theft that happened in the office at 8 pm.	john leaves work at 6 pm so that he is an unlikely suspect for theft that happened in the office at 8 pm.
while in a windy rainstorm, you should always point your umbrella away from the wind.	while in a windy rainstorm, you should always point your umbrella away from the wind.
while in a windy rainstorm, you should always point your umbrella into the wind.	while in a windy rainstorm, you should always point your umbrella into the wind.
since i want to improve my golf skill quickly, i spend 2 hours on the course every day.	since i want to improve my golf game, i spend 2 hours on the course every day.
if it is dark outside, opening the blinds will help you see.	if it is dark outside opening the blinds will not help you see.
because it was halloween eve and we had no candy, i decided to open the door and turn the porch light on.	because it was 6pm on halloween and we no candy, i decided to open the door and turn the porch light on.
having to teach a night class in thirty minutes, he should cook a three-course dinner instead of heating a frozen meal.	having to teach a night class in thirty minutes, he should make a three-course dinner instead of a frozen meal.
danny smokes a lot and drinks thirty beers per week while sarah doesn't smoke and doesn't drink, sarah will probably live longer.	danny smoke a lot and drink thirty beer per week while sarah dont smoke and dont drink, sarah will probably live longer.
if it is dark outside, opening the blinds will not help you see.	if it is dark outside opening the blinds will not help you see.
because it was halloween eve and we had plenty of candy, i decided to open the door and turn the porch light on.	because it was 6pm on halloween and we had plenty of candy, i decided to open the door and turn the porch light on.
having to teach a night class in thirty minutes, he should heat a frozen meal instead of cooking a three-course dinner.	having to teach a night class in thirty minutes, he should make a frozen meal instead of a three-course dinner.
danny smokes a lot and drinks thirty beers per week while sarah doesn't smoke and doesn't drink, danny will probably live longer.	danny smoke a lot and drink thirty beer per week while sarah dont smoke and dont drink, danny will probably live longer.
a spoon is more suitable for eating soup than a fork.	a spoon might be more suitable for eating soup than a fork.
it is easier to run one mile in 5 minutes than a half mile in 10 minutes.	it is easier to run two miles in five minutes than it is to run one mile in ten minutes.
a fork is more suitable for eating soup than a spoon.	a spoon might be more suitable for eating soup than a fork.

Table E.7: Top fifteen most similar pairs of questions between the training and test set of Com2sense.