



Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding

Mutian He^{1,2}, Philip N. Garner¹

¹ Idiap Research Institute, Martigny, Switzerland

² Ecole Polytechnique Fédérale de Lausanne, Switzerland

{mutian.he, phil.garner}@idiap.ch

Abstract

Recently, large pretrained language models have demonstrated strong language understanding capabilities. This is particularly reflected in their zero-shot and in-context learning abilities on downstream tasks through prompting. To assess their impact on spoken language understanding (SLU), we evaluate several such models like ChatGPT and OPT of different sizes on multiple benchmarks. We verify the emergent ability unique to the largest models as they can reach intent classification accuracy close to that of supervised models with zero or few shots on various languages given oracle transcripts. By contrast, the results for smaller models fitting a single GPU fall far behind. We note that the error cases often arise from the annotation scheme of the dataset; responses from ChatGPT are still reasonable. We show, however, that the model is worse at slot filling, and its performance is sensitive to ASR errors, suggesting serious challenges for the application of those textual models on SLU.

Index Terms: spoken language understanding, pretrained language models, zero-shot learning, in-context learning

1. Introduction

Gigantic pretrained language models like GPT3 with 175B parameters trained on 45TB texts have demonstrated surprisingly strong performance on various downstream language tasks with little or no data [1]. Since then, GPT3 has evolved into GPT3.5 through pretraining on code as in Codex [2] which powers GitHub Copilot, as well as through instruction fine-tuning that aligns the model's responses given instructions with human expectations using reinforcement learning, known as Instruct-GPT [3]. When further combined with fine-tuning on dialogues in a similar way, the resulting model, ChatGPT, has gained great popularity since its release in late 2022, displaying highly human-like language understanding and generation capabilities [4, 5, 6], and has become the core of a number of AI-powered applications. Combined with the ability to utilize tools with external APIs as in Toolformer [7] as well as conducting web search as in WebGPT [8] and New Bing, a competent and versatile AI assistant has taken shape. Therefore, a question arises: is the model capable of conducting spoken language understanding (SLU) tasks like current voice assistants?

Current SLU approaches are substantially different from how we use those GPT3-based models. Traditionally, SLU is carried out using a cascaded pipeline, which includes an automatic speech recognition (ASR) module taking audio as inputs, and a natural language understanding (NLU) module working on ASR transcripts, hypotheses, or lattice to predict labels for tasks like intent classification (IC) and slot filling (SF) [9, 10]. Recently, end-to-end approaches that directly predict labels from speech [11, 12, 13] become more popular, and pre-

trained language and speech models are also introduced into SLU [14, 15, 16]. Additionally, there are works focused on low-resource or few-shot textual IC/SF [17, 18, 19, 20, 21].

However, those methods are based on the paradigm of supervised training or fine-tuning with a set of possibly large-scale training data. In contrast, considering the difficulty of fine-tuning the whole GPT3 model, recent NLP research highlights a different scheme, namely **prompting** [22]: given a fixed textual description of the task known as a *prompt* without any training data, the language model may correctly carry out the task. Furthermore, the **in-context learning** approach adds a few paired examples in the textual prompt to further direct the model towards the desired outputs. Such methods are different from traditional zero or few-shot learning in which the model parameters are fixed. It appears to be clumsy and may perform worse than fine-tuned smaller models like T5-11B at the beginning [23]. Additionally, larger models were believed to be unscalable to reach the desired performance given the costs [1, 24, 25]. However, recent explorations reveal the **emergent abilities** of larger models like GPT3-175B and PaLM 540B [26]: prompting shows low or even close-to-random performance on multiple tasks until a certain scale of the model where a breakthrough emerges [27, 28]. This breakthrough enables chain-of-thought prompting to surpass the smaller models fine-tuned on rich data [29, 30], allows reasoning using internal knowledge with results comparable to external knowledge retrievers [31], and leads to better robustness and generalization [30, 32].

There have been several works on SLU employing prompts, such as fine-tuning pretrained models like T5 aided by prompts [33, 34], fine-tuning embeddings prepended to the inputs known as continuous prompts [35, 36], and end-to-end SLU by in-context learning on GPT2 with a fine-tuned audio encoder [37]. They are nevertheless distinct from the current prompting and in-context learning scheme, and have not approached the regime of emergent abilities. Hence the potential and limitations of this new type of method on SLU remain unexplored. Therefore we endeavor to undertake it by designing prompts and evaluating these models on multiple SLU benchmarks, including SLURP [38] and the multilingual MINDS-14 [39]. Since these models take textual inputs, beside oracle transcripts, we also use ASR transcripts from Whisper [40], which embodies a pipeline upon completely off-the-shelf pretrained models. Furthermore, we compare smaller models that can easily run on a common GPU, namely GPT2 [41] and several OPT models [42].

As a result, we discover that the largest GPT3.5 and ChatGPT models achieve high performance on intent classification under zero-shot or few-shot in-context scenarios that are close or even better than models fine-tuned on the whole dataset, when given the oracle transcripts. This is unique to those large models as the smaller GPT2, OPT and GPT3.5 Curie models

User: We will show you some commands by a user to a voice assistant like Siri or Olly. Please determine that the command is under which of the following predefined scenarios [...]
 Audio: volume up, volume down, volume mute, volume other

Some examples are:
 1. "book a taxi to the airport for tomorrow morning": the scenario and intent is "Transport: taxi". [...]

Please give answers like: 1. Email: query contact, 2. IoT: cleaning, etc. The intent in your answer must match one of the intents for the corresponding scenario given above. If you are uncertain, choose the one that you think is the most likely.

How about the following commands:
 1. event reminder mona tuesday
 2. what is the exchange rate of us dollar to pound sterling [...]

ChatGPT:
 1. Calendar: set
 2. Question Answering: currency [...]

Figure 1: An example of ChatGPT doing SLURP intent classification in a conversation. The list of options, questions, and answers are partially omitted for brevity.

have much lower performance and are entirely ineffective under zero-shot cases. Even in the cases where the predictions differ from the labels, the predictions are mostly reasonable, often due to the ambiguity of the sentence. This raises the question that the tasks to predict intermediate IC/SF labels might not sufficiently reveal the potential of the model. However, for the slot filling task with a more complicated task definition, the performance is much worse. Additionally, the accuracy drops significantly when using ASR transcripts. We show that the models have limited awareness of word pronunciations and possible ASR errors, which poses challenges for directly deploying those models for real-world SLU. To facilitate reproduction, relevant resources and prompts are available at https://mutiann.github.io/papers/ChatGPT_SLU/.

2. Methods

We design prompts to be given to ChatGPT for the intent classification and slot filling tasks as in Figure 1. We begin by explaining the background of the task, and then provide options, like possible intents. We rewrite the names of the scenarios and actions in natural language, for example, with the underlines in the name removed. For in-context learning cases, several examples are further appended. Following by instructions about the answer format are the questions. Several questions are asked in a batch for better efficiency. For instance, in the case of ChatGPT on SLURP, we ask 45 questions in a conversation, with the first 5 questions appended to the instructions, and the remaining 40 questions split into 2 rounds to be asked. We then collect results from each line of the answer that corresponds with each question by text matching. If the answer could not be identified, we retry up to 3 times. As for ChatGPT, we use the *legacy* (a.k.a. *default*, codenamed *text-davinci-002-render-paid*) model available on the webpage, which gives slightly different results compared to the faster *default* (a.k.a. *turbo*, codenamed *text-davinci-002-render-sha*) model. For slot filling, we further explain the meaning of each entity type. While as for GPT3.5, we adopt the largest *text-davinci-003* model as well as the smaller *text-curie-001* model¹, using a slightly different prompt to fit the task of text completion. Since GPT3.5 is called

¹The sizes of Davinci and Curie are estimated to be comparable to GPT3 175B and 6.7B respectively, according to Eleuther AI as in <https://blog.eleuther.ai/gpt3-model-sizes/>

Table 1: Accuracy for various GPT3-based models on SLURP intent classification task with zero or few examples.

#Examples	0	10	20	30
GPT3.5	72.86%	74.55%	77.27%	77.44%
w/ bias	75.86%	75.59%	78.31%	77.87%
Curie w/ bias	5.01%	4.91%	3.80%	3.77%
ChatGPT	79.25%	80.33%	83.93%	80.16%
Turbo ver.	78.98%	80.03%	81.78%	79.62%

by the OpenAI API, more customization is allowed, including logit bias to ensure that only tokens that may appear in the answer could be generated. While as for smaller models, we use GPT2 large (774M) [41], the widely-used predecessor, as well as OPT, an open-sourced reproduction of GPT3 with various sizes available [42]. We picked the 1.3B, 2.7B, and 6.7B versions that could be easily run on a single GPU, though we have to use half-precision inference on the 6.7B one. We performed generation with an 8-beam search, using a prompt similar to GPT3.5 but perform text completion with one question at a time. Whisper-large is chosen for performing ASR [40].

The models are evaluated on the test split of two different datasets: SLURP is a large-scale IC and SF dataset of commands to voice assistants with over 141k samples annotated with 60 different intents formulated as scenario-action pairs, as well as 56 types of entities or slots. There have been limited evaluations on a task with so many different types, as current works are mostly focused on tasks with a small label space (e.g. positive/neutral/negative in sentiment analysis), though lower performance has been found on named entity recognition (NER) [6] and dialogue state tracking [5] with a smaller label space but share some similarity with this task. MINDS-14 is a smaller banking scenario IC dataset of 14 intents in 14 different languages, for which we use the XTREME-S split [43]. Evaluations are performed on the test split, while in-context training examples are picked from the training or validation split.

3. Experiments

3.1. Oracle texts

We first evaluate the models from OpenAI using oracle transcripts in SLURP for intent classification. As shown in Table 1, ChatGPT achieves an accuracy of 79.25% with zero shot, and 83.93% with only 20 examples. Despite there being 60 different types of intents in SLURP, ChatGPT achieves the result even without examples in many of the categories. The result is on par with, or not far from, many supervised NLU models trained on the full oracle transcripts dataset, like SF-ID with 82.25%, HerMiT with 84.84% [38], and conformer deliberation with 89.0% [44], or end-to-end methods like fine-tuning HUBERT with 89.38% [14]. While more training examples seem to have limited or no improvements, as shown by the results on 30 examples. This could be possibly due to the bloated length of the prompt. When we use a more verbose version of the prompt with 20 examples but with 1260 tokens, similar to the 30-example one, the accuracy drops to 81.41%. ChatGPT gives slightly better performance compared to GPT3.5 even with token bias, possibly thanks to its additional dialogue fine-tuning, while the faster Turbo version shows slightly inferior results.

The results obtained with smaller models are presented in Table 2 in addition to the Curie results above. It is evident that the performance falls far behind the presumed 175B GPT3.5

Table 2: Accuracy for smaller language models on SLURP intent classification task with zero or few examples. A 30-example prompt is too long for GPT2 with a 1024 token window.

#Examples	0	10	20	30
GPT2 (774M)	6.66%	8.88%	8.31%	-
OPT-1.3B	5.58%	10.69%	17.85%	17.01%
OPT-2.7B	7.06%	28.65%	26.66%	36.97%
OPT-6.7B	4.37%	28.18%	35.14%	42.40%

Table 3: Accuracy for intent classification on different languages from MINDS-14 with ChatGPT, compared to the supervised textual model using full data.

	en-US	fr-FR	pl-PL	ko-KR
0-shot	95.4%	97.4%	90.0%	89.2%
1-shot	97.9%	99.3%	96.1%	90.5%
LaBSE [39]	95.1%	93.1%	89.2%	91.4%

and ChatGPT models. The model size is apparently the most critical factor affecting the performance, while the number of examples also makes a difference. Relatively larger (e.g. 6.7B) models can better leverage the increased examples, while none of the zero-shot experiments work. This is in stark contrast to the largest models where the number of examples has limited impact and the zero-shot cases also show good results. This suggests that the largest models possess sufficient capabilities through pretraining to comprehend the task explanation and the input question without additional training examples. The role of the training examples is more to guide the model to align with the requirements of the task that are not elaborated in the task description. In contrast, in-context examples are more important for smaller models that lack sufficient internal knowledge.

We then evaluate the models on four different languages from MINDS-14 selected based on their proportion in the GPT3 training data: Among 118 languages in the data, 92.6% are in English by word count, while the 2nd-ranked French accounts for 1.8%. Other languages are exceptionally sparse: Polish (ranked 10th) for 0.16%, and Korean (28th) for 0.017%. Nevertheless, ChatGPT generalizes to all of these languages with similar or better performance compared to the supervised LaBSE [45] reported in [39], though using a different split. The model can almost perfectly solve the task on English and the rather low-resource French with zero shot, as well as Polish using only 14 examples or 1 shot per category to align with the task. Even on Korean with extremely sparse training data, the zero-shot and one-shot results are still satisfactory. This demonstrates that the model enjoys inherent multilinguality to generalize its strong language understanding capability to various languages.

However, when it comes to the slot filling task with a more complicated task definition, the situation is different. As in Table 4, the F1 score is poor and much lower than the models like HerMiT with 78.19% F1 [38]. Therefore, the approach to better leveraging ChatGPT on this task remains to be explored.

3.2. ASR transcription

We then evaluate the model on SLURP intent classification using ASR transcripts, which reflects the real SLU scenario. The SLURP recordings are transcribed with 16.7% test WER with the off-the-shelf Whisper, which is acceptable, considering that the audio is often noisy and an XLSR-based ASR sys-

Table 4: F1 score for slot filling with ChatGPT with different number of examples.

#Examples	10	20	30
ChatGPT	12.03%	13.00%	13.35%

Table 5: Accuracy for intent classification on SLURP with different number of examples using ASR transcripts.

#Examples	10	20	30
GPT3.5	64.78%	68.95%	68.64%
ChatGPT	72.89%	73.96%	72.85%

tem adapted to the SLURP data reports 15.5% test WER [46]. To our disappointment, the models suffer from significant performance drop processing ASR transcripts as in Table 5, even though the model is prompted to account for ASR errors.

Considering that GPT3 is capable of correcting language errors, we are inspired by the chain-of-thought prompts [29] to first instruct the model to try to correct the ASR errors, and then determine the scenario and action. Such a prompt with 5 examples brings 67.15% accuracy on GPT3.5, which alleviates the issue with a 2.4% improvement compared to the 10-example prompt, although it is still far from the oracle transcript case and leads to bloated prompt. We also attempt to directly instruct GPT3.5 to correct ASR errors, characterized by the replacement of words with similar pronunciation. Given 1000 error cases, WER increases from 25.6% to 32.4% after the correction. In most cases the sentences are more fluent, but ASR errors remain. Hence we hypothesize that due to the nature of training on textual inputs with wordpiece tokenization, the model has limited awareness of phonetics, as implied by their high error rates on the IPA task [27]. We further verify that by asking ChatGPT the names of 50 well-known places with their Romanized transliteration in Chinese and Japanese, e.g., “A city is called Nyūyōku in Japanese. What’s its name in English?”, which can be easily guessed by a person knowing the common pronunciation of Latin letters. However, ChatGPT can correctly answer only 46% from Japanese and 10% from Chinese, which further confirms its deficiency in phonetic knowledge.

4. Discussions

Experiments have demonstrated the capability of the model, though many errors are found. However, it is worthwhile to look into the error cases to determine if ChatGPT really failed to understand those text, especially considering that SLURP labels are relatively noisy [46]. Therefore, we check 100 error cases in the 20-shot ChatGPT experiments, and find that 79% of them are not errors in the strict sense, as exemplified in Table 6. Some of the “errors” occur when the input sentence is ambiguous and open to multiple different interpretations. An example is shown in Figure 2, in which case we find that ChatGPT could actually provide the correct answer and explain the answer if we provide clarification that is not given in the command. There are also some examples for which ChatGPT gives a more accurate answer than the label. While in many other cases the issues arise due to the ambiguity of the categories, which are often specified by the annotation scheme of the SLURP dataset. For example, asking for news about stock is labeled as “Question answering: stock,” not “News: query.” This is a fundamental weakness of prompting: the instructions, examples, and label names in the

Table 6: Examples of “errors” made by ChatGPT on SLURP intent classification and slot filling. Many of the predictions are also reasonable, but different from the labels in the dataset.

Command	Label	ChatGPT Prediction
Is there a groomer in town for cats only?	General: quirky	Recommendation: locations
In how many hours will it be midnight in London England.	Date/time: query	Date/time: convert
Please tell me news related to the stock market.	Question answering: stock	News: query
Olly I need a drink.	IoT: coffee	General: quirky
Can you give me the details of upcoming annual function on twenty sixth march?	General: quirky	Recommendation: events
How long does it take to make vegetable lasagna?	Cooking: recipe	Cooking: query
Give details of rock sand.	Question answering: definition	Question answering: factoid
Event reminder Mona Tuesday.	Event name: Mona; Date: Tuesday	Alarm type: reminder; Person: Mona; Date: Tuesday
Exchange rate of US dollar to pound sterling.	News topic: exchange rate of US dollar to pound sterling	Definition word: exchange rate; Currency name: US dollar; Currency name: pound sterling
Take out the milk from the shopping list.	List name: shopping	List name: shopping; Ingredient: milk
Increase the brightness of the lights.	-	Change amount: brightness; Device type: lights
Please send a mail to my friend Divya how are you.	Relation: friend; Person: Divya	Email address: Divya; Person: friend
Please scan my social media and tell me what’s happening.	-	Media type: social media; Definition word: what’s happening

User: great song for commute
ChatGPT: The scenario and intent is “Recommendation: music”
User: The user might be asking the assistant to recommend a song for the commute unknown to the user. But what if we already know what song the user is talking about, as the device is playing it at the moment?
ChatGPT: If the device is already playing a song and the user says “great sone for the commute”, then the intent could be “Music: likeness” as the user is expressing their enjoyment of the current song and uggesting that it would be a good choice for a future commute.

Figure 2: An example for ChatGPT giving an answer different from the label (“Music: likeness”) on an ambiguous command. While ChatGPT is able to determine the answer that matches the label under an alternative situation.

prompt must fully reveal the goal of the task if it is not typically observed in the textual corpus, which is difficult when the task or labeling specification is complicated and subtle, even if the task itself is straightforward. Such annotation specifications could be captured by a supervised model given the whole training set, but impossible for a zero-shot or in-context model.

The phenomenon is more pronounced in the slot filling case where multiple entities with 56 different types could be extracted from a single command. The meanings of the labels often overlap with each other, and the annotations tailored to the specific scenario of the dataset. For example, normal apps are extracted as “app name,” and companies as “business name”, but Uber as “transport agency” and Northern Rail as “transport name,” which would baffle a zero/few-shot model as in Table 6. The harm of such counter-intuitive labelling scheme also matches the observations in [47]. This kind of misalignment has more impact in such a tagging task as tagging decisions need to be made on every individual word, which may explain the observed worse performance on similar tasks like NER and dialogue state tracking [5, 6]. We also find that ChatGPT may be biased towards patterns in the limited examples. For example, with “tell me about morel mushrooms” labeled as “Definition word: morel mushroom” in a training example, ChatGPT

tends to label “Definition word” on other sentences like “tell me what’s happening.” All these examples illustrate the difficulties of in-context learning on a task with a complicated task goal.

In the evaluations conducted above, we direct the model to fulfill requirements of the dataset, which are intermediate tasks in a pipeline. The predictions are then utilized to run external functions (e.g., sending an email, answering a question, telling a joke) available in the specific annotated scenario using certain rules. However, ChatGPT has demonstrated its versatility as a chatbot that can communicate with users without being bound by these objectives. For instance, the model often appears to “understand” the commands in the “error” cases above, and may directly request for clarification when the input can not be understood, such as when an ASR error happens. When equipped with the ability to call external APIs [7, 8], New Bing can read returned results and reason upon them to determine the final response to users’ requests. Additional instructions regarding the functions can be also fed to the model via some system prompt in multiple rounds of conversation along with user inputs. In this way, the model’s ability on such intermediate tasks may underestimate its actual capability as an end-to-end assistant.

5. Conclusions

In this paper we evaluate large language models on SLU tasks with prompting. We find that they reach results comparable to fine-tuning on intent classification of multiple languages, but not on slot filling, and being sensitive to ASR errors. Future work will be focused on better prompting and incorporating phonetic knowledge for ASR-robust or end-to-end systems.

6. Acknowledgement

This work received funding under project SteADI, Swiss National Science Foundation grant 197479.

7. References

- [1] T. B. Brown *et al.*, “Language models are few-shot learners,” in *NeurIPS 2020*.
- [2] M. Chen *et al.*, “Evaluating large language models trained on code,” *arxiv*, vol. abs/2107.03374, 2021.
- [3] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” in *NeurIPS 2022*.
- [4] T. Kung *et al.*, “Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models,” *PLOS Digital Health*, vol. 2, no. 2, p. e0000198, 2023.
- [5] Y. Bang *et al.*, “A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity,” *arxiv*, vol. abs/2302.04023, 2023.
- [6] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, “Is ChatGPT a general-purpose natural language processing task solver?” *arxiv*, vol. abs/2302.06476, 2023.
- [7] T. Schick *et al.*, “Toolformer: Language models can teach themselves to use tools,” *arxiv*, vol. abs/2302.04761, 2023.
- [8] R. Nakano *et al.*, “WebGPT: Browser-assisted question-answering with human feedback,” *arxiv*, vol. abs/2112.09332, 2021.
- [9] R. de Mori, “Spoken language understanding: A survey,” in *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2007*, pp. 365–376.
- [10] L. Qin, T. Xie, W. Che, and T. Liu, “A survey on spoken language understanding: Recent advances and new frontiers,” in *IJCAI 2021*, pp. 4577–4584.
- [11] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, “Towards end-to-end spoken language understanding,” in *ICASSP 2018*. IEEE, pp. 5754–5758.
- [12] P. Haghani *et al.*, “From audio to semantics: Approaches to end-to-end spoken language understanding,” in *IEEE Spoken Language Technology Workshop, SLT 2018*, pp. 720–726.
- [13] M. Saxon, S. Choudhary, J. P. McKenna, and A. Mouchtaris, “End-to-end spoken language understanding for generalized voice assistants,” in *Interspeech 2021*. ISCA, pp. 4738–4742.
- [14] Y. Wang, A. Boumadane, and A. Heba, “A fine-tuned wav2vec 2.0/HuBERT benchmark for speech emotion recognition, speaker verification and spoken language understanding,” *arxiv*, vol. abs/2111.02735, 2021.
- [15] S. Arora *et al.*, “ESPnet-SLU: Advancing spoken language understanding through ESPnet,” in *ICASSP 2022*. IEEE, pp. 7167–7171.
- [16] S. Seo, D. Kwak, and B. Lee, “Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding,” in *ICASSP 2022*. IEEE, pp. 7152–7156.
- [17] M. Yazdani and J. Henderson, “A model of zero-shot learning of spoken language understanding,” in *EMNLP 2015*, pp. 244–249.
- [18] E. Ferreira, B. Jabaian, and F. Lefèvre, “Zero-shot semantic parser for spoken language understanding,” in *Interspeech 2015*. ISCA, pp. 1403–1407.
- [19] Y. Hou *et al.*, “Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network,” in *ACL 2020*, pp. 1381–1393.
- [20] T. Wu, R. Su, and B. Juang, “A label-aware BERT attention network for zero-shot multi-intent detection in spoken language understanding,” in *EMNLP 2021*, pp. 4884–4896.
- [21] B. Peng, C. Zhu, M. Zeng, and J. Gao, “Data augmentation for spoken language understanding via pretrained language models,” in *Interspeech 2021*. ISCA, pp. 1219–1223.
- [22] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Comput. Surv.*, vol. 55, no. 9, pp. 195:1–195:35, 2023.
- [23] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” in *EMNLP 2021*, pp. 3045–3059.
- [24] J. Kaplan *et al.*, “Scaling laws for neural language models,” *arxiv*, vol. abs/2001.08361, 2020.
- [25] K. Cobbe *et al.*, “Training verifiers to solve math word problems,” *arxiv*, vol. abs/2110.14168, 2021.
- [26] A. Chowdhery, S. Narang *et al.*, “PaLM: Scaling language modeling with pathways,” *arxiv*, vol. abs/2204.02311, 2022.
- [27] A. Srivastava *et al.*, “Beyond the imitation game: Quantifying and extrapolating the capabilities of language models,” *arxiv*, vol. abs/2206.04615, 2022.
- [28] J. Wei *et al.*, “Emergent abilities of large language models,” *Transactions on Machine Learning Research*, 2022.
- [29] J. Wei *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” in *NeurIPS 2022*.
- [30] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, “Complexity-based prompting for multi-step reasoning,” in *ICLR 2023*.
- [31] W. Yu *et al.*, “Generate rather than retrieve: Large language models are strong context generators,” in *ICLR 2023*.
- [32] C. Si *et al.*, “Prompting GPT-3 to be reliable,” in *ICLR 2023*.
- [33] Y. Wu, H. Wang, D. Zhang, G. Chen, and H. Zhang, “Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling,” in *COLING 2022*, pp. 7203–7208.
- [34] F. Song, L. Huang, and H. Wang, “A unified framework for multi-intent spoken language understanding with prompting,” *arxiv*, vol. abs/2210.03337, 2022.
- [35] K. Chang, W. Tseng, S. Li, and H. Lee, “An exploration of prompt tuning on generative spoken language model for speech processing tasks,” in *Interspeech 2022*. ISCA, pp. 5005–5009.
- [36] K. Chang *et al.*, “SpeechPrompt v2: Prompt tuning for speech classification tasks,” *arxiv*, vol. abs/2303.00733, 2023.
- [37] H. Gao, J. Ni, K. Qian, Y. Zhang, S. Chang, and M. Hasegawa-Johnson, “WavPrompt: Towards few-shot spoken language understanding with frozen language models,” in *Interspeech 2022*. ISCA, pp. 2738–2742.
- [38] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, “SLURP: A spoken language understanding resource package,” in *EMNLP 2020*, pp. 7252–7262.
- [39] D. Gerz *et al.*, “Multilingual and cross-lingual intent detection from spoken data,” in *EMNLP 2021*, pp. 7468–7475.
- [40] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” *arxiv*, vol. abs/2212.04356, 2022.
- [41] A. Radford *et al.*, “Language models are unsupervised multitask learners,” OpenAI, Tech. Rep., 2019.
- [42] S. Zhang *et al.*, “OPT: Open pre-trained transformer language models,” *arxiv*, vol. abs/2205.01068, 2022.
- [43] A. Conneau *et al.*, “XTREME-S: Evaluating cross-lingual speech representations,” in *Interspeech 2022*. ISCA, pp. 3248–3252.
- [44] S. Arora, S. Dalmia, X. Chang, B. Yan, A. W. Black, and S. Watanabe, “Two-pass low latency end-to-end spoken language understanding,” in *Interspeech 2022*. ISCA, pp. 3478–3482.
- [45] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, “Language-agnostic BERT sentence embedding,” in *ACL 2022*, pp. 878–891.
- [46] E. Villatoro-Tello *et al.*, “Effectiveness of text, acoustic, and lattice-based representations in spoken language understanding tasks,” *arxiv*, vol. abs/2212.08489, 2022.
- [47] I. McKenzie *et al.*, “The inverse scaling prize,” 2022. [Online]. Available: <https://github.com/inverse-scaling/prize>