Error Types in Transformer-Based Paraphrasing Models: A Taxonomy, Paraphrase Annotation Model and Dataset

Auday Berro, Boualem Benatallah, Yacine Gaci, Khalid Benabdeslem

Error types in Transformer-based Paraphrasing Models: A Taxonomy, Paraphrase Annotation Model and Dataset

Auday Berro[1][0000−0003−2411−5761], Boualem Benatallah[2][0000−0002−8805−1130], Yacine Gaci[1][0009−0001−8206−9559], and Khalid Benabdeslem[1][0000−0002−4324−924X]

[1] Université claude Bernard Lyon 1, LIRIS UMR 5205, Lyon, France - (auday.berro|khalid.benabdeslem|yacine.gaci)@univ-lyon1.fr
[2] Insight SFI research center on Data Analytics, Dublin City University, Ireland - boualem.benatallah@dcu.ie

Abstract. Developing task-oriented bots requires diverse sets of annotated user utterances to learn mappings between natural language utterances and user intents. Automated paraphrase generation offers a cost-effective and scalable approach for generating varied training samples by creating different versions of the same utterance. However, existing sequence-to-sequence models used in automated paraphrasing often suffer from errors, such as repetition and grammar. Identifying these errors, particularly in *transformer* architectures, has become a challenge. In this paper, we propose a taxonomy of errors encountered in *transformer*-based paraphrase generation models based on a comprehensive error analysis of *transformer*-generated paraphrases. Leveraging this taxonomy, we introduced the Transformer-based Paraphrasing Model Errors dataset, consisting of 5880 annotated paraphrases labeled with error types and explanations. Additionally, we developed a novel multilabel paraphrase annotation model by fine-tuning a BERT model for error annotation task. Evaluation against human annotations demonstrates significant agreement, with the model showing robust performance in predicting error labels, even for unseen paraphrases.

Keywords: Paraphrasing · Transformers · Annotation · Taxonomy

1 Introduction

Dialogue systems (*DS*), such as virtual assistants, and task-oriented bots, are emerging as a new frontier of human-computer interaction in natural language, receiving considerable recent attention [43]. These services communicate with users in natural language (e.g. text, speech, or both), performing a wide range of tasks such as reporting the weather, booking flights, or booking restaurants [45]. To satisfy user requests, a *DS* requires a large set of utterances paired with their corresponding executable forms (e.g. API calls). In particular, *task-oriented*

bots must first identify the user's intent from a given utterance [43]. For example, in "*List of restaurants serving Lebanese food in Lyon*", the bot must recognize the intent (i.e. *find_ restaurant*) and the associated slots (location="Lyon", cuisine="Lebanese"). Due to the powerful expressiveness of human language, the same intent can be formulated differently, e.g. "*Which restaurants in Lyon serve Lebanese food?*". Thus it is essential for bots to grasp the richness of human language, by training them on a linguistically diverse set of utterances for each intent [45, 24]. Failing to handle these variations in natural language can negatively impact the effectiveness of bots, and ultimately the user experience.

Paraphrasing is a key technique to build large and diverse utterances for the intents of interest [43]. Paraphrasing is an NLP task that aims to reformulate a given natural language utterance into its lexical and syntactical variations while meaning is preserved [8, 43]. It has numerous applications in NLP tasks, such as sentence simplification, text summarization, and Natural Language generation [8, 43]. *Paraphrasing* methods can be categorized into crowdsourced and automated approaches [45, 25]. Crowdsourced paraphrasing involves human workers generating multiple paraphrases based on a seed utterance [45]. In *automatic paraphrasing* (AP), paraphrases are systematically generated [24, 49]. The literature on AP explored template-based, rule-based, and statistical machine translation approaches [25, 26]. Recent attention has shifted to neural network models [31, 26], particularly the *transformer* architecture [41], acknowledged for its state-of-the-art performance in various NLP tasks and widely adopted as the preferred sequence-to-sequence architecture for paraphrasing[7]. However, despite their success, seq2seq models frequently introduce errors such as repetition, grammatical inaccuracies, and incoherent text [39]. Ongoing efforts have concentrated on detecting and identifying paraphrasing errors in automatic neural models to enhance their robustness [35]. Meanwhile, the increasing complexity of *transformer*-based models complicates error identification, making it harder to distinguish between machine- and human-generated text [1, 11]. As these models evolve, the human ability to manually discern and tag machine-paraphrased text diminishes, especially with holistic alterations in sentence structure and word order instead of single-word replacements. Recognizing the pivotal role of errors as indicators for system improvement [40], the evaluation of errors in generated paraphrases becomes paramount.

This study focuses on quality control for paraphrasing, particularly the evaluation of paraphrase errors in *transformer*-based paraphrase generation (TPG) models. While quality control for text generated by NLG systems has been explored in a wide range of tasks, like machine translation (MT) [13], question generation [37], and open-ended generation with pre-trained language models [11], TPG has not been subjected to such scrutiny. To the best of our knowledge, this study is the first to categorize paraphrasing errors in TPG models. Note that our identified errors may not be exhaustive but rather serve as an initial pool of errors for further study and investigation. Efficient error identification and categorization in AP models yield multifaceted advantages, mainly contributing to the elevation of paraphrased content quality. A comprehensive grasp of prevalent

errors empowers researchers and developers to strategically augment the performance and reliability of their paraphrasing models. Also, categorizing errors not only sheds light onto the strengths and weaknesses of paraphrasing models, but also establishes strong benchmarks to facilitate fair comparisons between different models. Moreover, our proposed error categories help in enriching training datasets. This, in turn, allows models to navigate and handle common errors, making them more robust and production-ready for real-world applications. The contributions of this study can be summarized as follows:

1. We selected five *transformer*-based paraphrasing models to generate 22K paraphrases for 598 seed utterances extracted from a dataset of crowdsourced queries across two intents.
2. We synthesized the literature on paraphrasing quality control in three distinct areas: errors in crowdsourced paraphrasing, inconsistencies in crowdsourced slot annotations, and errors in the generation of pre-trained language models. We used this synthesis as a starting point for building our own taxonomy. We then extended this taxonomy through several rounds of qualitative evaluations of the generated paraphrases. Consequently, we identified a taxonomy of 15 error types in *TPG* models.
3. We used the proposed error taxonomy to annotate the generated paraphrases. Accordingly, we constructed an annotated dataset called *TPME*, in which the paraphrases were labeled with a range of different categorized errors.
4. We developed a multi-label paraphrase annotation model using the TPME dataset. The annotation model uses a fine-tuned BERT model to predict error types in paraphrases, enabling the automatic annotation of multiple errors in a paraphrase.
5. We released TPME dataset, code for generating paraphrases, fine-tuned BERT model, and information required to reproduce our study [3].

2 Related Work

Characterizing error has been done in many areas, such as MT [13], crowdsourced paraphrasing [45, 23], NLG systems [39, 11]. Overall, research from these efforts is certainly complementary and some elements are indeed adopted in our work.

Paraphrase generation (PG). Crowdsourced **PG** have been investigated to obtain training datasets for *DS* [21, 28, 44, 43, 33, 36]. In *crowdsourced* **PG**, an initial utterance, usually provided by an expert, is presented as a starting point, and crowdworkers are then recruited to obtain further paraphrases [43]. For instance, Chklovski et al. [9] used crowdsourcing to collect paraphrases using gamification. Contributors were asked to generate paraphrases based on given hints (e.g. words suggestions). Other crowdsourcing strategies were proposed [46, 47]: (i) *Sentence-based strategy*: Workers were tasked with paraphrasing a given

[3] https://github.com/AudayBerro/TPME/tree/master

sentence into new variations. (ii) *Goal-based strategy:* Workers were provided with a task goal (e.g., "book a restaurant") and a set of possible entity values (e.g., "cuisine: Indian, city: Paris") to produce paraphrases. (iii) *Scenario-based strategy:* This approach employs a storytelling framework that provides a scenario to workers and ask them to generate paraphrases accordingly (e.g., "Your goal is to book a restaurant; you are in Paris; you are hungry and want to eat Indian dishes").

Automated **PG** does not involve humans in the process and refers to a task in which a system generates paraphrases given an input sentence [24]. The literature on *automated* **PG** covers a wide range of approaches, including probabilistic, handwritten rules, and formal grammar models [15]; data-driven techniques [25, 27]; machine translation techniques [18]. However, these approaches struggle to capture the nuanced complexities of natural languages in contextual settings [14]. In addition, the manual design of rules is complex for practical implementation [48]. Consequently, neural-based and deep learning models have gained popularity for **PG** [49, 31], offering a solution free of previous limitations. However, a critical problem persists: generated paraphrases often fail to align with user preferences and produce uncontrolled results [48]. Although syntactically controlled paraphrasing **PG** [20, 16] offers a promising approach that incorporates syntactic templates, it requires users to possess linguistic expertise and define specific syntactic structures, which is challenging in practical applications. With the recent advances enabled by large language models (e.g. GPT), there is a shift towards their use to generate paraphrases [42, 17, 5].

Errors in crowdsourced paraphrasing. Crowdsourced paraphrases often contain errors, including misspellings, grammatical mistakes, and missing slot values [45]. Two approaches are commonly used to evaluate paraphrase quality [45, 29]. In *Pre-hoc*, paraphrases are evaluated during the crowdsourcing task before submission [29]. In *Post-hoc*, they are assessed after task completion [45]. Yaghoubzadeh et al.[45] employed a *post-hoc* method to investigate crowdsourced paraphrasing errors in task-oriented bots. They identified a taxonomy of six error types (misspelling, linguistic, cheating, answering, semantic, translation) and they developed the *Para-Quality* dataset based on these findings. Similarly, Larson et al. [23] identified different types of incorrect annotations of crowdsourced paraphrases. They identified a taxonomy of six types of inconsistencies in slot-filling annotations (e.g. slot format, omission, wrong label, slot addition).

Errors in MT systems. Significant work on errors has been reported for MT systems. Koponen et al. [22] investigated error classification with an emphasis on semantic accuracy. The error analysis was performed on human translations as well as on the outputs of 2 different types of MT systems: *rule-based* MT and *statistical* MT. They identified 13 errors grouped into 2 categories: concept (5) and relation (8) errors. Popovic et al. [30] investigated the nature and causes of MT errors observed by different evaluators on different quality criteria: adequacy, comprehension, and fluency. They identified 26 errors (e.g. omission, gender) and reported the results for 3 language pairs, 2 domains, and 11 MT systems.

3 Paraphrase Generation

Paraphrase generation poses unique challenges compared with other text generation tasks because of the requirement to produce sentences that convey the same meaning with different words or structures. This requires creativity and linguistic versatility. In addition, paraphrasing models must retain the mentions of intents and their respective slots, adding to the complexity. Sparse data collection for paraphrases further compounded this challenge, limiting exposure to diverse scenarios. Despite advancements, even transformer-based models exhibit errors, such as incorrect substitutions, missing words, awkward structures, or alterations in meaning. To address this, we developed a taxonomy to systematically categorize these errors, resulting in a TPME dataset with manually labeled paraphrases and errors. Furthermore, we fine-tuned a BERT-based model for automated error-detection. To collect paraphrases, we followed a methodical approach: (i) obtain seed utterances, (ii) select paraphrasing models, and (iii) generate paraphrases using these models. Each step is described in detail below:

3.1 Selection of Seed Utterances

In this study, we employ the SNIPS dataset [4], consisting of crowdsourced queries categorized into seven user intents. Each utterance in the dataset is paired with a list of required slots, which are specific pieces of information or textual parameters within an utterance that need to be identified and extracted. For each SNIPS excerpt, we extract the utterance (e.g. "*how cold is it in Princeton Junction*") and its list of required slots (e.g. *condition_temperature*=*"cold"* and *city*=*"Princeton Junction"*). To manage the manual labeling effort, we focused on two key intents: *GetWeather* and *BookRestaurant*, which enabled us to collect 598 seed utterances. **GetWeather** encompasses requests for weather forecasts comprising 9 slots (e.g. country, city, temperature). **BookRestaurant** includes queries relating to restaurant reservations, with 14 slots (e.g. city, time, dishes served).

3.2 Selection of Models

We used the following criteria to select the paraphrasing models used in this study: (i) models must fall under the category of text generation and can produce paraphrases in English; (ii) they should be built upon the transformer architecture [41] in any of its variations, such as decoder-only or encoder-decoder; (iii) The official checkpoints of the models must be publicly and freely accessible through platforms or web links provided by their authors. This ensures the avoidance of potential biases that might arise if we were to implement, train, or fine-tune the models. In this study, we chose the following five (TPG) models:

PROTAUGMENT [10]: fine-tuned a BART pre-trained transformer-based language model to generate paraphrases.

[4] https://github.com/sonos/nlu-benchmark

Fine-tuned T5 [3]: the authors fine-tuned T5 [32], a pre-trained transformer-based language model to generate paraphrases.

NL_Augmenter [5]: is a data-augmentation platform that supports various transformations. We selected the NL_Augmenter *Diverse Paraphrase Generation* transformation for this study, which generates paraphrases by leveraging a transformer through pivot-translation [2].

PRISM: Although PRISM [38] is a quality estimation model designed to evaluate the performance of MT systems, it includes an automatic paraphrase generation component. The authors trained a transformer-based MT model with approximately 745 million parameters to perform zero-shot paraphrasing in 39 languages. PRISM can be used in paraphrases generation.

GPT [4]: GPT is a generative transformer-based language model with outstanding performance. Recent GPT models can adapt to new, possibly unseen, tasks using In-context Learning through natural language instructions and input. This opens up the possibility of improving the paraphrasing process [6]. We leveraged *GPT-3.5-turbo* with tailored prompts for paraphrase generation [6].

3.3 Generation of Paraphrases from Utterances

Leveraging the aforementioned models, we systematically generated paraphrases for each seed utterance, yielding a comprehensive collection of 22864 paraphrases. We categorize the generated paraphrases in ten distinct datasets, based on the paraphrasing model used for their generation, and the underlying intent of the seed utterances. For instance, the ***T5-GW*** dataset (accessible via the provided additional materials link) consolidates all paraphrases generated by the T5 model in response to seed utterances associated with the *GetWeather* intent.

Each grouped dataset includes the following columns: *utterance*, *paraphrase*, *list_of_slots*, and *intent*. The *utterance* column corresponds to the seed sentence designated for paraphrasing, while the *list_of_slots* column encompasses any identified slots in the *utterance*, along with their corresponding values. The *paraphrase* column houses the resulting paraphrase. Finally, the *intent* column indicates the intent of the utterance, with only two possible values: *GetWeather* or *BookRestaurant* in this work.

4 Paraphrasing Error Types

We identified 15 error types in the paraphrases categorized into three error classes through a meticulous evaluation process conducted over multiple rounds. Initially, we randomly selected ten utterances with their corresponding paraphrases, totaling 700 paraphrases for evaluation. We labeled using error types identified by previous research in crowdsourced paraphrasing [45], crowdsourced slot annotation inconsistencies [23], and errors in text generated by pre-trained

[5] https://github.com/GEM-benchmark/NL-Augmenter

[6] The prompt we used can be found in the supplementary material link supplied.

Table 1. Samples of erroneously generated paraphrases. The colors in the *Paraphrase* column denote the locations of errors.

#	Utterance	Paraphrase	Error Label
1	Is it chillier here	Is it Colder Here ?	Spelling
2	book a party of 4 for a cafe rated best	booking a 4 party for a best rated cafe	Spelling
3	What is the weather forecast for Manassa?	What's the weather prognosis forManassa ?	Spelling
4	is it cold in Vigus TN	if it's cold in ~~the~~ Vigus TN	Grammar
5	What's the weather in Poland?	What is the time in Poland?	Semantic
6	Book spot in Fults in Federated States Of Micronesia	Book spot at Fults in the Federated Federal States of Micronesia.	Redundant
7	Book me a lunch at Boreas for a party of 8	Book me a lunch at Boreas for a party of 8!	Duplication
8	book a table in Yachats for 3 people	Procure iservation sayrettel batou hcastuohcaeY n selaecinimanyThgiretepotsereahtroforretaB	Incoherent
9	Will it be cold in Haiti?	Will There Be Cold in Haiti ?.	Punctuation
10	Tell me the weather forecast for France	Tell me the weather forecast for France for the weekend .	Slot addition
11	Book a spot for now in NY for 10	Book a place in NY for 10	Slot omission
12	What is the weather forecast here	What is the forecast for the weather in the future ?	Wrong Slot
13	Tell me if it will be foggy in Maxwell	Inform me, should ~~the location known as~~ 'Maxwell' suffer from ~~an impediment such as haze/mist/~~ fog ~~I'd really appreciate knowing about such trivia...~~	Wordy
14	Will there be wind in Hornitos DC?	Will there be wind in Hornito's DC?	Homonym
15	Will it be temperate in West Liberty MH?	Will it'll be temperate in West Freedom MH?	Homonym
16	Will the weather be good at breakfast in Bay Harbor Islands	Is the weather good for breakfast in the Bay Puerto Islands?	Homonym
17	is Saint Robert hotter than Turkmenistan?	Saint Robert is hotter than Turkmenistan.	Answering
18	book spot for 4 in Somalia	Book spot for 4 in Somalia is available for free .	Answering
19	Is it hot in the current location?	Is it hot in the current location? If so, why?	Questioning
20	book a table in CA for 2 people in 3 hours	booking a table in Central Asia for 2 people in 3 hours	Acronym

language models [11]. The initial list included *semantic*, *redundant*, *spelling*, *grammar*, *slot addition*, *incoherent*, and *duplication* errors. Through manual annotation and iterative refinement, involving random sampling and evaluation of paraphrases based on the evolving list of error types, similar errors were grouped into new types, resulting in the final taxonomy. We also refined error definitions, such as introducing the concept of "*near-copy*" (refer to § 4.1) for *duplication* errors. In the following, we proceed to describe our three error classes, highlighting the specific error types within each class.

4.1 Language Errors

Language Errors encompass a range of inaccuracies in paraphrased content, including spelling, grammar, syntax, and semantic inconsistencies. Seven types of *language errors* were identified.

Spelling refers to the correct arrangement of letters to form a word. Misspelling is one of the most common mistakes in crowdsourced paraphrasing [45]. In our evaluation, we also count as misspelling capitalization errors (sample 1 in Table 1), missing hyphens (sample 2) and missing spaces (sample 3).

Grammar These errors relate to the incorrect use of verbs, prepositions, singular/plural nouns, articles, and other grammatical elements [45]. In sample 4, the paraphrase exhibits a misuse of the article "*the*" before city names.

Semantic This error, further characterized as a "semantic deviation", arises when a paraphrase deviates from the intended meaning of a seed utterance. For instance, in sample 5, the paraphrase asks for time instead of weather conditions.

Redundant Redundancy arises when a word or phrase is duplicated in a paraphrase, either through exact repetition or the use of different words conveying the same context. In sample 6, the term *Federal* redundantly duplicates the meaning conveyed by the term *Federated*.

Duplication Duplication arises when the generated paraphrase either mirrors or closely resembles the utterance. We employ the term "*near-copy*" to describe instances where the paraphrase closely mirrors the utterance. "*Near-copy*" occurs when a paraphrase differs from the utterance solely in terms of punctuation (e.g (e.g., commas, periods, question marks, colons, etc) and capitalization. However, the "*near-copy*" condition is violated if the paraphrase contains at least one token that differs from the utterance. This is illustrated in sample 7.

Incoherent As in sample 8, we label a paraphrase as incoherent when the generated text is confusing, hard to understand, or appears nonsensical.

Punctuation A punctuation error occurs due to the overuse or inappropriate placement of punctuation marks in the paraphrase. This includes inserting question or exclamation marks in sentences without corresponding questions or exclamations. It also encompasses the misuse of currency, non-alphabetic or numeric symbols (, _, #, &, etc). For instance, sample 9 displays a punctuation error where a period follows a question mark incorrectly.

4.2 Slot Errors

These errors involve incorrect actions at the slot level, such as adding, removing, or altering slots.

Slot addition Slot addition occurs when the model inserts at least one additional slot value into a paraphrase. Consider sample 10 in Table 1. In this sample, token "the weekend" which is the value of the timeRange slot in the paraphrase, is an additional slot. It's important to note that for slots that accept multiple values, this is not considered an error. For example, the party_size_description slot in the *BookRestaurant* intent can have multiple values. In the utterance "*Book a table for Ali, Jo, and Max*" the tokens "Ali", "Jo" and "Max" form a single multi-token value, and the party_size_description slot treats them collectively as a single value.

Slot Omission Slot omission occurs when a slot, expected to be referenced, is overlooked in the paraphrase. Illustrated in sample 11 of Table 1, the paraphrase fails to include a value for the timeRange slot, even though it is explicitly mentioned as "now" in the original utterance.

Wrong Slot A wrong slot occurs when the value of a slot in the paraphrase deviates from the expected slot and is replaced by an non-matching token. In sample 12, rather than inquiring about the forecast for the present location, the paraphrase erroneously requests a weather forecast for a specific time period.

4.3 Errors of human characteristics

We identified 5 types of *human-characteristic errors*, which uniquely mimic human behavior. When these errors occur, the *transformer* behaves as if it were human, such as responding directly to a request instead of paraphrasing it.

Wordy Wordy errors occur when the generated text contains excessive wording or unnecessary information, leading to verbose paraphrasing. See sample 13.

Homonym Homonyms are words that share the same pronunciation but have different meanings or spellings. An error arises when a token in the paraphrase shares a similar or identical pronunciation with a token in the utterance. In sample 14, "Hornito's" and "Hornitos" sound alike, leading to incorrect use of the possessive apostrophe ("'s") in the paraphrase. This category includes cases which tokens are replaced with synonyms or translated, potentially altering the paraphrase's meaning (see samples 15 and 16 resp.).

Answering This error occurs when the paraphrased content responds to the utterance, causing the model to generate an answer instead of a paraphrase. In sample 17, the model transforms the entire paraphrase into an answer, while in sample 18, the answer is added to the query. To differentiate this from the Wordy Error, we label sentences as Answering Error if the additional tokens answer a query or question from the initial utterance.

Questioning Arises when the paraphrased text introduces an extra question not present in the utterance. In sample 19, the addition of "*If so, why?*" exemplifies this error by introducing an extra question.

Acronym An acronym is a word or name formed from the initial letters of a longer phrase. Acronym error occurs when a paraphrase improperly uses an acronym or includes an incorrect expansion of an acronym from the utterance. In sample 20, "Central Asia" is an inaccurate expansion of the acronym "**CA**" which actually represents California, a value for the "state" slot.

5 Creation of Annotated Paraphrasing Error Dataset

This section presents an overview of TPME, and gives insights and analyses of the paraphrasing errors.

5.1 The TPME Dataset

We annotated a representative subset of generated paraphrases, covering at least 22% of the entire set, resulting in 5880 annotated paraphrases. Each paraphrase

was labeled with one or more error types from our taxonomy (as detailed in §4) and was accompanied by an explanation in plain English. For "*slot errors*" (see §4.2), we compared each sampled paraphrase with its corresponding list of required slots listed in the "*list_ of_ slots*" column. The *TPME* dataset includes the following columns: "*utterance*" column which contains the seed sentence to be paraphrased. "*list_ of_ slots*" includes any slot present in the utterance along with its corresponding value. "*paraphrase*" contains a generated paraphrase. "*models*" denotes the model that generated the paraphrase. "*error_ category*" contains labels of the errors found in the paraphrase, and their justification in natural language is described in the column "*explanation*". "*Intent*" indicates the intention conveyed by the utterance.

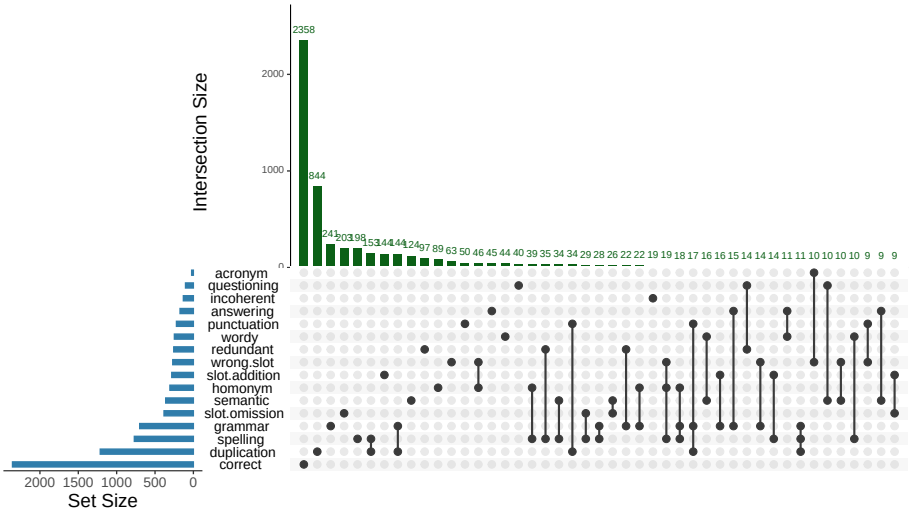

Fig. 1. TPME dataset label statistics.

5.2 Insights into Error Frequency and Co-Occurrences

Figure 1 visually presents the distribution, frequencies and co-occurences of labels within the *TPME* dataset, employing an UpSet plot through Intervene platform [7]. Notably, only 40.1% of the paraphrases were labeled as *correct*, underscoring the prevalence of paraphrasing errors, and their negative impact in the context of developing *DS*. Specifically, only 3.5% were exclusively labeled with *semantic* errors, without any additional labels. In addition, 6.8% of the paraphrases were identified with *grammar* errors only. The plot also demonstrates the frequency of the co-occurrence of two or more labels. For instance,

[7] https://upset.app/ and https://asntech.shinyapps.io/intervene/

all paraphrases labeled as *slot omission* (29 occurrences) are also labeled as *spelling*. Additionally, 34 paraphrases shared both *semantic* and *spelling* error labels, constituting 0.9% of the erroneous paraphrases. Furthermore, 18 paraphrases were labeled with *homonym*, *spelling*, and *grammar* errors. Moreover, 144 paraphrases were concurrently labeled with *duplication* and *grammar*.

To analyze the distribution of errors across the selected models, individual Upset Plots were generated for each of the five models. The initial observation highlights that out of 1092 instances, GPT yielded 659 correct paraphrases (60.3%). In contrast, **T5** achieved a correctness rate of only 25.1%, and **PROTAUGMENT** demonstrated an even lower rate of 17.64%. The second significant finding is that none of the paraphrases generated by GPT in TPME were labeled as *duplication*, distinguishing it from other models. For example, **PROTAUGMENT** had a *duplication* error rate of 51.04%, and **T5** exhibited a rate of 21.5% for these errors. In summary, GPT displays resilience against duplication compared with other *TPG* models. However, duplication may occur among the generated paraphrases. For example, in a list of 10 paraphrases, we may have three paraphrases that are replicated.

5.3 Analysis of the annotated paraphrases

The relevance of capitalization in paraphrases. Understanding user utterances relies on entity extraction, known as slot-filling, which aims to identify the values of different slots in a user utterance [19]. For instance, when a user requests *nearby restaurants* the values of the *location* and *cuisine* slots are essential for a bot to retrieve the appropriate information. We observed challenges with capitalization in slot-filling tasks, particularly when using *case-sensitive* slot-filling models. Consider the utterance *"book smoking room in OR at a bar"*[8] and its paraphrase *"Book Smoking Room In OR at bar OR at hotel"*. In the paraphrase, the second "OR" token serves as a conjunction indicating a choice between a *bar* ***or*** a *hotel* but it is erroneously written in uppercase. This capitalization introduces a spelling error and leads to a redundant error as the paraphrase already includes the token "*OR*". Additionally, the capitalization issue may result in a slot addition error, where the model misinterprets "*OR*" as an abbreviation for the state of "*Oregon*", introducing ambiguity. To address these issues, we propose accurately representing capitalization in paraphrases to reflect both the input and output of the transformer architecture.

The propagation of the source utterance errors in the paraphrase. *Transformer*-based language models are highly effective in learning language properties [12], yet instances of errors persisting in generated paraphrases have been identified. For example, in the utterance "*book spot at Candle Cafe*" and its paraphrase "*Book spot at Candle Cafe*", the *transformer* omitted the indefinite article "*a*" before "*spot*", leading to a grammar error. We attribute such errors to

[8] OR refers to the state of Oregon

the inherent nature of *transformer* architecture and the paraphrase generation task [9]. Transformers rely on the *self-attention* mechanism, meaning errors in the input may receive more attention and propagate through subsequent layers. However, transformers also demonstrate the ability to rectify errors in the input during paraphrase generation. For instance, the input utterance "*is it going ot be chillier in Maumee*" contains "*ot*" instead of "*to*", but the model corrected this error in the paraphrase, resulting in "*is it going to be colder in Maumee?*".

The insertion of the determiner "*the*" in front of geographical names. In some paraphrases, we encountered errors that violated basic grammar rules. For instance, consider the utterance "*weather in Hillsview MA*" and its paraphrase "*Weather in **the** Hillsview MA*". The model mistakenly inserted the determiner **the** before the token "*Hillsview*", resulting in a grammar error. Notably, when dealing with geographical names such as the city name "*Hillsview*", the definite article "**the**" is not used in English, making this a common grammatical inconsistency that we observed in this study. Another prevalent error involves the insertion of the possessive form ('s) in phrases like "*Will there be wind in Hornito's DC?*" where the model generated "Hornito's" instead of the correct "Hornitos" as found in the utterance "*Will there be wind in Hornitos DC?*".

Errors may be context- and domain-dependent. For the utterance "*book a table for me, heidi and cara in Saudi Arabia*" and its paraphrase "*Book a table for me, Heidi and Vedi in Saudi Arabia"*, in the paraphrase the token "*Vedi*" is an appropriate value for the party_size_description slot. The transformer paid more attention to the previous "*Heidi*" token, which resulted in the generation of the "*Vedi*" token in the paraphrase. However if we pay more attention, the two names "*Heidi*" and "*Vedi*" have close pronunciation which leads to a homonym error. This error, more than a minor pronunciation anomaly, can significantly impact the performance of the *DS* trained with such paraphrases. Consider a bot in the banking sector. Executing a money transfer to "*Heidi and Vedi*" instead of "*Heidi and Cara*", as stated in the utterance, would be incorrect. While this variation enhances lexical diversity, it can also negatively impact critical domains. In addition, homonym errors may lead to wrong slot errors. In the paraphrase "*Will there be wind in Hornito's DC?*" the token "*Hornito's*" is a homonym of the token "*Hornitos*" in the utterance. The addition of the possessive "s" introduces a homonym error, resulting in the city slot having the value "*Hornito's*", which does not match the correct "*Hornitos*" for the city slot. Thus, a wrong slot error emerges. Consequently, the tolerance for such errors, varies based on the context, domain, and associated slots when the error affects the slot value level.

Errors in GPT generated paraphrases. For the utterance "Will it be windy at 4 Pm in NY?", GPT generated the paraphrase "Are we expecting any strong winds by 4 PM in New York City?". GPT incorrectly generated the value "*New York City*" which represents a value for the "*city*" slot instead of the intended

[9] Paraphrase generation is a multi-step (word-by-word) prediction task, where a small error at an early time-step may lead to poor predictions for the rest of the sentence, as the error is compounded over the next token predictions [8]

"*state*" slot, mentioned as "*NY*" in the utterance. Similarly, for the utterance "book a turkish restaurant in DE", GPT generated the paraphrase "Reserve a Turkish restaurant in Germany", incorectly inserting "*Germany*" instead of the correct value "*Delaware*" (abbreviated as "*DE*" in the utterance), where GPT considered "*DE*" to be the acronym for "*Deutschland*". However, this introduces a wrong slot error, as the correct slot is "state", not "country". Across all GPT-generated paraphrases, 10.3% were labeled as wrong slot errors. For "weather close-by Lone Elk County Park at 4 am", GPT actually answered the weather-forecast query: "By dawn, you can anticipate interesting conditions nearsby at Lone Elk county park". The answering error occurred in 5.1% of GPT-generated paraphrases. Additionally, 9.4% of GPT paraphrases were marked with semantic errors. For example, in "What will the weather be at six o'clock in the Virgin Islands?" GPT generated only a question mark "?" as a paraphrase. For "book a table in Yachats for 3 people" GPT generated extensive gibberish tokens with numerous misspelled words, such as "iservation" which should be "a reservation". The paraphrase is incoherent, making it hard to understand. Also in "Ensure there is seating available in Yachtseservt yeNameYeepsaYehT-fruintap eneeaAredtonOSsegnosrepednIhseltiuqeobsuocotohibm ateletibasenaY-hctayssesacllaeviser" GPT generated an arbitrary content.

For the utterance "*I want to book a highly rated restaurant for Sue, Madeline, and me in eight years*", GPT generated "*<introductory statement> I desire very much that we find <quality adjective>restaurant well-known far&wide ..-for our select groupIO*" and "*Imagine the celebration of such a beautiful day<including excitement) <Pronoun> has made reservations at the highest-ranked restaurants*". Apart from incoherent, spelling, and grammar errors, we observed the insertion of tagged tokens like "<Pronoun>" and "<introductory statement>". Instead of generating slots values, GPT generated canonical tokens to indicate the need to insert values at those positions.

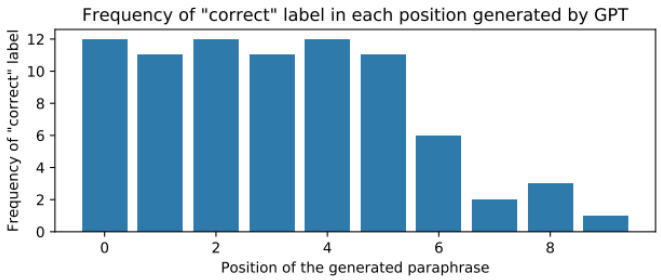

Fig. 2. Frequency Analysis of Correct Labels in 130 GPT Paraphrases across Positions.

Another GPT-specific finding is the removal of certain paraphrases due to toxic content, aligning with OpenAI's policy. For instance, when prompted with "need a table at a close-by restaurant right now in Marco" GPT generated

"***Paraphrase removed due to inappropriate content***" illustrating paraphrase suppression due to toxicity. GPT produced three times the same paraphrase for this utterance. Because the model is exclusively accessible via API calls with undisclosed details, pinpointing the exact content that triggers filtering is impractical. Given its exclusivity to GPT, we opted not to categorize it as a distinct error in our taxonomy but rather to highlight its occurrence. Furthermore, we evaluated the frequency of "*correct*" labels in GPT-generated paraphrases for each seed utterance. We randomly selected 13 utterances and extracted their corresponding sets of 10 paraphrases, totaling 130 paraphrases. The key observation is that errors consistently appear in the paraphrases generated towards the end. As we progressed through the list of 10 paraphrases for each seed utterance, errors became more prevalent, with the majority occurring after the sixth position. Figure 2 illustrates this trend, emphasizing the concentration of errors towards the later positions in the paraphrase lists.

6 BERT-based Multi-label Paraphrase Annotation Model

In this section, we explore using the TPME dataset to fine-tune a BERT model for multi-label paraphrase annotation. While deep learning models like BERT have shown impressive performance across various NLP tasks, including sentence classification [34], annotating paraphrase errors with their respective error types requires a significant amount of labeled data, posing a challenge. Thus, we fine-tuned BERT on TPME to develop a multi-label paraphrase error annotation model aimed at predicting error types in paraphrases.

6.1 BERT Fine-tuning

The fine-tuned BERT model (FBM) is a multi-label prediction model that takes as input a pair of an input utterance u and its paraphrase p and predicts one or more error labels. At the fine-tuning time, u and its p are provided as inputs to the BERT model and tokenized through the BERT tokenizer into one sequence (BERT input: $< u >< sep >< p >$). The "$< sep >$" token acts as a separator between u and p. When the input text is tokenized, BERT interprets the segments as distinct parts of the input sequence. "$< sep >$" helps BERT to understand the structure of the input and learn contextualized representations for each segment. For fine-tuning, the TPME dataset was split into training (80% $\equiv$ 4704 paraphrases) and validation (20% $\equiv$ 1176 paraphrases) datasets. We fine-tuned a *bert-base-uncased* model. BERT logits (i.e. output) have the form (batch size, number of labels) and represent the non-normalized scoring for each label. To convert these logits into predicted labels, we added a linear layer on top of BERT. Thus, applying a sigmoïd function to each logit independently scales the values between 0 and 1, treating them as "probabilities" for label presence. These probabilities are then classified using a standard threshold, usually set at 0.5. If the probability exceeds the threshold, the label is predicted for p; otherwise, it is not predicted.

Table 2. Prediction performance of FBM in terms of Krippendorff's alpha (values range from 0 to 1, where 0 is perfect disagreement and 1 is perfect agreement), Exact Match Ratio, Hamming loss (smaller the value, better the performance). F1, precision and recall are samples-averaged.

	Krippendorff	Exact Match Ratio	Hamming Loss	Recall	Precision	F1
FBMvsVal	0.549	0.693	0.035	0.753	0.766	0.753
FBMvsGold	0.809	0.699	0.027	0.807	0.754	0.770

6.2 Evaluation

This section presents the experimental results of the annotation of two paraphrase datasets using the FBM. First, FBM was applied to a benchmark dataset, called FBMvsGold, comprising a 20% subset of the TPME dataset, including utterances, corresponding paraphrases, and error labels. FBM predicts error labels for each utterance-paraphrase pair to assess the annotation quality of familiar data seen during the fine-tuning stage. Second, FBM annotation was evaluated on dataset $\mathcal{B}$, comprising 1000 pairs of utterances and paraphrases randomly selected from the 22K unannotated paraphrases. Dataset $\mathcal{B}$ is defined as $\mathcal{B} = \{p \mid p \, \epsilon \, (\mathcal{P} - TPME)\}$, where $\mathcal{P}$ denotes the dataset of 22k automatically generated paraphrases and TPME denotes the dataset of 5880 annotated paraphrases. This evaluation assesses FBM annotation on unseen data. After predicting the error labels for Dataset $\mathcal{B}$, 100 rows were randomly selected for manual annotation, resulting in the FBMvsVal dataset. FBMvsVal enables the assessment of FBM annotation on unseen data. Finally, FBM annotation was evaluated against human annotation in FBMvsGold and FBMvsVal using established multilabel evaluation metrics from the literature, including Krippendorff's alpha, Exact Match Ratio, Hamming Loss, recall, precision, and F1 metrics.

Analysis of results: The Krippendorff metric revealed strong agreement between FBM and human annotations in both FBMvsGold and FBMvsVal datasets, with Krippendorff's alpha scores of 54% and 80%, respectively (Table 2). Scores exceeding 50% indicated good agreement, suggesting a robust correlation between the model predictions and human annotation. However, there was a notable disparity between the datasets, with FBMvsVal scoring 54% and FBMvsGold scoring 80%. This variation may stem from the uneven distribution of errors in the TPME dataset used to fine-tune BERT. For instance, the incoherent label is applied to only 19 items, compared to 241 and 844 items labeled as grammar and duplication respectively. Consequently, accurately predicting errors becomes more challenging. Future work will involve augmenting the TPME dataset by annotating more paraphrases while ensuring a balanced representation across error types. Regarding Hamming Loss, which indicates misclassification frequency, both FBMvsVal and FBMvsGold datasets exhibited remarkable performance, with scores of 3.5% and 2.7%, respectively. (Further details on additional measures are omitted due to space constraints.)

7 Conclusion and Future Work

In this study, we used a data-driven approach to investigate and quantitatively identify errors in *TPG* models. Identifying the nature and frequency of these errors is important for enhancing *DS* and improving *TPG* models performance. We first discussed and outlined the importance of paraphrasing in the acquisition of training data for *DS* development. Subsequently, we emphasized the importance of error evaluation in the paraphrases generated by transformer-based models. Through empirical analysis, we identified a taxonomy of 15 error types, which we used to annotate a paraphrasing dataset with associated errors. Our analysis revealed that despite the success of transformers, paraphrase generation remains error-prone, with only 40.1% of paraphrases being correct. Finally, we released the dataset of paraphrases and errors to the research community.

In our future work, we plan to expand the TPME dataset by annotating additional paraphrases to achieve balance across different error labels. Moreover, we aim to enhance the proposed error taxonomy by exploring various transformer architecture variants, including encoder-only, decoder-only, and encoder-decoder models. The TPME dataset can serves as a valuable training data for diverse tasks. In addition to error detection, we also plan to investigate to error correction using fine-tuning language models. While the TPME dataset is relatively small, its manual annotation process was thorough yet time-consuming and costly. BERT was selected for its effectiveness in multi label error annotation, demonstrating the utility of our dataset. In future research, we will focus on exploring other newer models (e.g., GPT, mistral-7b, LLaMA) through a comparative study, aiming to validate and extend the applicability of our dataset in multi-label error annotation tasks.

Acknowledgments. We acknowledge the financial support provided by the PICASSO Idex Lyon scholarship, which supported the research conducted by Auday Berro as part of Ph.D. studies.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alikaniotis, D., Raheja, V.: The unreasonable effectiveness of transformer language models in grammatical error correction. BEA@ACL (2019)
2. Bannard, C., Callison, C.: Paraphrasing with bilingual parallel corpora. ACL'05 pp. 597–604 (2005), https://aclanthology.org/P05-1074
3. Berro, A., Fard, M.A.Y.Z., et al.: An extensible and reusable pipeline for automated utterance paraphrases. PVLDB (2021)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., et al.: Language models are few-shot learners. NeurIPS (2020)
5. Bui, T.C., Le, V.D., To, H.T., Cha, S.K.: Generative pre-training for paraphrase generation by representing and predicting spans in exemplars. In: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp). pp. 83–90. IEEE (2021)

6. Cegin, J., Simko, J., Brusilovsky, P.: Chatgpt to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. arXiv preprint arXiv:2305.12947 (2023)
7. Celikyilmaz, A., Clark, E., Gao, J.: Evaluation of text generation: A survey (2020)
8. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. ACL-HLT pp. 190–200 (2011), https://aclanthology.org/P11-1020
9. Chklovski, T.: Collecting paraphrase corpora from volunteer contributors. In: Proceedings of the 3rd international conference on Knowledge capture. pp. 115–120 (2005)
10. Dopierre, T., Gravier, C., Logerais, W.: Protaugment: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. ACL-IJCNLP (2021), https://aclanthology.org/2021.acl-long.191
11. Dou, Y., Forbes, M., et al.: Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. ACL pp. 7250–7274 (2022)
12. Ethayarajh, K.: How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. EMNLP-IJCNLP (2019)
13. Freitag, M., Foster, G., et al.: Experts, errors, and context: A large-scale study of human evaluation for machine translation. Transactions of the Association for Computational Linguistics (2021), https://aclanthology.org/2021.tacl-1.87
14. Fujita, A.: Automatic generation of syntactically well-formed and semantically appropriate paraphrases. Ph.D. thesis, Ph. D. thesis, Nara Institute of Science and Technology (2005), https://api.semanticscholar.org/CorpusID:16348044
15. Fujita, A., Furihata, K., Inui, K., Matsumoto, Y., Takeuchi, K.: Paraphrasing of japanese light-verb constructions based on lexical conceptual structure (2004)
16. Goyal, T., Durrett, G.: Neural syntactic preordering for controlled paraphrase generation pp. 238–252 (Jul 2020)
17. Hegde, C., Patil, S.: Unsupervised paraphrase generation using pre-trained language models (2020)
18. Huang, S., Wu, Y., Wei, F., Luan, Z.: Dictionary-guided editing networks for paraphrase generation **33**, 6546–6553 (2019)
19. Huang, T.H., Chen, Y.N., Bigham, J.P.: Real-time on-demand crowd-powered entity extraction. https://arxiv.org/abs/1704.03627 (2017)
20. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks pp. 1875–1885 (Jun 2018)
21. Jiang, Y., Kummerfeld, J.K., Lasecki, W.S.: Understanding task design trade-offs in crowdsourced paraphrase collection. In: ACL 55th Annual Meeting. pp. 103–109. Vancouver, Canada (Jul 2017)
22. Koponen, M.: Assessing machine translation quality with error analysis (2010)
23. Larson, Cheung, Mahendran, et al.: Inconsistencies in crowdsourced slot-filling annotations: A typology and identification methods. COLING (2020), https://aclanthology.org/2020.coling-main.442
24. Li, Z., Jiang, X., Shang, L., Li, H.: Paraphrase generation with deep reinforcement learning. EMNLP (2018), https://aclanthology.org/D18-1421
25. Madnani, N., Dorr, B.J.: Generating phrasal and sentential paraphrases: A survey of data-driven methods. CL (2010), https://aclanthology.org/J10-3003
26. Mallinson, J., Sennrich, R., Lapata, M.: Paraphrasing revisited with neural machine translation. ACL European Chapter (2017), https://aclanthology.org/E17-1083
27. Metzler, D., Hovy, E., Zhang, C.: An empirical evaluation of data-driven paraphrase generation techniques. In: ACL 49th Annual Meeting. pp. 546–551. Portland, Oregon, USA (Jun 2011)

28. Negri, M., Mehdad, Y., Marchetti, A., Giampiccolo, D., Bentivogli, L.: Chinese whispers: Cooperative paraphrase acquisition. In: LREC'12. pp. 2659–2665. Istanbul, Turkey (May 2012)
29. Nilforoshan, H., Wang, J., Wu, E.: Precog: Improving crowdsourced data quality before acquisition. arXiv preprint arXiv:1704.02384 (2017)
30. Popović, M.: On nature and causes of observed mt errors. MTSummitXVIII (2021)
31. Prakash, A., Hasan, S.A., Lee, K., Datla, V., Qadir, A., Liu, J., Farri, O.: Neural paraphrase generation with stacked residual lstm networks. COLING (2016)
32. Raffel, C., Shazeer, N., Roberts, A., Lee, K., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR (2020)
33. Ramírez, J., Berro, A., Baez, M., Benatallah, B., Casati, F.: Crowdsourcing diverse paraphrases for training task-oriented bots (2021)
34. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. EMNLP (2019). https://doi.org/https://aclanthology.org/D19-1410
35. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: Behavioral testing of nlp models with checklist. ACL pp. 4902–4912 (2020), https://aclanthology.org/2020.acl-main.442
36. Su, Y., Awadallah, A.H., Khabsa, M., Pantel, P., Gamon, M., Encarnacion, M.: Building natural language interfaces to web apis (2017)
37. Sun, X., Liu, J., Lyu, Y., et al.: Answer-focused and position-aware neural question generation. EMNLP (2018), https://aclanthology.org/D18-1427
38. Thompson, B., Post, M.: Automatic machine translation evaluation in many languages via zero-shot paraphrasing. EMNLP (2020)
39. Thomson, C., Reiter, E.: A gold standard methodology for evaluating accuracy in data-to-text systems. INLG (2020), https://aclanthology.org/2020.inlg-1.22
40. Van, E., Clinciu, M., et al.: Underreporting of errors in nlg output, and what to do about it. INLG (2021), https://aclanthology.org/2021.inlg-1.14
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al.: Attention is all you need. Advances in neural information processing systems (2017)
42. Witteveen, S., Andrews, M.: Paraphrasing with large language models (2019)
43. Yaghoub-Zadeh-Fard, M., Benatallah, B., et al.: Dynamic word recommendation to obtain diverse crowdsourced paraphrases of user utterances. IUI (2020)
44. Yaghoub-Zadeh-Fard, M.A., Benatallah, B., et al.: User utterance acquisition for training task-oriented bots: A review of challenges, techniques and opportunities (2020)
45. Yaghoubzadeh, M., Benatallah, B., et al.: A study of incorrect paraphrases in crowdsourced user utterances. NAACL'19 (2019), https://aclanthology.org/N19-1026
46. Yaghoubzadehfard, M.: Scalable and Quality-Aware Training Data Acquisition for Conversational Cognitive Services. Ph.D. thesis, UNSW Sydney (2021)
47. Zamanirad, S.: Superimposition of natural language conversations over software enabled services. Ph.D. thesis, University of New South Wales, Sydney, Australia (2019)
48. Zeng, D., Zhang, H., Xiang, L., Wang, J., Ji, G.: User-oriented paraphrase generation with keywords controlled network. IEEE Access **7**, 80542–80551 (2019)
49. Zhou, J., Bhat, S.: Paraphrase generation: A survey of the state of the art. EMNLP (2021), https://aclanthology.org/2021.emnlp-main.414