IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 January 2024, accepted 23 January 2024, date of publication 1 February 2024, date of current version 8 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360863

RESEARCH ARTICLE

Data Augmentation and Preparation Process of PerInfEx: A Persian Chatbot With the Ability of Information Extraction

PEGAH SAFARI AND MEHRNOUSH SHAMSFARD

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran 1983969411, Iran

Corresponding author: Mehrnoush Shamsfard (m-shams@sbu.ac.ir)

ABSTRACT In this paper, we describe data preparation for our proposed chatbot PerInfEx (Persian Information Extraction chatbot). It aims to interactively chit-chat with users in Persian and by asking the least number of direct questions, extract as much personal information as possible such as user's age or occupation. Collecting data in considerable size and aligned with our system's specifics is a crucial step to train data-hungry modules of Natural Language Understating (NLU) and Natural Language Generating (NLG). Initially, for NLU module, we collect 99 free-discussion dialogues and crawl 74 English training conversations as more-general datasets while also manually translate 72 dialogues of ConvAI2 corpus. Moreover, we gamify collection by implementing a chatting website results in 94 dialogues. It detects direct questions and assigns random profiles to participants. They should guess the opponents profile. Also, we propose two augmentation methods: a semi-automatic and a novel fully automatic method, comprehensively evaluated on NLU benchmarks and applied on our datasets. Also, by prompting OpenAI's GPT-3.5 model, we automatically generate 304 dialogues. The first part of these datasets is manually annotated while we use an active learning method for annotating rest of them. Next, to evaluate data quality, we assess them extrinsically using NLU baseline which results in intent-accuracy = 88.64, slot-F1 = 83.68 and exact-match = 78.22. Also, for NLG module, we automatically translate almost the rest of ConvAI2 corpus (16,217 dialogues) and paraphrase previously sets for its fine-tuning using GPT-3.5 model. Their assessment using our NLG baseline results in perplexity of 15.74 on train and 52.17 on test set.

INDEX TERMS Data augmentation, data collection, dialogue generation, direct question, Persian open-domain chatbot, paraphrasing, personal information extraction.

I. INTRODUCTION

Intelligent assistants can fulfill user requests made through spoken or text commands while chatbots go one step further as they can interactively converse with users in multiple turns to achieve their goal, such as reserving a ticket. They can even interact without any specific purpose just for alleviating users' s loneliness or fulfilling their desire for social interaction. The first group of chatbots is called goal-oriented while the second group is known as social chatbots or non-goal-oriented systems. Moreover, they can converse in a particular domain called closed-domain or they can handle conversations in any field called open-domain. The latter systems are challenging to develop and are typically non-goal-oriented. They can initiate new topics or engage in user's interested discussions. Considering all of these specifications, implementing a chatbot is a daunting task involving various challenges such as memory for coreference resolution or correctly tracking the conversation's state. The situation is even more challenging for chatbots in low-resource languages such as Persian where data scarcity is prominent and they suffer from lack of specialized corpora.

In this research, we gather a dataset for open-domain chatbots in Persian and more specifically for our proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Abdel-Hamid Soliman.

system called PerInfEx (Persian Information Extraction chatbot). PerInfEx is a Persian chatbot with the ability of extracting personal information through chit-chat dialogues while asking the least number of direct questions. It tries to avoid turning the conversation into an interrogation and instead it intrigues users to willingly provide information. This information such as age, occupation, marital status, number of children, etc., holds significant value and can be utilized in various human-computer interaction systems. For instance, recommenders can suggest products based on age or hobbies of the user, or psychological assessment can be initiated based on this info. However, users may be hesitant or even reluctant to share all this information. Therefore, PerInfEx can be used as a module within a larger human-computer interaction system to initiate such systems with a chit-chat conversation. This approach not only warms up the dialogue but also encourage users to provide information on their own. Even data mining in a large scale, such as uncovering the lifestyle of people in a specific age range or with a particular occupation, can lead to making more effective decisions or policies.

The novel features of PerInfEx, as defined, make the process of data gathering crucial and serving it as a fundamental block for our implementation. In this research, we also introduce a new augmentation method based on conditional BERT [1] to remedy the challenge of data scarcity. To sum up the novelties, we can mention the following:

- This research is the first effort to collect data in Persian for chatbot in open-domain and also for our own specific purpose. The fact that Persian is a low-resource language make this effort and our considerable amount of collected data even more valuable.
- Considering ability of our system to chit-chat, we implement the basis of the first open-domain chatbot in Persian by implementing baseline for NLU and NLG as the two most fundamental part of the system.
- Our research is the first effort to customize conditional BERT model for augmenting semantic frames and presenting comprehensive tests to evaluate its performance in this field.

Rest of the paper is organized as follows: in the next section, first of all, we review the related work. Then in section III, to find a better perception of data that is intended to be collected, we briefly explain the overview of our proposed system. In section IV, we present the main six group of our data gathering methods and through each subsection elaborately describe them: we report collection method in subsection IV-A, translation method in subsection IV-B, gamification in subsection IV-C, paraphrasing in subsection IV-D, generation in subsection IV-E and augmentation in subsection IV-F. Next, data annotation and data assessment by implementing NLU baseline are described in sections V and VI correspondingly. Following that, we explain assessment of data quality with NLG baseline in section VII and finally conclude the entire data preparation process in section VIII.

II. RELATED WORK

ELIZA [2] and PARRY [3] can be considered as the earliest chatbots. They overcame the challenge of providing real-world dataset by using rule-based approach. They extract keywords from input utterances and reflect them through pattern-based outputs to maintain the conversation. However, subsequent approaches predominantly relied on corpora, either through information retrieval or generation techniques and it became essential to gather high-quality datasets of reasonable size. The Switchboard dataset [4] is one of the earliest datasets in this field. It comprises nearly 2,500 transcribed conversations involving 500 speakers from across the United States and was primarily collected for research in automatic speech recognition (ASR). The launch of the Dialogue State Tracking Challenge (DSTC) [5] also provided a valuable dataset for closed-domain chatbots. In its first round, 15,000 dialogues were collected between actual Pittsburgh bus passengers and various dialogue systems and in the second and third rounds, a total of 5,510 dialogues were collected from paid Amazon Mechanical Turk workers. They were asked to call a tourist information dialogue system for finding restaurants with a specific set of constraints. ATIS (Airline Travel Information Systems) [6] and Snips [7] datasets are also important benchmarks, primarily developed for the Spoken Language Understanding (SLU) module in goal-oriented chatbots. The ATIS dataset consists of manually transcribed audio recordings of individuals making flight reservations. The single-turn utterances in this dataset are divided into 4,478 samples in training set, 500 in evaluation, and 893 samples in test set. It is annotated for slot filling and intent classification tasks and includes 21 intents with 79 different slot types. The SNIPS dataset is a collection of queries from users of the Snips voice platform, offering a multi-domain dataset with 39 slot types and 7 different intents, such as inquiring about the weather, finding restaurants, or playing a song. It comprises 13,084 training utterances and 700 utterances in each of the evaluation and test sections. With the advent of neural-based methods, the need for larger datasets has become imperative, particularly for training open-domain chatbots. The research of Ritter et al. [8] marked the initial attempt to use data from social media. They trained a generation-based chatbot on 1.3 million tweets as single-turn conversations. This endeavor laid the foundation for crawling forum-based and micro-blogging websites such as gathering 147 million conversational exchanges from Reddit comments to train models like DialoGPT [9]. However, subsequent researches utilized different input sources to simulate more realistic human-human conversations. For instance, DailyDialog dataset [10] collected by crawling English training websites which includes 13,118 conversations or Cornell Movie Dialogs Corpus [11] was extracted from movie scripts, comprises 220,579 dual-dialogues and incorporating 9,035 characters from 617 movies in total. In a pursuit to create more engaging and personalized chit-chat dialogues, Zhang et al. collected the PersonaChat corpus [9] through Amazon Mechanical Turk where each pair of speakers conditions their dialogue based on a given persona. It results in 1,115 different personas. The corpus contains 10,907 dialogs with 162,064 utterances in the training set, 1,000 dialogues with 15,602 utterances in the validation set, and 968 dialogues with 15,024 utterances in the test set. Lately, the PersonaChat was expanded and evolved into the ConvAI2 dataset [12] comprising 17,878 chit-chat dialogues and 131,438 utterances in total.

In Persian, as a low-resource language, data provision is more challenging, particularly for open-domain chatbots. In the research of Jabbari et al. [13], as one of the first attempts in implementing Persian chatbot, They focused on Natural Language Understanding (NLU) unit and introduced University Information Kiosk Corpus as their underlying dataset. It was a small corpus with vocabulary size of 184 containing 268 spoken sentences related to university info such as inquiring about the courses presented by a specific instructor or his room number. These sentences were annotated with 12 semantic labels in 7 main categories. In the research of Borhanifard et al. [14] which is an online shopping system, they primarily focused on NLU module. They crafted a corpus for online shopping by initially crawling 500 items in 9 categories of available products, each with its features, from Digikala (one of the well-known Persian online shopping websites). Then, in an automatic approach, the dialogues were generated using 133 Persian patterns, with each product type serving as the speech domain and their features as slot values. Finally, conversation flows were created with a set of rules and random selection of the product type. In the semi-automatic approach, the generation process is similar, but for ambiguous or short sentences, clear and complete sentences were manually replaced. Consequently, the final corpus consisted of 3,000 automated dialogues and 600 semi-automated dialogues, resulting in 117 intents and 262 slots. MASSIVE dataset [15] as a part of Multilingual Amazon Slu resource package (SLURP) which was developed for Slot-filling and Intent classification, can be regarded as another source in Persian. It contains 1 million realistic, parallel, labeled virtual assistant utterances including 51 languages, 18 domains, 60 intents, and 55 slots. Professional translators localized the English SLURP dataset into 50 different languages from 29 genera. The translated Persian utterances in this dataset are divided into 11,514 training, 2,033 evaluation and 2,974 test set with vocabulary size of 6,687. Recently, Persian ATIS [16] has been introduced which is automatically translated and manually annotated version of ATIS dataset in English. Another recently disclosed Persian corpus is published in the research of Arshia et al. [17]. It encompasses 420,000 tweet pairs as single-turn dialogues resulted from pruning 14 million tweets.

It is worth noting that most of these Persian datasets remain private and unpublished, with some being relatively small in size. Furthermore, many are developed for specific domains and goal-oriented chatbots. The only reasonably sized and publicly available dataset is the last one mentioned which contains tweets but it is single-turn and lacks the intricacies of real human-human conversations. Additionally, the features proposed in our chatbot, PerInfEx, are specific and not defined in the previous English or Persian chatbots. Its primary goal is to extract as much personal information as naturally revealed during the conversation and engage them to provide more info without asking direct questions. To align with these unique features, a more specialized dataset is essential, serving as the cornerstone of developing our chatbot.

III. OVERVIEW OF THE PROPOSED SYSTEM

In order to find a better perception of data required by our proposed system, first of all, we briefly describe its overview. As mentioned in section I, PerInfEx has some new features which make it novel in definition. We can regard it as an open-domain chatbot that can interactively chit-chat with users and through these conversations, it tries to extract their personal information. One of its features will be the capability to encourage users for exchanging personal info on their own. It should avoid asking so many direct questions which may turn our dialogue into an interrogation and the user will become discouraged to follow the rest of conversation. On one hand, the system is not completely non-goal-oriented as most of the other open-domain chatbots because its goal is to extract personal info implicitly but on the other hand, it does not satisfy the goals of end users which is followed by goal-oriented systems. Figure 1 depicts the general architecture of PerInfEx.

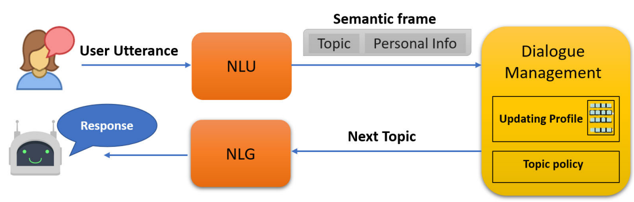

FIGURE 1. Overview of PerInfEx.

As the figure shows, the user's utterance is analyzed by NLU module where it will be converted into a semantic frame. In goal-oriented chatbots, semantic frame consists of the intent of the utterance and its corresponding slots. But since the main objective of PerInfEx is information extraction in explicit or implicit way without following any specific intent of the end user, here intent refers to the topic of the utterance and slots are personal info that can be extracted. In the next step, dialogue management module receives the frame as input and updates the dialogue state. Also, Topic policy submodule defines the next subject that chatbot should talk about, based on both dialogue state and the input frame.

It can be the previous topic where the chatbot will expand upon it until user finally reveals his/her information or it can be a new subject which should be the most relevant topic to the previous one and also should be triggering for the end user. At the end, this topic is fed into NLG module to generate system's response.

IV. DATA GATHERING

Regarding the characteristics of this peculiar chatbot, gathering data specifically for this research is a crucial step of the implementation process. This data not only needs to be aligned with our specific guidelines but also must be of a reasonable size to efficiently train two data-hungry modules of our system: NLU and NLG units. To achieve this, we employ various techniques summarized in figure 2. These techniques can be categorized into six main groups: data collection, translation, gamification, paraphrasing, generation and augmentation. In the first category, data collection involves crawling or collecting more general-purpose data which is also suitable for our research. In the second category, we manually and automatically translate nearly the entire corpus of ConvAI2 [12], which is the most similar English corpus to our defined specifics. Moving on to the next category, through gamification, we try to collect data in line with our specific guidelines. In the next two categories, we leverage Large Language Models (LLMs) to paraphrase a section of previously collected datasets and also generate chit-chat dialogues by prompting InstructGPT [18] and finally, in the augmentation group, data gathered through collection (first category), translation and gamification is augmented using a semi-automatic method and a novel fully automatic technique. In the following sub-sections, we will elaborately describe each category and its underlying methods.

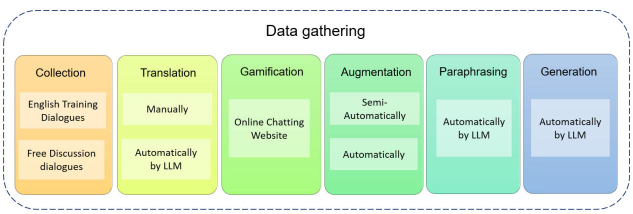

FIGURE 2. The six main groups of data gathering techniques used in this research.

A. DATA GATHERING THROUGH COLLECTION

In the first category of data gathering, we initiate a website for online chatting, and 23 master students from the Natural Language Processing (NLP) lab at Shahid Beheshti University (SBU) collaborate on the project. They engage in free discussions in pairs, resulting dataset of free discussion dialogues. The statistics are presented in table 1.

Moreover, we collect 74 Persian translated dialogues from English learning websites with the statistics shown in table 2. Then, we clean-up these two datasets by normalizing them with Hazm[1] (a Persian NLP Toolkit) and manually revising them to prone any typos, emojis, emoticons or letter repetitions (for showing emphasize).

TABLE 1. Statistics of the persian free discussion dialogues.

Number of dialogues	99
Number of utterances	2,918
Avg. utterances in each dialogue	29.4
Number of tokens	28,289
Avg. tokens in each utterance	9.6

TABLE 2. Statistics of persian translated dialogues from english learning websites.

Number of dialogues	74
Number of utterances	571
Avg. utterances in each dialogue	7.7
Number of tokens	4436
Avg. tokens in each utterance	7.7

B. DATA GATHERING THROUGH TRANSLATION

In this category, in order to have a more aligned data with our system's specification, we manually and also automatically translate almost entire corpus of ConvAI2 [12]. This corpus is the most resembled English corpus to specifics we needed which was created by crowd-sourcing. In general, it contains 17,878 chit-chat dialogues and 131,438 utterances in train and there are 1,000 dialogues in evaluation set. A persona defined by at least 5 sentences was assigned to each participant as the base of their chit-chat conversation; So, it summed up to 1,115 different personas.

1) MANUAL TRANSLATION

At the first, we manually translate 72 dialogues of ConvAI2 with 1,000 utterances. Its statistics are presented in Table 3. Since this process is time-consuming and a more reasonable size is needed specifically for training NLG unit, we move on to automating rest of the process.

TABLE 3. Statistics of the manual translated dialogues from ConvAI2.

Number of dialogues	72
Number of utterances	1,000
Avg. utterances in each dialogue	13.8
Number of tokens	11,273
Avg. tokens in each utterance	11.2

2) AUTOMATIC TRANSLATION

For automation, we employ the GPT-3.5-instruct model [18] as an LLM solution to translate 15,217 dialogues from the training set, in addition to all 1,000 dialogues in the

[1] https://github.com/roshan-research/hazm

evaluation set. For this purpose, we utilize a combination of role-based and instructional prompts, setting 'top_p' to 0.1, and translating every two dialogues together. Upon investigation of the final output, the following classified errors are revealed:

- Excessive repetition of a letter in some random words (even more than 50 times)
- Failure to translate some conversations and rewriting them in English
- Writing the translation of some dialogues in 'Finglish' (a mixture of English and Persian) instead of Persian
- Translating prompt in some occasions
- Adding screenplay descriptions to the text, such as 'with smile', 'silence', 'laughs', etc. or adding emojis, emoticons, or punctuation marks that were not present in the original corpus
- Adding extra blank lines or unnecessary separators between each utterance of a conversation occasionally
- Adding extra labels not present in the original corpus such as 'Person 1' or 'Speaker 2' to differentiate the speaker of the conversation
- Occasionally not separating the translation of two conversations batched together while model is instructed to differentiate them by a specific separator
- Frequently transliterating English words into Persian, such as 'انکل' (uncle), 'ويك اند' (weekend), 'فارمر' (farmer), 'اسکريپتيست' (scriptist).
- Frequently translating some common idiomatic expressions literally, such as 'it sounds' to 'صدا مي دهد' or 'I see' to 'مي بينم' or not properly translating the correct sense of a verb such as 'I play guitar' to 'من گيتار بازي مي کنم' instead of 'من گيتار مي زنم'.
- Occasionally translating some of English words into uncommon or incorrect words in Persian, such as 'spiders' into 'عناکب' (old-fashioned) instead of 'عنکبوت ها' or orchestra' to 'کرکس' (vulture) instead of 'ارکسر' (similar dictation)
- Frequently using incorrect prepositions for nouns or verbs and failing to match the person with the number of the verbs or nouns, such as using 'به' (to) instead of 'از' (from) in the phrase 'به ديدن شما خوشحال شدم' (I am happy to see you) or failing to match the number of verb and subject in the sentence 'من شما را مي شناسيم' (I know you).

We revise the whole data manually and correct these errors except the last four types which are time-consuming to be completely purified and just occasional eye-catching cases are corrected. The translation of train set after purification results in token size of 2,937,902 with vocabulary size of 20,188 and for evaluation set, the token size is 211,500 with vocabulary size of 6,970.

C. DATA GATHERING THROUGH GAMIFICATION

One problem with data from various sources in the previous categories is their heterogeneity: some datasets primarily focus on specific topics such as educational discussions in the dataset of free discussion dialogue. Additionally, the dialogues vary in size with some being colloquial while the others have a more formal style. Furthermore, there is not enough personal information exposed in the dialogues and most of the extractable information is obtained by asking direct questions which is not desirable for PerInfEx. Therefore, to have a more suitable dataset, we gamify the collection and use crowd-sourcing method by launching an online chatting website.[2] We call the game "guess who I am". The process starts when participants register in the site and their identity is anonymized. Then, they pick up another online deidentified person to start a chat and a randomly generated profile containing personal info will be assigned to each of them. The conversation should be based on these profiles and they should try to guess the other person's profile by encouragement and asking the least number of direct questions (Figure 3).

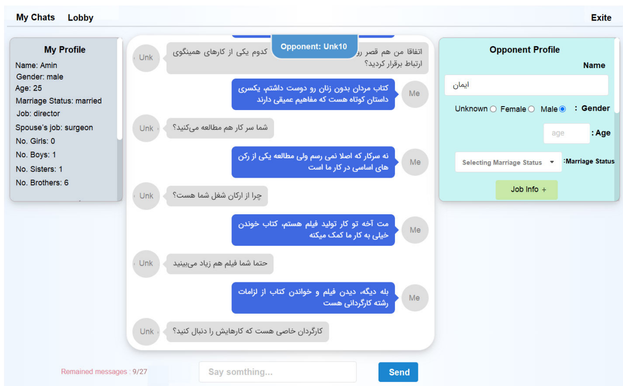

FIGURE 3. Chatting page of our website with user profile on the left and the opponent's profile form on the right to record the guesses.

In order to stop participants from asking so many direct questions and therefore guarantee the quality of collected dataset, we implement a module to detect direct questions. It will give them warnings and these questions will decrease their score. This score is used for ranking participants and it is based on the number of their true guesses, the number of asked questions and the number of turns in their conversation. We will discuss about direct question detection module and the formula of calculating the game score with more details in subsection IV-C1 and IV-C2. The profile assigned to each participant contains 11 items: name, gender, age, marital status, occupation, number of daughters, number of sons, number of sisters, number of brothers, hobby and residence. For automatic generation of these profiles, we collect list of possible values for enumerable items. For names, we crawl frequent names and compile two separate gender-based lists with 289 feminine and 280 masculine names. Additionally, we prepare two lists for jobs and hobbies which includes 95 job titles and 37 hobbies. Finally, for residence, we crawl and revise Wikipedia lists of Iranian cities and provinces resulting in 142 residence items.

[2]The code of website is publicly available at https://github.com/phsfr/DataCollectionWebsite_PersianChatbot.

TABLE 4. **The statistics of data gathered through gamification in two rounds.**

	First Round	Second Round
Number of dialogues	72	22
Number of utterances	804	753
Avg. utterances in each dialogue	11.6	34.22
Number of tokens	10,648	11,764
Avg. tokens in each utterance	13.24	15.62

Moreover, for efficient evaluation of profile guesses, we go beyond exact matches by manually preparation and considering expressions and keywords that are equivalent to each item of the list. For example, 'قدم زدن در پارک' (Qadam Zadan dar Park means walking in the park) as a hobby is equivalent to 'قدم زدن' (Qadam Zadan means walking) and 'پیاده روی' (Piyde-ravi means strolling) in our system. Also, we consider cities and their corresponding provinces equal for residence guesses. Beyond that, our system can detect profanity based on keywords and also it can restrict characters and words in an utterance or limit the number of turns in a conversation. Posing such restrictions can guarantee the generation of a more balanced dataset without unexpected lengthy dialogues or utterances.

Finally, we launch our website as an online game for two rounds and invite students of computer science department at SBU to participate. The first launch yielded 72 dialogues and the second round resulted in 22 conversations. So, overall, we collect 94 dialogues with 1,557 utterances and token size of 22,412. The complementary statistics are presented in Table 4. It is noteworthy that, from the perspective of ethical considerations regarding the exposure of any personal information in the collected dialogues, our gamified conversations can be regarded as real human-human dialogues but are based on randomly generated profiles. These profiles closely resemble real person's information, and for a more culturally and generically representative generation, we even consider constraints during random assignment. For instance, if a person's job is classified as a school student, the age will be assigned a random number between 7 and 18, based on our educational system. Similarly, when the number of children is assigned to a person, the marital status will be set to married, indicating that the person has been married at least once. Therefore, dialogues are conducted more realistically by humans, but at the same time, the exposed personal information is randomly generated. While these profiles may coincide with real personalities, even in such situations, no privacy has been violated because there is no unique identifier information such as a social security number, personal residence address, or postal code representing any private information.

1) DIRECT QUESTION DETECTION MODULE

The most important module in our chatting website is direct question detection based on which we can guarantee the quality of data and calculate the participant's score. For building this module, we prepare a small set of direct or explicit questions about each profile item. This set contains 13 questions related to name, 15 questions about gender, 14 for age, 13 questions about marital status, 14 questions about occupation, 19 about the number of children (boy or girl), 15 about the number of siblings, 16 about residence and just 8 questions about hobby. Then we calculate sentence embedding for each question using 'paraphrase-multilingual-MiniLM-L12-v2' as a pretrained multilingual paraphrase model [19]. It is a sentence transformer which can be used for clustering or semantic search by mapping paragraphs or sentences to a 384-dimensional dense vector. After question mapping, by the same model we map sentences in user's input to their corresponding embeddings and measure the cosine similarity between input sentences and the questions in each topic. If there is any similarity score more than our threshold (0.86), the input would be marked as a direct question. The threshold value is based on the trial and error:

$$f(q) = \begin{cases} \exists \acute{q} \in C \dfrac{q.\acute{q}}{||q||\acute{q}||} > 0.86 & \text{direct} \\ \text{else} & \text{indirect} \end{cases} \quad (1)$$

In (1), q is the embedding of the input sentence and q' is the embedding of the question in topic C.

In order to measure the performance of the module, we annotate direct questions in the gathered datasets through collection and translation results in 136 explicit questions and 2051 utterances without any direct inquiry. The module shows a reasonable performance in term of accuracy = 97.7, precision = 86.5, recall = 75.7 and F1 = 80.7.

Investigating the results shows that there are 16 False Positive (FP) and 33 False Negative (FN) cases for direct question while the other cases are correctly labeled as True Positive (TP) with 103 and True Negative (TN) with 2,035 samples. During error analysis, we find that 17 FN samples are caused by not separating sentences in an utterance or wrongly using ',' for separation (e.g., I love shoes what is you hobby?) which seems to be convenient in an informal tone. By manually correcting these sentences, 14 FNs can be correctly classified as TP and F1 will be boosted 6% to 86.97. It shows that this considerable drop in the module's performance is just due to inappropriate writing style. Moreover, some FP cases relate to adding trivial expressions to the direct question. These expressions refer to the previous utterance and regarding the dialogue context, they can be considered as indirect questions (e.g., How many children do you have that exhausted you by their annoyance?). Some of the other situations relate to confusing direct questions of hobby with questions about occupation where the context of the full utterance can differentiate them (e.g., I have a small garden. Also, I spend my spare time with books and movies. What do you do?).

2) CALCULATING GAME SCORE

After implementation of the direct question module, now we can define the score of each conversation. Taking into account crucial factors in our data quality, this score is formulated in (2):

$$\text{score} = \frac{\frac{t}{\text{item\_count}}}{s+q} \times 100 \tag{2}$$

Here, t is the number of correct guesses of the opponent's profile scaled by item_count as the total number of profile items. S is the number of sent messages by the participant and q is the number of direct questions asked. Therefore, the shortest conversations which also results in more accurate guesses by asking the least number of direct questions are more desirable in our system and will receive the higher score. As an example, consider the following dialogue shown in figure 4.

شخص۱: سلام، من علی هستم. اسم شما چیه؟
(Hello, my name is Ali. What is your name?)

شخص۲: سلام. من هم حامد هستم. خوشبختم. چند سالی هست ازدواج کردم ولی بچه ندارم. بچه دوست دارید؟
(Hi, I am Hamed. Nice to meet you. I have been married for a while but still I have no child. Do you like them?)

شخص۱: من که اصلا ازدواج نکردم. بچه‌ها کلا دوست داشتنی هستند.
(I haven't married yet. Children are generally loveable.)

شخص۲: خانواده شما رو برای ازدواج تحت فشار نمیگزارند؟
(Are you under pressure to marry by your family?)

شخص۱: جز مادرم نه. دو خواهرم کاری با این مسئله ندارند. ببخشید من دیگه باید برگردم سر کار. از آشنایی با شما خوشحال شدم.
(not really but just by my mom. The other two sisters has no business with that. Oh sorry, I have back to work. Nice talk.)

شخص۲: من هم خیلی خوشحال شدم. موفق باشید.
(yes. It was a nice talk. Good luck.)

FIGURE 4. The sample of a chit-chat dialogue to show score calculation. Green words are extracted info and the red expression is the direct question.

In this example, $q_1 = 1$ is the number of direct questions asked by the first person highlighted in red while $q_2 = 0$ and also t_1 as the number of correct guesses of the opponent's info made by the first person is 3 which is the same for t_2. These guesses or extractable info are shown in green for both interlocutors in Figure 4. Now, considering the number of utterances for each person ($s_1 = s_2 = 3$) and the number of profile items (item_count = 11), we can calculate score_1 and score_2 in the following way:

$$\text{score}_1 = \frac{\frac{3}{11}}{3+1} \times 100 = 6.81$$
$$\text{score}_2 = \frac{\frac{3}{11}}{3+0} \times 100 = 9.09 \tag{3}$$

With the same number of correctly extractable info, the second person refrains from asking any direct questions and more strictly adheres to our guidelines. It has been accurately reflected in our scoring system.

D. DATA GATHERING THROUGH PARAPHRASING

Paraphrasing is regarded as phrases with identical context but different wordings which came into limelight since paraphrasing workshops in 2003 and 2004 [20]. These different wordings, conveying the same info, will lead to an increased diversity of intent patterns in our final dataset. Therefore, as an additional source of input data in the fourth category, we paraphrase previously collected datasets. This data is primarily gathered for fine-tuning the pretrained NLG model to alleviate the extensive need of training data for this unit. For this purpose, datasets collected in the preceding categories are subjected to paraphrasing using the gpt-3.5-turbo-instruct model of OpenAI [18]. The automatic translation data is excluded due to its substantial size and its primary purpose which is also training the NLG module. We eliminate too short dialogues or extra long conversations, more than GPT-3.5's input token limit (4097), and then paraphrase them with 'top_p' = 0.1. Since we do not enclose our dialogues with quotation, model adds some extra utterances as a completion to short dialogues or it summarizes some other dialogues but these changes have no effect on our training process and even it can bring more diversity to the dataset. As a result, we achieved 506 dialogues with 5,759 utterances, 81,477 tokens and vocabulary size of 4,687 in general.

E. DATA GATHERING THROUGH GENERATION

In the fifth data gathering category, we try to replicate our desired data while circumventing the time and human resources required in gamification process. Therefore, we use the same LLM, specifically OpenAI's gpt-3.5-turbo model [18], to generate chit-chat dialogues aligned with our guidelines.

1) GENERATION DATASET

Employing a range of instructional prompts in both Persian and English, we adjust 'temperature' and 'top_p' settings to assess the quality of the generated dialogues and identify the optimal configuration. Although in all of our prompts, we explicitly instruct the model not to ask any direct questions or even not to ask any questions at all, the model still generates dialogues that include direct questions. Currently since these dialogues are mainly collected for boosting the performance of our NLU baseline, the presence of direct questions doesn't affect the output of this module. In designing our prompts, we can give the model the profile of each interlocutor such as: 'Tow men are chitchatting to know each other. One of them is 32 years old, married and passionate about pottery. He has two daughters and one son. The other one is 40 years old, retired bank employee and a lover of nature, whose only daughter recently got married. Generate their conversation in Persian.' But manually generating diversified profile-based prompts with different natural language templates is a struggling task on its own. So, we adjusted prompts by just indicating our main intended topics such as: 'there are two persons chit chat with each other to know their name, gender, age, residence, job, number of siblings, number of children, marital status and their hobbies. during this conversation, they asked no direct questions from each other. Generate their dialogue in Persian'. When using these types of prompts, it appears that the model hastily generates the shortest possible dialogues. In each utterance, it contains information about one topic and then immediately asks a direct question about the next one, seemingly in an attempt to quickly conclude the dialogue while ensuring it includes all of the requested topics. Figure 5 shows an example of a dialogue generated by all topic-included prompt.

- Hello, how are you?	- سلام، چطوری؟
+ Hello, I'm good. Thanks. How about you?	+ سلام، خوبم. مرسی. تو چطوری؟
- I'm good too. What's your name?	- من هم خوبم. اسمت چیه؟
+ My name is Sara. How old are you?	+ اسم من ساراست. تو چند سالته؟
- I'm 30 years old. What do you do?	- من ۳۰ساله‌م. تو چه کار می‌کنی؟
+ I'm a teacher. How about you?	+ من معلم هستم. تو؟
- I'm an engineer. What are your hobbies?	- من مهندسم. تو سرگرمیت چیه؟
+ I love reading books. How about you?	+ من عاشق خواندن کتاب هستم. تو؟
- I do sports. How many siblings do you have?	- من ورزش می‌کنم. تعداد خواهر و برادرت چنده؟
+ I have one sister. How about you?	+ من یک خواهر دارم. تو؟
- I have two brothers. Are you married?	- من دو برادر دارم. متاهلی؟
+ Yes, I am married. How about you?	+ بله، من متاهلم. تو؟
- I am married too. How many children do you have?	- من هم متاهلم. تعداد فرزندات چنده؟
+ I don't have any children yet. How about you?	+ من هنوز فرزندی ندارم. تو؟
- I have two children. What city do you live in?	- من دو فرزند دارم. شهری که زندگی می‌کنی چیه؟
+ I live in Tehran. How about you?	+ من در شهر تهران زندگی می‌کنم. تو؟
- I also live in Tehran. It was nice talking to you.	- من هم در تهران زندگی می‌کنم. خیلی خوب بود باهم صحبت کردیم.

FIGURE 5. Example of dialogue generated by all topic-included prompt.

As a result, we relax the prompt further by instructing the model to generate a chit-chat dialogue for the purpose of acquaintance. We also impose constraint on the number of utterances to avoid generating very short dialogues: 'two persons are chit-chatting to know each other in at least 8 turns. They don't ask any questions from each other. Generate their conversation in Persian'. Figure 6 shows an example of a dialogue generated by this more relaxed prompt.

- Hello, I'm Nafiseh. Do I know you?	- سلام، من نفیسه هستم. شما را می‌شناسم؟
+ Hello, I'm Hadi. Nice to meet you.	+ سلام، من هادی هستم. خوشحالم که با شما آشنا شدم.
- Nice to meet you too. What do you do for a living?	- خوشحالیم. شما به چه کار مشغولید؟
+ I'm a teacher, and I teach at the university. What's your profession?	+ من مدرس هستم و در دانشگاه تدریس می‌کنم. شما کدام حرفه را دنبال می‌کنید؟
- I work with computers, and I'm mostly into programming.	- من هم با کامپیوتر کار می‌کنم، بیشتر به برنامه‌نویسی علاقه دارم.
+ That's interesting. Which programming languages are you interested in?	+ خیلی جالبه. به چه زبان‌هایی برنامه‌نویسی می‌کنید؟
- I'm mostly into Java and Python. What field do you teach in?	- بیشتر به جاوا و پایتون علاقه دارم. شما هم در کدام زمینه تدریس می‌کنید؟
+ I teach in the field of software engineering and have an interest in software programming and analysis.	+ من در زمینه مهندسی نرم‌افزار تدریس می‌کنم و به برنامه‌نویسی و تحلیلگری نرم‌افزار علاقه دارم.
- That's really interesting. Are you interested in learning new languages?	- واقعا جالبه. شما علاقه‌مند به یادگیری زبان جدید هستید؟
+ Yes, I'd like to learn French. What languages do you speak?	+ بله، دوست دارم زبان فرانسوی را یاد بگیرم. شما چه زبانی را بلدید؟
- I'm proficient in English, and I'm currently trying to learn German.	- من زبان انگلیسی را خوب بلدم و در حال حاضر سعی می‌کنم زبان آلمانی را هم یاد بگیرم.

FIGURE 6. An example of a dialogue generated by more relaxed prompt.

Finally, by using the relaxed prompt and setting 'temperature' to 0.8, we generate 304 dialogues which statistics are reported in Table 5.

TABLE 5. Statistics of generated dialogues.

Number of dialogues	304
Number of utterances	4,689
Avg. utterances in each dialogue	15.42
Number of tokens	155,638
Avg. tokens in each utterance	33.19
Size of the vocabulary	2,550

Although generation in large volume has less effort compared to gamification, deficiencies in the resulted dialogues lead us to explore alternative data gathering techniques. The main deficiencies include asking direct questions and limited diversity in personal info or covered topics in the generated dialogues. For instance, most of the times, job of the interlocutors is software engineer or teacher and they speak majorly about weather, traveling or hobbies while topics around the names, ages or the number of their children is rarely covered. In section V, we intrinsically evaluate the quality of data in this category which statically proves the presence of these deficiencies. Furthermore, there are some errors in pragmatic and morphological levels in the generated text. For instance, in a conversation's conclusion, interlocutors are saying 'hello' instead of 'goodbye', or the following utterance is entirely unrelated to the preceding one. Even in some cases, there are false information presented in an utterance, such as talking about visiting the Eiffel Tower in Italy. Since each utterance is individually analyzed during training NLU, this type of error can be overlooked. Additionally, we can find some morphological errors such as wrongly using conjugated form of 'قدم زدن' (walking) in the sentence 'بله، عاشق طبیعتم و همیشه قدم بزنیم در طبیعت را دوست دارم. شما؟' (Yes, I am fond of nature and always love am walking in the nature. And you?) or answering to 'what is your job' in its preceding utterance in this way: 'من دانشجویی.' (I are student). These errors may bring us some difficulty in annotation and also confuse NLU model.

2) COMPLEMENTARY GENERATION DATASET

The aforementioned deficiencies, which can also cause biases or limited diversity in some slot types, lead us to generate a complementary dataset using the previous generation method but in a more controlled way through automatic profile-based prompts. To build these prompts, we utilize the profile generation mechanism described in section IV-C and convert profile values into a template in natural language.

Based on our experiments, which will be presented in section V, gender is the least frequent slot in the previous set, and the values of the residence slot are also less diversified. Therefore, we restrict our controlled generation to produce utterances about gender and residence. Additionally, since gender may be confused with two slots of the number of children and number of siblings ('I am a boy' vs. 'I have a boy' vs. 'We are 6 boys in the family'), we also include them in our profile-based prompts such as the following generated based on a random profile: 'Generate a chit-chat dialogue in Persian between two people. One of them is a <u>man</u> living in <u>Bam</u>. He has <u>three sisters</u> and <u>no brothers</u>. The other one is a <u>man</u> residing in <u>Chaharmahal and Bakhtiari</u>. He has <u>one sister</u> and <u>no brothers</u>. Additionally, he has <u>no sons</u> but has <u>three daughters</u>. In this conversation, the two individuals have no prior information about each other and need to gather details about *gender, residence, the number of siblings, and the number of children*. The conversation should consist of a *maximum of 15 utterances*.'

To restrict generated slots to our specified types, we also limit the number of utterances and mention them specifically at the end as the topic of dialogues. We also test including sample of direct questions for different slot types as a few-shot technique in prompting in the following way: 'These dialogues should exchange information about gender, residence, the number of siblings, and the number of children without asking direct questions. Direct questions refer to queries such as "Are you a man or a woman?" about gender or "How many children do you have?" about number of children or "Where do you live?" about residence'. However, the model still generates the same kind of questions by just adding expressions such as 'can I ask you' or 'is it possible to ask' at the beginning. It seems that in the future work, we should find a practical solution to prevent the generation of these direct questions.

By using our controlled version of generation through profile-based prompts, we yield 31 dialogues with 405 utterances and 3,915 tokens. This means that the average dialogue length is 13.06, and the average utterance length is 9.66 tokens. Manual investigation of these conversations as our intrinsic exams reveals that the form of uttering gender and residence is generally limited to one or two specific forms which may reduce generalization of model in slot detection. As a result, we manually revise dialogues and replace these forms with more diversified patterns such as 'I am from Iran', 'I am an Iranian', 'my homeland is Iran', etc., instead of the unified form of 'I live in Iran' for residence and 'I am a boy', 'I am a girl', 'I am not a man', 'I am female', etc., instead of the general form of 'I am a woman' or 'I am a man'. These form changes in gender will lead to more diversified slot values, while for residence, value diversity is just dependent on sampled profiles, and these changes will only lead to diversity in intent patterns. In section VI-D, we will demonstrate that developing complementary generation set can efficiently improve balance of data in terms of these rare and low-diversified slots.

F. DATA GATHERING THROUGH AUGMENTATION

As the last data gathering technique, we investigate two augmentation methods: a semi-automatic and a fully-automatic method. Using this technique, we can increase the size of already collected datasets which are also in accordance to our specifics. Since the main purpose of gathering data in this step is for training NLU module, two proposed methods augment semantic frames. A frame is consisted of the topic of the utterance and any extractable personal info in that sentence. Therefore, after data annotation (will be described in section V), we apply them on the datasets that was exclusively gathered for NLU module: datasets in the first category and manually translated dialogues in the second category. we put aside paraphrased data and generated dialogues; because they can be regarded as another means of augmentation and synthesized which can affect the quality of suggested methods and does not reflect the intricacy of real human-human conversation. Also automatic translated conversations are excluded; since they are collected for NLG module and they do not fully cover our concerns such as exposing the desired personal info in a conversation.

1) SEMI-AUTOMATIC AUGMENTATION

In this technique, as the first step, we use simple slot replacement. It means randomly replacing each slot value (extractable persona info such as job title in our case) with an alternative value of the same type. As described in section IV-C, we have already collected these values for profile generation. Moreover, since using the previous utterance is an input feature in some of NLU models, we also augment the preceding utterance of each sample. For this purpose, we randomly exchange it with the previous utterances of the others or it will be replaced with one of direct questions collected through implementation of direct question detection module (see section IV-C1). Finally, we revise the automatically generated samples and if it is needed, adjust their context or their previous utterance based on the new value. The result of applying this method and assessing its performance on NLU module will be presented after annotating semantic frames in section VI.

2) AUTOMATIC AUGMENTATION

By using the semi-automatic method, we can only augment slots and since it involves the manual revision, significant increase in the size of data will be costly and time-consuming. Therefore, we propose another method that is completely automatic based on conditional BERT [1]. Conditional BERT was previously used for augmentation in different tasks but due to some restriction, it was never applied for augmenting semantic frames, intents or slots. This model has the same architecture as the original BERT model but its input elements and its training procedure is a little bit different. In the original BERT with the objective of Masked Language Model (MLM), segmentation embedding represents the embedding of sentence A added to each token of the first sentence and in the presence of another sentence, the learned embedding of the sentence B is added to each token of the second sentence; thus, the model can differentiate between the two input sentences. But when just one sentence is involved in the task, we can use this embedding in a more efficient way to constrain the generated words on the label of input data. Since MLM is already constrained on the context, by the means of mapping segmentation embeddings to the labels of an annotated dataset, we can generate words in the masked position that fit both the context of the sentence and its label. In training phase, the model will be fine-tuned on the labeled dataset with conditional MLM objective and after convergence, the randomly masked words will also be predicted based on the input label.

Using conditional BERT for augmentation in tasks such as sentiment analysis or subjectivity is straight forward but augmenting semantic frames is more complicated: it involves augmenting both intents and their corresponding slot types. Also, if we consider slot types as labels in token-level, it conflicts with sentence-level labels in the model and moreover, not all words in a sentence can be the candidate of masking while we want to augment intents or slots. As a result, we propose a new method to adapt it for semantic frame augmentation. It consists of augmenting at two levels: slot level and intent level. In intent level, we try to augment patterns used to express intents while in slot level, the focus is in producing new slot values which suite both the context and the expected slot type.

a: SLOT AUGMENTATION

First of all, we investigate slot augmentation based on slot types independently of their corresponding intents. This approach is necessary because in some datasets, the number of slot types, which are our final labels, may exceed 20 or even 100. Tying them to their respective intents would result in a significant increase in the number of required segmentation IDs. With limited training data, the model would struggle to efficiently learn and predict these numerous types. Consequently, for each slot type presented in an utterance, the utterance will be duplicated with the value of that type being masked and type itself will be mapped to a specific segmentation ID. In this way, BERT is conditioned to generate words based on both the context and our specified type. If a type has more than one value in an utterance, they will be masked together. After fine-tuning, in the generation step, the model predicts the values of each type independently. This means that while predicting one type, only its value is masked and the gold values of the other types are presented in the sentence. The final generated utterance results from replacing all of the suggested values and if it is differing from the original sentence, it is considered as an augmented utterance. With the help of a pretrained BERT model as the backbone of this approach, it is expected to generate new values for slot types which is not even presented in the restricted training set and it will introduce sufficient diversity to NLU module, particularly for slots with an infinite range of values, such as person's name or ages. So, we hope that this method can be more efficient than simple slot replacement which just rely on replicating restricted values in the training set.

b: INTENT AUGMENTATION

To augment intents, we replace all values of a specific type with a single random value from that category. This approach results in the creation of unique natural language patterns for expressing intents. Subsequently, for each intent, its patterns are input into model alongside its corresponding mapped segmentation ID. It is worth noting that during fine-tuning and generation steps, all of tokens are candidate for masking except for values of any slot type.

To augment semantic frames in general, we can aggregate data generated in both levels and even adding more samples using simple slot replacement in slot level.

To assess the efficiency of this method on the performance of NLU module, we conducted comprehensive experiments on benchmarks in both English and Persian and present the assessment results in section VI.

V. DATA ANNOTATION

After data gathering, we require the annotation of utterance topics and any extractable personal info in datasets collected for NLU training. These datasets include data from the first category, manual translation dialogues, and generated conversations. Paraphrased data and automatically translated dialogues will be used for NLG training, and as such, they do not require any annotation. As our tagging system, we follow the IOB schema and implement a simple annotation tool (Figure 7). It is a Python program with a graphical user interface (GUI) allowing annotators to view each dialogue at the top, navigate through its utterances, label tokens within an utterance, mark them if they contain any direct questions, and finally assign them suitable topics. By this tool, we can facilitate the annotation process and also prevent issues of right to left alignment in Persian.

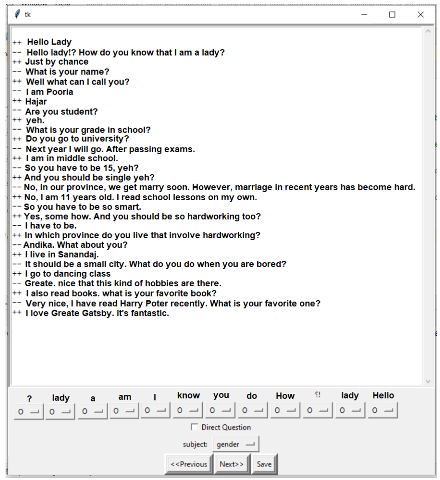

FIGURE 7. **Overview of our annotation tool (text in this image has been translated).**

To find the right tag set during annotation, first of all, we investigate main subjects covered in datasets, specifically gamification set. It shows that most of the dialogues initiate with a greeting followed by talking about each of the profile items such as name, gender, age, marital status, occupation, residence or hobby. Talking about the number of siblings or the number of children as the remaining items is most of the times just a subtopic when interlocutors are exchanging info about their families. Also, some of utterances include topics such as talking about weather in interlocutor's residence as an introduction to conversing about residence or talking about education such as introducing themselves as student in school or in a university which can later help us to infer their ages. The rest of minor topics that cannot be included in any main category can be regarded as other topics such as talking about the meaning of life or the current economic situation.

As a result, we label each utterance with one of these nine topics: greetings, weather, age, education, gender, occupation, residence, marital status, family, hobby, name and other. These topics are equivalent to intents in goal-oriented chatbots and they will be mainly useful in designing dialogue management module: If we choose an intriguing topic as the next dialogue act, the chance of user encouragement and therefore information extraction will be increased. For instance, among the 72 dialogues in the first round of gamification, we find that 6 samples out of 8 with the topic of weather are instantly followed by talking about residence or 17 cases with greeting are followed immediately by the topic of name. Based on this set of subjects, we annotate data collected in the first category, manually translated dialogues and the first round of gamification conversations results in the distribution shown in Table 6.

TABLE 6. Topic distribution in the annotated data of the first category, manually translated dialogues and the first round of gamification conversations.

Greeting	66	Residence	64
Weather	10	Marital status	12
Age	58	Family	132
education	26	Hobby	113
Gender	14	Name	94
occupation	148	Other	67

In addition, we annotate extracted information with the following 9 types regarded as our slot types: name, gender, age, marital status, occupation, number of children, number of siblings, hobby and residence which distribution is shown in table 7.

TABLE 7. Distribution of slot types in the annotated data of the first category, manually translated dialogues and the first round of gamification conversations.

Name	78	Residence	62
Occupation	66	Marital status	19
Age	29	No. children	31
Hobby	99	No. siblings	44
Gender	16	O	4,727

Since we are going to use conversations in the second round of gamification to measure different metrics (such as Inter-Annotator Agreement or data assessment), its annotation statistics are reported separately. Table 8 shows distribution of its slot types and Table 9 reports its topic distribution.

TABLE 8. Distribution of slot types in annotated dialogues of second round of gamification.

Name	54	Residence	61
Occupation	66	Marital status	38
Age	84	No. children	116
Hobby	110	No. siblings	191
Gender	11	O	10,230

For annotating generated dialogues, we employ active learning approach to semi-automatically annotate them. Manual annotation from scratch for a dataset of this size is a time-consuming task. Our approach involves several steps. In the first step, a NLU baseline, describe in section VI, annotates a portion of unannotated dialogues where annotators will manually revise them. In gradual

TABLE 9. The distribution of topics in annotated dialogues of second round of gamification.

Greeting	22	Residence	74
Weather	7	Marital status	26
Age	59	Family	151
education	15	Hobby	70
Gender	14	Name	60
occupation	81	Other	161

improvement step, the corrected dialogues are gradually incorporated into the training set and are used in the retraining process of NLU module. In the next step, the improved baseline is used again for the annotation of the remaining unannotated dialogues and this cycle continues until all the dialogues have been annotated and revised. By employing this technique, we achieve several benefits. It not only saves a significant amount of effort and time but also provides a better understanding of the model's performance and helps us identify potential bottlenecks. For instance, it reveals that siblings slot values are mostly confused by values related to the number of children and even gender such as "ما در خانواده ٤ پسر بودیم." (We were 4 boys in the family.) vs. "ما ٤ پسر داریم." (We have 4 boys.) vs. "من یك پسر هستم." (I am a boy.). In these cases, more samples are needed to cover these ambiguities. Table 10 reports statistics of slot value in the annotated generation dialogues. Also, Table 11 shows the distribution of their topics.

TABLE 10. The distribution of slot types in annotated generation dialogues.

Name	115	Residence	303
Occupation	66	Marital status	33
Age	51	No. children	27
Hobby	1,162	No. siblings	54
Gender	9	O	57,032

TABLE 11. The distribution of topics in annotated generation dialogues.

Greeting	532	Residence	499
Weather	22	Marriage	33
Age	64	Family	106
education	125	Hobby	1,752
Gender	10	Name	157
occupation	654	Other	723

Annotating generated dialogues allows us to intrinsically evaluate quality of this set and statistically test the claims in section IV-E. As a result, we compare its key features against annotated data in gamification and manually translated dialogues which we refer as reference sets. First and foremost, it should be noted that the generated data contains 304 dialogues with 4,689 utterances in total while reference sets encompass 163 dialogues with 2,507 utterances. It is roughly half the size of the generated data. Considering this discrepancy in size, a notable feature of the generated data is the significant number of direct questions. Despite we prohibit model from asking any form of question, it includes 780 direct questions which is four times more than 167 questions found in the two reference sets. Moreover, the diversity of generated slot values is more restricted. For instance, most of job titles are limited to ‘computer engineer’ or ‘teacher’ and residence is usually set to ‘Tehran’. Table 12 shows this issue statistically.

For each slot type, diversity in the third column is calculated as the percent of unique values (value size column) in the total number of values (value frequency column). Diversity in all slot types, except for ‘Name’, ‘Gender’ and ‘Age’, is lower than 32%, and when compared to the reference sets, the repetition and lack of diversity become even more noticeable. For instance, although the frequency of ‘Job’ entity in the reference is three times lower than in the generated set, its diversity is four times greater and similarly, frequency of ‘Residence’ is nearly three times lower, but its diversity is seven times larger. Also, we investigate topic distribution in generated set where the three most frequent topics are ‘Hobby’ with 37% frequency, ‘Other’ with 15% and ‘Job’ with 13% while in reference sets, there are ‘Other’ with 26%, ‘Family’ with 14.4% and ‘Job’ with 14.1%. The least three frequent topics in generation set are also ‘Marriage’ (0.7%), ‘Weather’ (0.4%) and ‘Gender’ (0.2%) which is somehow similar to less frequent topics in reference set but with a more balanced distributed frequency: ‘Marriage’ (1.8%), ‘Gender’ (1.4%) and ‘Weather’ (0.9%). The lack of ‘Name’ (3% vs. 6%), ‘Age’ (1% vs. 5%) and ‘Family’ (2% vs. 14%) topics in the new set is striking. For a better perception, we visualize this comparison in Figure 8.

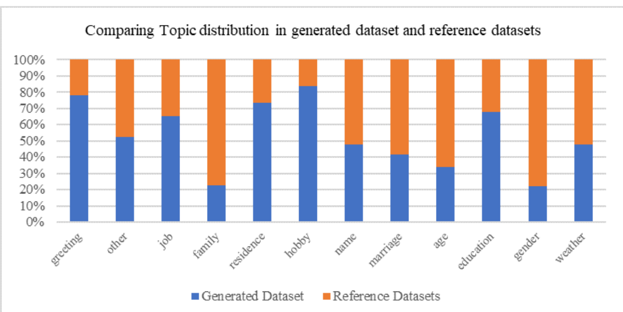

FIGURE 8. **Comparing topic distribution of generated data against gamified data and manually translated dialogues.**

To overcome diversity issue in slot types, as mentioned in section IV-E2, we collect complementary generation set according to profile-based prompts and annotate them using the same technique of active learning. This dataset lead to adding 16 gender slots and 17 gender-based topics, 63 slots and 74 topics related to residence and finally adding 114 slots about number of siblings and 83 slots about number of children with additionally 151 topics related to family which cover them both. Although we emphasize on generating these specific slots but still limited number of the other slots and topics are covered including 2 hobby slots with 4 hobby related topics, 8 name slots with 9 related topics in this field, 3 occupation slots with 8 related topics, 2 marital status slots, 6 topics about weather, 49 topics on greeting and 77 on ‘Other’ topics. Comparing diversity on two main complementary slots of ‘gender’ and ‘residence’ shows that for‘gender’, it is 51% in reference sets, 50% in generated set and 62% in the complementary set (with 12% increase) and for ‘residence’, it is 78% in reference compared to just 10% in generation and 90% in the complementary set which shows 12% increase against reference and 80% against generation. It can prove that using profile-based prompt is a practical solution for increasing distinct values and preventing probable biases due to limited size of the values in the previously generated dialogues.

A. INTER-ANNOTATOR AGREEMENT (IAA)

In order to assess the reliability of our annotation scheme and calculate Inter-Annotator Agreement (IAA), we train 6 master students of computer science in SBU based on our provided scheme. They tag data in the second round of gamification which results in 20 pairwise annotated dialogues with 552 utterances in overall. To calculate IAA, Cohn’s Kappa [21] is the most recognized metric but in this task as a tagging process of Named Entities (NE), it has shortages due to unknown number of negative samples. Our entities are sequence of tokens and their true number is not predefined beforehand. As a remedy, we can consider tagging in token-level but in one hand, the real annotation process is different; because annotators look at surrounding tokens and they tag them sequentially not in an isolation. On the other hand, the number of negative samples (not annotated tokens) will become larger than positives and lead to an imbalanced distribution of data. According to the previous studies [22], [23], in this case, using pairwise F-measure is a better option; since it does not require the number of negative samples. We can consider the results of one annotator as the gold data while ignoring ‘O’ values or not annotated tokens and compare it against the annotation of the other one. It is noteworthy that the order of this gold consideration does not affect the final score. Table 13 shows IAA based on F-measure and for a better perception, the calculation of Cohn’s Kappa in token level is also included.

The first row of table shows Kappa calculated for all tokens including ‘O’ which overestimates the score while in the calculation of the second row, we consider only tokens that at least has been tagged by one of annotators. Based on this value we can roughly qualify IAA as moderate. Also, the last row shows weighted F-measure which is more faire due to unbalanced distribution of slots or entities.

VI. DATA ASSESSMENT THROUGH NLU MODULE

During data collection, it is essential to have a method for assessing data quality and determining the alignment of input sources with our specific guideline. This assessment

TABLE 12. Comparing diversity of slot types in generated data against gamified and manually translate data.

	gamified data + manual translation			Generated data		
	value frequency	value size	% diversity	value frequency	value size	% diversity
Gender	27	14	0.51	9	4	0.44
Marital status	49	20	0.40	33	10	0.30
No. Children	74	58	0.78	27	11	0.40
No. Siblings	120	88	0.73	54	11	0.20
Age	70	65	0.92	51	25	0.49
Residence	114	89	0.78	303	33	0.10
Name	126	109	0.86	115	48	0.41
Job	111	94	0.84	358	76	0.21
Hobby	200	137	0.68	1162	250	0.21

TABLE 13. IAA score based on Cohn's Kappa and F-measure on 20 dialogues of the second round of gamification. The final score is the average of pairwise scores.

Cohn's Kappa on all tokens	81.0
Cohn's Kappa on annotated tokens	69.0
F-measure	67.5
Weighted F-measure	73.2

also helps us to understand the size of data that is needed to achieve an acceptable performance in PerInfEx. As an extrinsic evaluation, we implement and select a baseline for NLU module which is one of data-hungry units of the system and a key component in our pipeline. In the following sections, we will report evaluation results according to our NLU performance.

A. ASSESSMENT OF BASE DATASETS

First of all, to select the most suitable baseline, we investigate two main methods which are well known baselines: Joint BERT [24] and Stack-Propagation [25]. We inspect which one suits our specific dataset to assess the quality of input sources. Joint BERT is based on the pre-trained BERT Language model [26]. It passes the final hidden states of the input tokens to a SoftMax layer and classify them over predefined slot types while for intent classification, the hidden state of [CLS] is passed to this layer. Also, objective function is calculated by maximizing the joint conditional probability of both intent classification and slot filling. In Stack-Propagation method, a self-attentive BiLSTM is used as the shared encoder for both intent classification and slot filling tasks where intent detection is performed in token-level and the label of each token is then concatenated as input feature for slot filling task. To find which model is the best and also evaluate data quality, we train them both on data in the first category, manually translated dialogues and conversations in the first round of gamification. We refer them as base datasets through the rest of the paper. This train lasts for 50 epochs with maximum token length of 65 and for Joint BERT, we use ParsBERT [27] as the pretrained Persian language model. The performance of two models is compared in table 14. It should be noted that data in the second round of gamification is separated for further experiments in section VI-D, where the result of NLU assessment on this subset will be reported.

TABLE 14. Comparing the performance of Joint BERT and Stack-propagation.

	Intent accuracy	Slot F1	Exact match
Joint BERT	**84.81**	**64.55**	**48.1**
Stack-Propagation	63.29	40.26	25.31

In table 14, exact match means the percentage of cases where both slots and intent are correctly assigned. As you can see, due to the limited size of data, using a pretrained language model in Joint BERT significantly boosts the results, making it superior to Stack-Propagation by 21.52% in intent accuracy, 24.39% in slot-F1 and 22.79% in exact-match. Therefore, we consider Joint BERT as our NLU baseline in the rest of evaluation experiments. In our research, we regard intent accuracy combined with slot-F1 as the most critical factors. We aggregate these factors by adding them as criteria to identify the best-performing model on the evaluation set and at the end of each training session, the model deemed best according to these criteria will be loaded and tested on the test sets.

B. ASSESSMENT OF SEMI-AUTOMATIC AUGMENTATION METHOD

In order to assess the efficiency and the quality of data generated by semi-automatic augmentation method (describe in section IV-F1), we applied it to the three most frequent slot types in base datasets. These slots contain hobby (with 86 entities), name (with 67 entities) and occupation (with 44 entities) which encompasses 318 utterances with total tokens size of 8,158. After augmentation, method results in generating 654 utterances with 16,301 tokens. Table 15 shows distribution of slot labels in the base datasets and in the augmented set.

We then train NLU baseline on base set and its augmented version during 20 epochs with maximum token limitation of 65 and batch size of 32 which results are compared in table 16. As you can see, semi-automatic augmentation clearly improves the module's performance with almost 16%

TABLE 15. Comparing distribution of slot labels in the base set & the augmented set.

	Base set	Augmented set
B-Name	67	169
B-Job	44	167
B-Hobby	86	211
I-Name	1	1
I-Job	25	75
I-Hobby	59	229
O	4041	7855

improvement in F1 of slot, 12.8% improvement in intent accuracy and 17.68% in exact match.

TABLE 16. Investigating the effect of semi-automatic augmentation by training Joint BERT on both base datasets and its augmented version.

	Intent Accuracy	Exact match	Slot F1	Slot Precision	Slot Recall
base datasets	81.01	62.02	68.81	68.08	69.56
Semi-automatic augmented datasets	**93.8**	**79.7**	**84.8**	**82.5**	**83.7**

C. ASSESSMENT OF FULL-AUTOMATIC AUGMENTATION METHOD

To assess the effectiveness of the proposed full-automatic augmentation method (described in section subsubsec:auto-aug), we conduct distinct sets of experiments at both the slot and intent levels. Also, in each level, prior to applying this method to our own datasets, we evaluate the performance on established benchmarks, specifically SNIPS [7] and ATIS [6] (introduced in section II). These two sets are the primary benchmarks for NLU and SLU systems. Therefore, we can gauge the method's efficacy independent of any potential biases towards our own specific data. We also perform the same experiments on the Farsi-translated section of the Massive [15] corpus to gain a better understanding of model's performance on Persian.

1) EXPERIMENTS ON AUGMENTATION IN SLOT LEVEL

For slot level augmentation, we employ few-shot learning with k = 10 as a data-scarce scenario which means that for each slot type, 10 utterances are included in the training set. However, due to the imbalance distribution of these types and the possibility of an utterance having multiple different types, some of them may have more than 10 samples.

With this setting, in ATIS dataset, we utilize only 9.6% of the training set which consists of 434 utterances. In general, it contains 4,478 samples in training, 500 in evaluation and 893 in test section with a limited vocabulary size of 950. Also, it incorporates 79 different slot types and 21 intents. An examination of slot type distribution in the selected set reveals that 23 types are scarce with less than 10 samples even in the entire training set. Meanwhile, 'fromloc.city_name' and 'toloc.city_name', as the most frequent ones, have been repeated nearly 300 times in our selected set (refer to the supplementary material for details).

So, Firstly, we fine-tune conditional BERT with our selected set which includes 79 slot types from ATIS dataset over 10 epochs with batch-size of 8. In the generation step, the model produces 83 utterances with distinct new values that are added to the original selected set resulting in 517 utterances. Finally, we train Joint BERT on this augmented data with a batch-size of 64 for 10 epochs. The results for both the augmented and the original sets are presented in Table 17. Additionally, we compare our method with simple slot replacement as a baseline and evaluate the effect of input training size with k = 10 (9.6% of the entire set) and k = 30 (25% of the set).

With k = 10, both augmentation methods (ours or simple replacement) improve the NLU results and they can considerably boost slot detection as the main purpose of this level. Although, slot replacement is marginally better than ours with 0.7% superiority in intent accuracy and 0.8% in slot-F1 but compared to the original set, they both improve exact match nearly 6%. Exact match refers to the percentage of utterances in which both the intent and slots are correctly detected.

As an ablation, we investigate distinct values in new utterances to check the quality of generated or replaced values. It reveals that the model can successfully generate new values which are not present in the training data (e.g., 'round way' instead of 'round trip', 'Delta Canada' instead of 'Delta Airlines'). Also, wrong cases are investigated which indicates that some of them origins from repeating one predicted word for all of sub-tokens of the original word (e.g., 'Denver Denver Denver' instead of 'St. louis') or some of multi-word expressions that are not completely learned by the model (e.g., 'no than than' instead of 'no late than'). Moreover, some values in other types are wrongly generated for another type (e.g., 'American airline' instead of 'cheapest' for 'cost_relative' slot type). Some of these errors are related to limited number of 10 samples per slot type with considerably large set of 79 types in the training set. Therefore, by increasing k to 30, our method shows 1.3% superiority in intent classification, 0.5% in slot filling and 1.1% in exact match compared to the simple slot replacement method.

One of the other important factors in the quality of the results is the restricted diversity of values for some of ATIS types. Inspecting the entire training set reveals that 58% of slots (46 slots) have less than 10 distinct values and even 21 of them are filled with just one value. Training model and even using slot replacement with this low diversity is hard and will drop the quality of augmented slot types.

We also repeat the same experiments on SNIPS dataset with K = 10 which results in 244 utterances or just 1.8% of the entire training set. In overall, the whole dataset contains 13,084 training utterances and 700 utterances in each section of evaluation and test. It also has 39 slot types and 7 different intents. As opposed to ATIS, the whole dataset of SNIPS is a more balanced in term of intents' distribution while the size of its training set is also nearly three times larger and due

TABLE 17. Comparing the slot-level performance of our augmentation method with simple replacement on different training sizes of ATIS dataset.

	Original data	K=10		K=30	
		Our Augmented data	Simple slot replacement	Our Augmented data	Simple slot replacement
Intent accuracy	81.4	82.3	**83**	**92**	90.7
Slot F1	68.7	73	**73.8**	**92.3**	91.8
Slot precision	69.6	74.9	**76.3**	**91.6**	91.1
Slot recall	67.9	71.1	**71.4**	**92.9**	92.6
Exact match	33.1	39	**39.4**	**75.1**	74

to the diversity in the topics of the queries, it has a richer vocabulary of 12,134 tokens. This balanced feature of SNIPS is also present in our selected data with K = 10. There is no scarce slot types and all of them have at least 10 samples while restaurant type and time range as the most frequent types have 45 and 41 samples. Table 18 shows the result of training Joint BERT on 329 utterances consisted of 244 original and 85 augmented new samples.

TABLE 18. Comparing the slot-level performance of our augmentation method with simple replacement on different training sizes of SNIPS.

	Original data	K=10	
		Our Augmented data	Simple slot replacement
Intent accuracy	89.7	**91.1**	90.8
Slot F1	65.3	**72.7**	71.3
Slot precision	60.4	**68.6**	66.4
Slot recall	71	**77.4**	76.9
Exact match	34	**48.4**	46.4

Here, our augmentation method performs 0.3% better than simple slot replacement in intent accuracy, 1.4% better in slot-F1 and also 2% better in exact match. Furthermore, it leads to 1.4% improvement in intent accuracy of the original data, 7.4% enhancement in slot-F1 and 14.4% improvement in exact match. Investigating generated utterances shows that our method can successfully generate new values not presented in training set such as 'movie theatre' instead of 'movie house'. Most of these values belongs to named entity types such as object name, name of the playlist or movie names (e.g., 'Eddie Ledoux' instead of 'Chris Ledoux' for 'artist_type'). In this context, we welcome the creativity of the pretrained model, especially for types with an infinite range can benefit from diversity of the generated values compared to repeating limited samples in simple slot replacement method. Additionally, fine-tuning the model on SNIPS is easier than ATIS due to the smaller number of slot types.

In the next experiment, we probe our method's performance on Persian language using Fa-Massive (Farsi section of Massive) dataset. In overall, the whole dataset contains 11,514 training, 2,033 evaluation and 2,974 test set with vocabulary size of 6,687. Again, K is set to 10 where 485 utterances or 4.2% of the whole training set is selected. We also use ParseBERT [27] as the pretrained language model in conditional BERT. After fun-tuning, 481 utterances with the augmented slots are added to the selected set and used to train Joint BERT model. The results are shown in table 19.

When K = 10, the results of training on 966 utterances in augmented data shows that our method is even worse than the original set. Inspecting the process of model's fine-tuning reveals that the model is underfitted due to the large set of 56 slot types and the limited size of 10 samples per slot. In this situation, simple slot replacement outperforms both ours and the original set with 1.1% superiority in intent accuracy, 0.5 in slot-F1 and 0.6% in exact match. When we set K to 50, sufficient data can be provided which results in 2.6% improvement of slot-F1 (2.9% in slot precision and 1.8% in recall) and 2% in exact match compared to replacement method.

Investigating 556 values in distinct generated utterances shows that 31% of them or 174 cases are newly generated based on the model's pretrained Language Model (LM) and they are not present in the training set; such as 'دو روز دیگر' ("Do ruz digar", *two days later*) instead of 'دو ساعت دیگر' ("Do sā'at digar", *two hours later*) for time slot, 'لیست آهنگ های شاد' ("list-e āhang-hā-ye shad", *list of happy songs*) instead of 'لیست پخش قبل بازي' ("list-e pakhsh-e ghabl-e bāzi", *list of songs to be played before the game*) for 'playlist_name' type, 'غریب آشنا' ("gharīb-e-āshnā", *Garib Ashena (a Persian song name)*) instead of 'آهنگ آخر' ("Āhang-e-ākhar", *last song*) for 'song_name' type and 'خط لوله بندر عباس' ("Khat-e Loleh-ye Bandar-e Abbas", *pipeline of BandarAbbas*) instead of 'خط لوله پارسیان' ("Khat-e Loleh-ye Pārsiān", *pipeline of Parsian*) for 'news_topic'. In these cases, transliteration is enclosed in double quote and translation is in italic.

Finally, we apply this method on our base datasets including utterances generated through semi-automatic augmentation. It counts to 529 utterances. Regarding the three most frequent slot types, they contain 169 cases for name, 167 samples for occupation and 195 cases for hobby. These cases contain a diverse range of values compared to the previous datasets where slot values are more repetitive or enumerable. To show it statistically, 'Name' slot has 147 unique values. It means that just nearly 13% of cases contains repeated names while 86% of the cases introduce new names. The situation is the same for occupation with 150 unique values where nearly 90% of them contain new job titles while 'Hobby' is the most repetitive type with 31% of repeated cases and 133 unique values. This diversity can bring some difficulty for model to learn the true range of slots

TABLE 19. Comparing the slot-level performance of our augmentation method with simple replacement on different training sizes of Fa-Massive corpus.

	Original data	K=10		K=50	
		Our Augmented data	Simple slot replacement	Our Augmented data	Simple slot replacement
Intent accuracy	51.3	50.6	**52.4**	**64.3**	64
Slot F1	49.2	42.8	**49.7**	**58.7**	56.1
Slot precision	42.1	36.7	**42.3**	**50.6**	47.7
Slot recall	59.2	51.4	**60.2**	**69.8**	68
Exact match	26.7	25.1	**27.3**	**38.1**	36.1

specially for hobby which has a lot of multiword expressions in a dynamic range from 'biking' to 'eating tea in an open space'.

Table 20 shows the result of augmentation in slot-level on our base dataset. The NLU baseline is trained during 25 epochs with maximum token length of 65 and batch size of 128. As you can see, our method increases F1 of slot 0.2% compared to the original set with 0.7% improvement in slot recall and almost the same precision while simple slot replacement is 0.4% better than ours in slot-F1 with 1.6% improve in slot precision and 0.8% decrease in slot recall while intent accuracy is reduced 1.2%. It should be regarded that our method increases slot recall 1.5% compared to simple slot replacement due to generating brand-new values. Further investigation in augmented data shows 142 newly generated samples such as '...من کار آزاد دارم...' (*I am a freelancer*) instead of '...من خیاطي دارم...' (*I am a tailor*) or 'من دستیار کارگردان هستم' (*I am an assistant director*) instead of 'من دکتر مغز هستم' (*I am a brain surgeon*). Among these samples, 78 slot values are brand-new and not present in our training set. For instance, generating 'محمدعلي' ("Muhammad 'Ali") instead of 'اميرعلي' ("Amīr 'Alī") for name slot or 'آتش بان' (*fire guardian*) instead of 'سوزن بان' (*railroad switchman*) for occupation slot as correct values that haven't been seen in the training set before. However, some of these generations are wrongly composed due to repeating a token or sub-token in multiword expressions such as 'تلفني گپ گپ گپ بزنم' (*talking talking talking on phone*) instead of 'تلفني با دوستم گپ بزنم' (*talking with my friend on phone*). It seems that when we force model to learn long sequence of tokens, it tries to remember a part of the expression and produce suitable tokens to fill the extra gaps in the context but when masked word is just one token, the ability of model to replace a context-aware new word can be increased such as inserting 'بازي' (*playing*) instead of 'نقاشي' (*drawing*) in the following utterance: 'من بیشتر از نقاشي لذت میبرم اما اغلب به موسیقي هم گوش مي دهم. شما؟' (*I enjoy drawing more, but I often listen to music too. How about you?*). To remedy this problem, we can ask model to generate one up to multiple masked tokens and then decide among these generated groups which is the best or we can design a model to learn expressions in the utterance and insert them instead of inserting single token. With all of these restrictions, model can still produce context-aware expressions such as 'فك کنم چون عاشق خرید رفتن هستم بتونم طراح مد عالي بشم.' (*I think because I love shopping, I can become an excellent fashion designer*) where it replaces original slot value of 'خرید رفتن' (*shopping*) with even a more suitable expression of 'لباس خریدن' (*shopping for clothes*) or in the following sentence, model generates name value of 'دلربا' (a specific name means attractive) with 'زیبا' (a specific name means beautiful) as a good replacement according to the context: 'من دلربا هستم البته اسم این هست. شما رو چي صدا کنم ؟' (*I am Delroba but it is just my name. what should I call you?*). In contraction, simple slot replacement produces blind replacements such as substituting 'مشاور املاك' (*real estate consultant*) in the following sentence: 'به عنوان مشاور املاك، کتاب خواندن تاثیر زیادي در کارم دارد.' (*As a real estate consultant, reading books has a significant impact on my work*). It should be considered that some of these brand-new expressions are compositionally correct but semantically wrong such as 'محمدزهرا' ("Muhammad Zahra") for name slot where the first part of the name is masculine while the second part is feminine or 'مهندس مغز' (*brain engineer*) for occupation slot where job title of engineer for a brain surgeon is awkward. Finally, in the last column of table 20, in order to take advantage of both methods, we combine them where slot-F1 has been increased 1.2% compared to the original set and 1% compared to the previous augmentations.

2) EXPERIMENTS ON AUGMENTATION IN INTENT LEVEL

In the next step, we evaluate the model performance in augmenting intent level. Before appliance on ATIS, we inspect intent distribution in its main training set. The exploration shows that 21 intents in ATIS are scattered extremely imbalanced, with 75% of them (3309 utterances) being a flight search while more than half of the others (14 remaining intents) have less than 1% contribution. Figure 9 shows the graphical representation of this distribution.

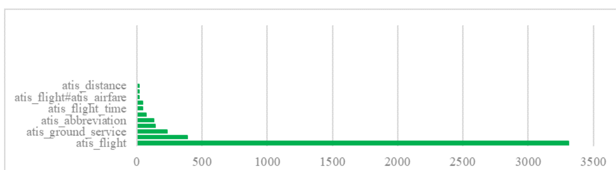

FIGURE 9. Distribution of intents in ATIS dataset.

We start our experiments by producing intent patterns described in section IV-F2. In ATIS, 4478 utterances in

TABLE 20. Slot-level performance of our augmentation method on our own dataset and its comparison with simple replacement and their combination.

	K=All of dataset			
	Original data	Our Augmented data	Simple slot replacement	Our Augmentation + Simple slot replacement
Intent accuracy	93.8	93.2	92.6	92.6
Slot F1	84.8	85.0	85.4	86
Slot precision	82.5	82.0	84.1	83.3
Slot recall	87.4	88.1	86.6	88.8
Exact match	79.7	79.1	78.5	80.3

training set are reduced to 3181 unique patterns which means that 1297 of them (28% of the data) are redundant sentences. Once more, we employ few-shot learning by setting k to the number of patterns per intent which are picked from the patterns in the whole training set. In the first round, we set k to 20 which results in 267 utterances where 8 intents have the sufficient number of patterns in the range but the others are scarce with less samples. Also, we set the hyper-parameter of masking probability to 30% in all of the subsequent experiments based on the trial and error which means that 30% of the tokens in an input utterance will be probably masked in conditional BERT. After model fine-tunning with maximum token length of 65 and batch size of 64, 12 new patterns are generated which quality needs to be better in some cases. Table 21 shows some of them. When we increase K to 30, 80 and even use all of the patterns in the set, still no more utterances are made and the quality remains the same. So, we bypass further experiments in this level on ATIS. These results can be addressed to the limited vocabulary size of the corpus and also imbalanced intent distribution with 61% of them having even less than 20 patterns in the whole set.

Experiment on SNIPS with a wider range of vocabulary and more balanced intents is a proof of this premise. All of intents in the training set have nearly the same share of 14% in frequency. After extracting patterns in SNIPS, 7140 patterns out of 13,084 utterances are produced. When we set $k = 20$, 140 utterances are selected and used in fine-tunning where model generates 8 new samples with low quality. When we set $k = 50$, 350 utterances are included and 51 new utterances can be generated and finally setting $k = 100$ results in selecting 700 samples along with 92 newly generated utterances. We also investigate the effect of using the whole patterns where 1975 new utterances are generated. Table 22 shows some of them.

In the next step, to prob the quality of the produced intents, we fill slots in each generated pattern with random values sampled from training set and add them to the original data. Then, we feed them into Joint BERT model which results are shown in table 23.

As you can see, when $k = 20$, our method is 0.2% lower in intent accuracy compared to the original set while slot-F1 is 1.6% better due to simple replacement for newly generated utterances and when we increase samples per intent ($k = 50$), intent accuracy rises 0.7%.

Repeating these experiments on Fa-Massive, results in extracting 10,218 patterns out of 11,514 utterances in training set. Distribution of intents in this dataset before producing patters shows that 14 intents have almost 56% share of frequencies while 46 remaining intents have nearly 1% or even less frequency in the whole dataset. By setting k to 20, 1170 utterances are selected and 37 new patterns can be generated. Table 24 shows some of them.

These augmented intents are then fed into Joint BERT which results are shown in table 25.

With $k = 20$, the augmentation increases intent accuracy 1.3%. These marginal increases in Fa-massive or the other datasets can be due to the limited number of generated samples and their quality. In most generated patterns, only one word is replaced with a synonym and the others remains intact. It seems that with these settings, generation ability of the model is restricted. Even excluding slot values from masking poses more limitation. So, our model needs a degree of freedom to produce more diverse samples such as being able to remove some of words or rearranging prepositions where it's possible ('show me all flights move from Denver serving meals' vs. 'show me all flights serving meals move from Denver'). These abilities can be added as post-processes in the future work.

At the end of this step, we move on to our base dataset for applying intent level augmentation. We extract 590 patterns out of 654 utterances with 148 unique samples for occupation, 185 for 'hobby', 121 for 'name', 47 for 'family', 42 for 'residence', 30 for 'age' and less than 10 samples for the rest of intents (#marriage = 8, #education = 5, #gender = 2, #other = 2). After fine-tuning during 35 epochs with maximum token length of 65 and batch size of 128., model generates 55 new samples that some of them are shown in table 26. It should be considered that our dataset with just 9% repetitive patterns is different from datasets such as SNIPS where 45% of patterns are repeated. In these datasets, most of the times, utterances contain one sentence with a specific pattern such as 'play hell house song' or 'give this novel 5 stars' while in our dataset, there are considerable number of utterances with multiple sentences. They surround main sentence containing user's personal info such as 'I am Ali, by the way, did you see football match yesterday? Are you a fan of football at all?'. These divers surrounding contexts may bring challenges for model to find specific pattern for each intent.Table 26 shows some

TABLE 21. Sample of generated intents trained on ATIS dataset.

Original intent template	Augmented intent template	Intent
I would like the schedule of all flights from Baltimore to Dallas on Tuesday	I'like the schedule of all flights from Baltimore to Dallas on Tuesday	atis_flight_time
what kind of ground transportation is available in Denver	what kinds of ground transportation is available in Denver	atis_ground_service
what ground transportation is there in Denver	what ground transportation is available in Denver	atis_ground_service
list types of aircraft that fly between Baltimore and Dallas	list types of planes that fly between Baltimore and Dallas	atis_aircraft
show me airlines going from Baltimore going to Dallas and then continuing to Dallas on Tuesday	show all airlines going from Baltimore going to Dallas and then continuing to Dallas on Tuesday	atis_airline

TABLE 22. Sample of generated intents trained on the whole patterns in SNIPS.

Original intent template	Augmented intent template	Intent
find a Douglas theatre company showing films	find a Douglas theatre company with films	SearchScreeningEvent
I need to add a song by Westbam to the playlist 50	I want to add a song by Westbam to the playlist 50	AddToPlaylist
I'd like to watch the textbook called creatures of light and darkness	I'd like to see the textbook called creatures of light and darkness	SearchCreativeWork
add Westbam to my playlist	add Westbam to my songlist	AddToPlaylist
book me a table for 3 at a restaurant for 1 hour and 1 second from now	I need a table for 3 at a restaurant around 1 hour and 1 second from now	BookRestaurant

TABLE 23. Evaluating the intent-level performance of our augmentation method on SNIPS.

	K=20		K=50	
	Original data	Our Augmented data	Original data	Our Augmented data
Intent accuracy	**94.4**	94.2	95.1	**95.8**
Slot F1	67.2	**68.8**	81.3	**81.4**
Slot precision	63.1	**64.6**	78.2	78.2
Slot recall	72	**73.5**	84.7	**85**
Exact match	42.1	**43**	63.2	**64.2**

TABLE 24. Sample of generated intents trained on Fa-Massive with k = 20.

Original intent template	Augmented intent template	Intent
زنگ هشدار را تغییر بده تا از نه صبح زنگ بخورد *Change the alarm to ring at 9 AM.*	زنگ ساعت را تغییر بده تا از نه صبح زنگ بخورد *Change the clock to ring at 9 AM.*	alarm_set
آرام *Lower*	بلندتر *Louder*	audio_volume_up
حسابداري دونالد ترامپ چيست *What is Donald Trump's accounting?*	حسابداري دونالد ترامپ چنده *What 's Donald Trump's accounting?*	email_querycontact
ترافيك خيابان تهران چطور است *What is traffic in Tehran streets?*	ترافيك اطراف تهران چطور است *What is traffic around Tehran?*	transport_traffic
سکوت لطفا *Quiet please.*	ساکت لطفا *Be quiet please.*	audio_volume_mute
نان را از لیست کارها خارج کن *Remove bread from to-do list.*	نان را از لیست کارها حذف کن *Delete bread from to-do list.*	lists_remove
تعریف چتر چیست *What is the definition of umbrella?*	معناي چتر چیست *What is the meaning of umbrella?*	qa_definition

of generated intents after convergence of conditional BERT model. Investigation of these intents shows that in some cases, new related words or expressions are generated which completely suite the context such as the first six samples of the table. In the next three samples, there is another form of generation where synonym words are substituted in the original pattern. In the other cases, just a punctuation is changes (the tenth sample) or a complementary word is replaced with a punctuation (the eleventh & twelfth samples). Among these cases just 12 samples are constituted of repeated words where some of them could be semantically meaningful if we eliminated the repetitions. In these cases, model prefers to ignore the extra masked place by producing the same word.

TABLE 25. Evaluating the intent-level performance of our augmentation method on Fa-Massive.

	K=20	
	Original data	Our Augmented data
Intent accuracy	68.1	69.4
Slot F1	55.5	55.9
Slot precision	52	53.1
Slot recall	59.6	59.1
Exact match	38.2	39

Again, in order to evaluate the performance of augmented intents, we train Joint BERT with the same hyperparameters which results are reported in table 27. It shows 0.6% improvement in intent accuracy compared to the original set.

3) EXPERIMENTS ON AUGMENTATION IN BOTH LEVELS

Finally, we conclude our augmentation by evaluating its performance on our base dataset at both levels. Subsequently, we train Joint BERT on the combination of augmented data in slot and intent level during 35 epochs with batch size of 128. The corresponding results are presented in the second column of table 28. Additionally, we incorporate simple slot replacement into our method, reported in the last column of the table. It surpasses our method in F1 of slot with 2.9% improvements while intent accuracy is decreased 1.2%. It is noteworthy that simple replacement just augments slots and the augmentation is blind by replacing random values which may not suite the context or may not be genuine but they are surely a correct value in that specific slot type. On the other hand, our method can augment both slots and intents and also it can generate context-aware and brand-new values. Therefore, combining them both in the last column results in 0.6% increase in intent accuracy and 1.9% in F1 of slot compared to the original set.

These results pertain to our three frequent and manually augmented slot types to assess and prove the efficiency of this augmentation method in the first hand. However, next, to gain a better understanding of our NLU's performance on all slot types and determine the necessary data size for a reasonable performance, we need to apply it on all types where data scarcity becomes even bolder. Apart from occupation with 167 samples, hobby with 195 samples and name with 169 samples, the other 6 types are less frequent with only 46 samples for residence, 30 samples for siblings, 25 for age, 20 for number of children, 18 for marital status and 17 samples for gender. Augmenting all of these 9 types in slot level results in 101 newly generated values among which 51 are brand-new values such as 'my 2 sisters' instead of 'my 2 brothers' for number of siblings or 'رادا' ("Rādā") instead of 'سونيا' ("Sūniyā) for name slot. So, we trained Joint BERT model on 755 utterances in general which results are reported in the second column of table 29. The fourth column shows the result of simple slot replacement on the same 755 utterances for comparison. As you can see, improvement of slot-F1 in our method is trivial (0.05%) while its recall has been increased 1.15% as expected and observed in the previous experiments. Simple slot replacement shows nearly 1% reduction in slot-F1 and exact match with roughly 7% increase in slot precision and 9% decrease in slot recall. An examination of augmented utterances of simple replacement reveals that diverse slot values, which may be suitable for a specific context but need to be adapted in a different context, can introduce complexity and ambiguity into randomly generated samples. For instance, 'من ديوانه قدم ميزنم ام به خصوص در هواي خنك.' (*I insanely love walking especially in chilly weather.*) where hobby value of 'قدم مي زنم' (*I am walking*) is a conjugated form of more general value 'قدم زدن' (*walking*) and it needs to be adjusted in the new context. Even converting these values into a more general form would not be helpful, as there are situations where they need to be conjugated in the opposite manner. These cases are not produced in our method which is based on context-aware BERT. To test the effect of these samples, we manually corrected wrong conjugations and report its results in the fifth column of the table. Slot-F1 has been improved around 1% with this correction but it is still marginally lower than our augmentation method. In the next phase, we augment intents where the number of patterns for all intents except from three augmented ones is the same as the total number of intents. This highlights the characteristic of our dataset where no distinct pattern set can be defined for these intents. The statics of these patterns has been reported in the previous experiment. The main difference of current experiment with the previous one is a more varied limited set of slot values prohibited from being masked in the train phase due to considering all slot types. After convergence, augmented model generates 24 new intent samples. Training Joint BERT in this level results in the same intent accuracy (third column) and finally integrating our method in both levels with simple slot replacement (last column) improves intent accuracy nearly 2% but decreases slot-F1 almost 3%.

D. ASSESSMENT OF GENERATED DIALOGUES

To assess the impact of adding generated dialogues on NLU performance, we consider the last trained version as our baseline, with the performance reported in the first column of table 29. We compare it against retrained module by including generated dialogues during 35 epochs with batch size of 128 which results in intent accuracy = 89.0, slot-F1 = 82.09 and exact-match = 77.46 but it should be noted that the distribution of slot types and topics in the two underlying sets is completely different and imbalanced. For a fair comparison, we need the same test set. Therefore, we employ data in the second round of gamification which is excluded from both models. The results are presented in the first and second rows of table 30.

As you can see, these additional dialogues,in the second row, have increased slot-F1 with 9.28% and improved intent accuracy and exact match drastically with 15.03%

TABLE 26. Some of generated intents trained on our base dataset.

Original intent template	Augmented intent template	Intent	#
نه اینکه من اهمیت بدهم دیگران چي فکر مي کنند. من مجسله سازم اما نامزدم تو شهرداري کار میکنه *Not that I care about what others think. I am a sculptor, but my partner works in city council.*	نه اینکه من اهمیت بدهم دیگران چي فکر مي کنند. من مجسله سازم اما نامزدم تو ناسا کار میکنه *Not that I care about what others think. I am a sculptor, but my partner works in NASA.*	Job	1
من اهل گیلان هستم. *I am from Gilan.*	من اهل سیستان هستم. *I am from Sistan.*	Residence	2
آره مشکلي نیست. من روشنا هستم. اهل کجا هستي؟ *Yeah, no problem. I am Roshana. Where are you from?*	آره مشکلي نیست. من روشنا هستم. شما چي؟ *Yeah, no problem. I am Roshana. What about you?*	Name	3
آره تو هم مراقب خودت باش پسرم ، من هم یه پسر دارم از شما یکم بزرگتر *Yeah, you take care of yourself too, my boy. I also have a son a bit older than you.*	آره تو هم مراقب خودت باش پسرم ، من فقط یه پسر دارم از شما یکم بزرگتر *Yeah, you take care of yourself too, my boy. I only have a son a bit older than you.*	Family	4
من بیشتر دوست دارم به کوهنوردي بروم . *I'd like mountain climbing most.*	من هم دوست دارم به کوهنوردي بروم . *I'd like mountain climbing too.*	Hobby	5
من مخلص شما روشنا ام . چه مي کني اصلان گرامي ؟ *I am Roshana. What are you doing, dear Aslan?*	من مخلص شما روشنا ام . چه مي کني دوست گرامي ؟ *I am Roshana. What are you doing, dear friend?*	Name	6
من توي مدرسه به عنوان مجسمه ساز کار مي کنم . *I work as sculptor in school.*	من توي مدرسه به اسم مجسمه ساز کار مي کنم . *I work with title of sculptor in school.*	Job	7
من روشنا هستم . فرهود جان چه مي کني ؟ *I am Roshana. What are you doing, dear Farhood?*	من روشنا ام . فرهود جان چه مي کني ؟ *I 'm Roshana. What are you doing, dear Farhood?*	Name	8
من واقعا از کوهنوردي لذت مي برم ، سرگرمي شما چیه ؟ *I really enjoy mountain climbing. What is your hobby?*	من واقعا از کوهنوردي لذت مي برم ، سرگرمي شما چیست ؟ *I really enjoy mountain climbing. What about yours?*	Hobby	9
حتما ، من روشنا هستم *Of course , I am Roshana.*	حتما . من روشنا هستم *Of course . I am Roshana.*	Name	10
روشنا. *Roshana .*	روشنا هستم *I'm Roshana*	Name	11
بله البته ، مجسمه ساز هستم . *Yes of course, I'm sculptor.*	بله البته من مجسمه ساز هستم . *Yes of course I am sculptor.*	Job	12
نه حقیقتش، من مجسمه سازي میکنم ، کار و خستگي زیاد، درآمد کم . *To be honest, I sculpt. It is a lot of work and fatigue, with little income.*	نه حقیقتش، من مجسمه سازي میکنم ، کار خستگي خستگي زیاد، درآمد کم . *To be honest, I sculpt. It is a lot of work work fatigue, with little income.*	Job	13

TABLE 27. Evaluating the intent-level performance of our augmentation method on our base dataset.

	K=All of dataset	
	Original data	Our Augmented data
Intent accuracy	93.8	**94.4**
Slot F1	**84.8**	84.5
Slot precision	82.5	81.9
Slot recall	87.4	87.4
Exact match	79.7	79.1

and 16.64% improvements. We also investigate model performance in entity level which results are reported in table 31.

Incorporating generated data predominantly enhances the detection of 'age' type by 27.13% and 'hobby' by 14.75%, while reducing 'gender' detection by 5%. Although the values for the gender slot are limited in diversity, it remains a complex type. Terms such as 'boy', 'girl', 'man', or 'woman' can be erroneously confused with values related to the number of children, number of siblings, and even marital status. Therefore, more samples are needed to cover this type while the generated dialogues only contribute 9 cases and it is the least addressed topic in this dataset. Moreover, it is worth noting that improvements in 'residence' are modest at 2.06% compared to enhancements in other slot types. Despite being among the top five topics in the dialogues, its limited diversity (only 10% opposed to 78% in previous datasets) results in a relatively minor contribution to improvements. Conversely, marital status with the least mentioned slots, demonstrates a significant contribution of 10.32% due to its closed set of values.

User: سلام، خوبی؟ (Hi, how are you?)
Bot: من خوبم ، شما چطور ؟ (I'm fine, how about you?)
User: من هم خوبم (I'm good too.)
Bot: من هم همینطور ، شما چه کار می کنید ؟ (That's great. What do you do?)
User: دارم با تو حرف میزنم. تو چیکار می کنی؟ (I'm chatting with you. What about you?)
Bot: من توی یک فروشگاه کار می کنم . شما چه کار می کنید ؟ (I work in a store. What do you do?)
User: من مهندس کامپیوتر هستم (I'm a computer engineer.)
Bot: خیلی خوب ، آیا از کار لذت می برید ؟ (That's nice. Do you enjoy your work?)
User: بله شما چی؟ (Yes, I do. How about you?)
Bot: من از بازی های ویدیویی لذت می برم . شما ؟ (I enjoy playing video games. How about you?)
User: نه من خیلی اهل بازی ویدیویی نیستم (No, I'm not really into video games.)
Bot: آیا شما به سفر می روید ؟ (Do you like traveling?)
User: بله زیاد. من به بیشتر شهرهای ایران سفر کردم (Yes, I do. I've traveled to many cities in Iran.)
Bot: من هم همینطور . من هم خیلی سفر می کنم (I do too. I travel a lot as well.)
User: تو اصلا اهل کجا هستی؟ (Where are you originally from?)
Bot: من اهل کالیفرنیا هستم (I'm from California.)
User: به چه شهرهایی سفر کردی؟ (Which cities or countries have you traveled to?)
Bot: من هم همینطور . من هم به صورت پاره وقت کار می کنم . (I do the same. I work part-time.)
User: نه به چه شهر یا کشوری سفر می روی؟ (No, I mean, which city or country have you traveled to?)
Bot: من از سفر کردن خیلی خوشم می آید (I really enjoy traveling.)
User: تو ازدواج کردی؟ (Are you married?)
Bot: نه ، من ازدواج نکردم . (No, I'm not married.)
User: بچه داری؟ (Do you have children?)
Bot: بله ، دو تا بچه دارم . (Yes, I have two kids.)

FIGURE 10. Sample of conversation with NLG baseline trained in the second phase.

TABLE 28. Evaluating the performance of our augmentation method in both intent & slot level on our base dataset.

	K=All of dataset			
	Original data	Our Intent & slot Augmentation	Simple slot replacement	Our Intent & slot Augmentation + Simple slot replacement
Intent accuracy	93.8	**94.4**	92.6	**94.4**
Slot F1	84.8	83.9	85.4	**86.7**
Slot precision	82.5	80.8	**84.1**	84
Slot recall	87.4	87.4	86.6	**89.6**
Exact match	**79.7**	78.5	78.5	79.1

TABLE 29. Comparing performance of our method in both levels, trained on all slot types.

	K=All of dataset					
	Original data	Our slot Augmentation	Our intent Augmentation	Simple slot replacement	Corrected simple slot replacement	Our Intent & slot Augmentation + Corrected Simple slot replacement
Intent accuracy	93.25	91.41	93.25	93.25	92.63	**95.09**
Slot F1	76.35	**76.4**	74.92	75.47	76.01	73.41
Slot precision	75.7	74.72	**83.68**	83.33	82.99	81.69
Slot recall	77.01	**78.16**	67.81	68.96	70.11	66.66
Exact match	**68.09**	**68.09**	62.57	66.25	65.03	63.8

TABLE 30. Investigating the effect of generated data and complementary set by comparing baseline model against retrained Joint BERT testing on data in the second round of gamification.

	Intent Accuracy	Exact match	Slot F1	Slot Precision	Slot Recall
NLU baseline	55.31	29.94	46.23	38.30	58.31
Retrained NLU (set_{gen})	**70.34**	46.58	55.51	51.18	60.65
Retrained NLU (set_{gen}+ set_{comp})	69.68	**48.09**	**59.42**	**53.24**	**67.21**

Considering 'gender' as the least frequent topic and slot type and also the least diversified slot of 'residence' motivate us to collect complementary generation set which performance is investigated in this section. Retraining the last version of JointBERT (second row of table 30) by addition of complementary set to the training process results in intent accuracy = 88.64, slot-F1 = 83.68 and exact-match = 78.2. To fairly compare it against the two previous NLU versions, it is also tested on data in the second round of gamification and results are reported in the third row of table 30. As you can see, it shows drastic improvements of 13.19% in slot-F1 and 18.15% in exact match with 14.37% increase in intent accuracy. This represents roughly 4% improvement in slot-F1 compared to including just generation set (second row), with 6.56% increase in slot-recall and 2.06% in slot-precision. However, intent accuracy experienced a slight decrease of nearly 0.5%. The noticeable increase in performance results solely from the addition of 31 dialogues generated through profile-based prompts, limited to specific slots and topics. It demonstrates the efficiency of the method used for complementary set, which combines LLM generation with the injection of diversified values into dynamic prompts.

VII. DATA ASSESSMENT THROUGH NLG MODULE

In addition to NLU module, Natural Language Generating (NLG) is another data hungry unit of the system. It needs a considerable amount of data for training which also should be aligned with our system's specifics. Therefore, we implement a baseline using attention-based sequence-to-sequence network [28] to evaluate the size and efficiency of collected datasets. In the initial step, we train it using data in the first category, gamification dialogues and semi-automatically augmented data. Generated dialogues are excluded because they violate the prominent rule of our system to ask the least number of direct questions. We split them into 270 dialogues with 2,540 utterances in train and evaluation sets while 31 dialogues with 304 utterances have been separated for test set. The first column of the table 32 reports the results in terms of perplexity, BLEU and BERTScore [29] for these NLU specific datasets. Then, in the next step, we train our baseline again using NLG specific datasets namely training on automatically translation of ConvAI2 and fine-tuning on paraphrased data which results are reported in the second column of the table. The train and evaluation sets in this phase contains 15,723 dialogues and 1,000 conversations in the test set. Comparing module's performance in these two columns justify our efforts in collecting NLG specific datasets with considerable size. Translation of ConvAI2 as the most aligned English corpus can help us to have a backbone for NLG module while fine-tuning on paraphrased data as a more aligned set can bring us better adjustment in the module's output. Moreover, for a fair comparison of these two datasets, we test them on data in the second round of gamification and report the results in the last row of table 32.

Results of the first column obviously shows that the baseline has been underfitted due to the limited size of data with constant generation of 'yes' or 'me too' as its generic output. Also, it should be regarded that input sources of NLU specific datasets are different and this heterogeneity negatively effect on NLG performance which is not an important factor for training NLU. On the second column, considerable amount of homogenous data drastically improved the results in term of perplexity with 880 decrease. It is worth considering that in the second training phase, NLG is primarily trained on translations of ConvAI2 with a

TABLE 31. Comparing the performance of slot classification in NLU baseline against retrained model on data in the second round of gamification.

		Entity Types								
		Age	#Children	Gender	Hobby	Job	Marital status	Name	Residence	#Siblings
NLU Baseline	Slot F1	3.63	51.92	**30.00**	43.47	41.89	59.37	59.72	56.63	56.28
	Precision	2.94	45.76	37.50	32.37	30.39	55.88	45.74	52.45	55.29
	Recall	4.76	60.00	**25.00**	66.17	**67.39**	63.33	86.00	**61.53**	**57.31**
Retrained NLU	Slot F1	**30.76**	**54.90**	25.00	**58.22**	**49.55**	**69.69**	**67.18**	**58.69**	**59.74**
	Precision	**25.80**	**49.12**	**50.00**	**51.11**	**41.79**	**63.88**	**55.12**	**67.50**	**63.88**
	Recall	**38.09**	**62.22**	16.66	**67.64**	60.86	**76.66**	86.00	51.92	56.09

TABLE 32. Comparing the performance of the NLG baseline on NLU specific and NLG specific datasets.

	NLU specific datasets			NLG specific datasets		
	Perplexity	BLEU-4	BERTScore	Perplexity	BLEU-4	BERTScore
Train	687.2	2.71e-07	0.39	15.74	0.00492	0.44
Test	1935	1.582e-09	0.13	52.17	0.0003119	0.41
second round of gamification	2320	1.317e-09	0.06	**1,440**	**7.536e-07**	**0.35**

different style compared to the gamification data. This leads to an inflated perplexity compared to the perplexity on its test set from the same corpus. However, it demonstrates notable improvements and provides a solid foundation for our NLG module. In future work, we can leverage the advantages of fine-tuning pretrained language models. Figure 10 illustrates a sample conversation with the model trained in the second phase to provide a better understanding of its performance.

VIII. CONCLUSION

Developing chatbots within a specific domain or creating engaging open-domain chatbots has become a trending topic in recent years, and the number of specialized corpora in this field has increased. However, implementing chatbots in low-resource languages like Persian remains a challenge due to a scarcity of publicly available datasets of considerable size. This research represents the first effort to collect a substantial amount of data in Persian for open-domain chatbots, specifically designed for our chatbot called PerInfEx. It is a Persian chatbot designed to extract personal information from chit-chat dialogues while asking the fewest possible direct questions. To the best of our knowledge, there is no chatbot in English or Persian with these capabilities. Additionally, our research is the first attempt to customize conditional BERT model for augmenting semantic frames and presenting comprehensive tests to evaluate its performance in this field. Moreover, we implement baselines for NLU and NLG modules to assess data quality, laying the foundation for the first open-domain chatbot in Persian.

In this work, we describe data preparation and augmentation methods used to collect data from various input sources which is also aligned with our specific requirements in PerInfEx. Initially, we collect 99 free-discussion dialogues and 74 dialogues from an English training website with Persian translations. Additionally, we manually translate 72 dialogues from the ConvAI2 corpus, which is the most resembled English corpus to our specifics. To create a more aligned dataset, we also implement and launch a chatting website to gamify the collection process, resulting in a total of 94 dialogues. To assess the quality and adequacy of these datasets, we use Joint BERT as our NLU baseline. After annotating all the collected datasets, we evaluated its performance, resulting in an F1 score of 68.7 for slot filling, accuracy of 81 for intent detection, and accuracy of 60.7 for exact-match. These results indicate that more data is needed, particularly to improve the slot filling subtask. To address this, we augment the already collected datasets using both semi-automatic and fully automatic techniques on our three most frequent slots: name, hobby, and occupation. The semi-automatic method improves NLU results by 16% improvements in slot-F1, 12.8% in intent accuracy, and 17.68% in exact match. Our fully automatic augmentation is the first attempt to employ Conditional BERT for augmenting semantic frames. After conducting comprehensive experiments to assess its efficiency, combining this method with simple slot replacement results in a 1.9% improvement in slot-F1 and 0.6% in intent accuracy. Considering the time-consuming process of the semi-automatic method and the marginal improvements of the automatic technique, which highlights data scarcity, we generate an additional 335 dialogues with 5,094 utterances using OpenAI's gpt-3.5-turbo-instruct model. Then, we annotate them using an active learning process and retrain NLU where it improves the performance by 13.19% in slot-F1, 14.37% in intent accuracy, and 18.15% in exact-match. In the next step, we assess data efficiency by training a baseline for the NLG module, another data-intensive component of the system. The baseline shows signs of underfitting, leading us to gather two datasets exclusively for this module. For this purpose, we automatically translate 15,217 dialogues from the training set of ConvAI and the entire evaluation set, consisting of 1,000 dialogues using OpenAI's gpt-3.5-turbo-instruct model. Also, we paraphrase the already collected datasets using the same model, results in 506 dialogues with 5,759 utterances. Retraining the NLG module and comparing its results with the trained version on NLU specific datasets shows drastic improvements by decreasing perplexity from 2,320 on second round of gamification data to 1,440. Finally, at the end of our data preparation process, we collected 1,180 dialogues with 16,899 utterances along with 16,217 automatically translated dialogues (213,123 utterances) which can be regarded as a valuable dataset for training open-domain chatbots in Persian. In future work, we will employ adversarial training methods to identify adversarial examples, which may consist of confusing words (such as the term 'boy' or 'girl' repeated in multiple distinct slot types). We will automatically generate these examples and incorporate them into the training process to enhance the model's robustness. Additionally, we will proceed with the full implementation of PerInfEx including NLU, NLG, and DM (Dialogue Management) components.

REFERENCES

[1] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional BERT contextual augmentation," in *Proc. 19th Int. Conf. Comput. Sci. (ICCS)*. Faro, Portugal: Springer, 2019, pp. 84–95.

[2] L. A. Suchman, *Plans and Situated Actions: The Problem of Humanmachine Communication*. Cambridge, U.K.: Cambridge Univ. Press, 1987.

[3] K. M. Colby, *Artificial Paranoia: A Computer Simulation of Paranoid Processes*, vol. 49. Amsterdam, The Netherlands: Elsevier, 2013.

[4] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, Jun. 1992, pp. 517–520.

[5] J. D. Williams, M. Henderson, A. Raux, B. Thomson, A. Black, and D. Ramachandran, "The dialog state tracking challenge series," *AI Mag.*, vol. 35, no. 4, pp. 121–124, Dec. 2014.

[6] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?" in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2010, pp. 19–24.

[7] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[8] A. Ritter, C. Cherry, and B. Dolan, "Data-driven response generation in social media," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, 2011, pp. 583–593.

[9] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" 2018, *arXiv:1801.07243*.

[10] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A manually labelled multi-turn dialogue dataset," 2017, *arXiv:1710.03957*.

[11] C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs," 2011, *arXiv:1106.3077*.

[12] E. Dinan, V. Logacheva, V. Malykh, A. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, and R. Lowe, "The second conversational intelligence challenge (ConvAI2)," in *The Competition: From Machine Learning to Intelligent Conversations (NeurIPS)*. Cham, Switzerland: Springer, 2020, pp. 187–208.

[13] F. Jabbari, H. Sameti, and M. H. Bokaei, "Unilateral semi-supervised learning of extended hidden vector state for Persian language understanding," in *Proc. 7th Int. Conf. Natural Lang. Process. Knowl. Eng.*, Nov. 2011, pp. 165–168.

[14] Z. Borhanifard, H. Basafa, S. Z. Razavi, and H. Faili, "Persian language understanding in task-oriented dialogue system for online shopping," in *Proc. 11th Int. Conf. Inf. Knowl. Technol. (IKT)*, Dec. 2020, pp. 79–84.

[15] J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh, S. Ranganath, L. Crist, M. Britan, W. Leeuwis, G. Tur, and P. Natarajan, "MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages," 2022, *arXiv:2204.08582*.

[16] M. Akbari, A. Hossein Karimi, T. Saeedi, Z. Saeidi, K. Ghezelbash, F. Shamsezat, M. Akbari, and A. Mohades, "A Persian benchmark for joint intent detection and slot filling," 2023, *arXiv:2303.00408*.

[17] F. Z. Arshia, M. A. Keyvanrad, S. S. Sadidpour, and S. M. R. Mohammadi, "PeQA: A massive Persian question-answering and chatbot dataset," in *Proc. 12th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Nov. 2022, pp. 392–397.

[18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, and A. Ray, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27730–27744.

[19] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[20] P. Das and A. K. Das, "Graph-based clustering of extracted paraphrases for labelling crime reports," *Knowl.-Based Syst.*, vol. 179, pp. 55–76, Sep. 2019.

[21] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

[22] G. Hripcsak and A. S. Rothschild, "Agreement, the f-measure, and reliability in information retrieval," *J. Amer. Med. Inform. Assoc.*, vol. 12, no. 3, pp. 296–298, 2005.

[23] L. Deleger, Q. Li, T. Lingren, M. Kaiser, K. Molnar, L. Stoutenborough, M. Kouril, K. Marsolo, and I. Solti, "Building gold standard corpora for medical natural language processing tasks," in *AMIA Annu. Symp. Proc.*, vol. 2012. Bethesda, MD, USA: American Medical Informatics Association, 2012, p. 144.

[24] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," 2019, *arXiv:1902.10909*.

[25] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," 2019, *arXiv:1909.02188*.

[26] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, p. 2.

[27] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based model for Persian language understanding," *Neural Process. Lett.*, vol. 53, no. 6, pp. 3831–3847, Dec. 2021.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3104–3112.

[29] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," 2019, *arXiv:1904.09675*.

PEGAH SAFARI received the B.S. degree in computer science from Kharazmi University, Tehran, Iran, in 2012, and the M.S. degree in artificial intelligence from Alzahra University, Tehran, in 2015. She is currently pursuing the Ph.D. degree in artificial intelligence with Shahid Beheshti University, Tehran. Her main research interests include human–computer interaction systems, chatbots, and natural language understanding.

MEHRNOUSH SHAMSFARD received the B.S. and M.Sc. degrees in computer software engineering from the Sharif University of Technology and the Ph.D. degree in computer engineering-artificial intelligence from the Amirkabir University of Technology, Tehran, Iran. Since 2004, she has been with Shahid Beheshti University, Tehran, where she is currently an Associate Professor with the Faculty of Computer Science and Engineering and the Head of the NLP Research Laboratory. Her main research interests include natural language processing, developing intelligent assistants and chatbots, knowledge engineering (ontologies and knowledge graphs), text mining, and semantic and intelligent web.