Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 14, No. 2, 2026, 183-196.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Dynamic Retrieval-Based Prompting for Cross-Lingual Dialogue Understanding in Persian

Saedeh Tahery* and Saeed Farzi

Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.

Article Info

Article History:
Received 27 July 2025
Revised 13 November 2025
Accepted 24 November 2025

DOI:10.22044/jadm.2025.16583.2785

Keywords:
Cross-lingual Adaptation, Natural Language Understanding, Persian Language, Large Language Models, ChatGPT.

**Corresponding author: saedeh.tahery@email.kntu.ac.ir (S. Tahery).*

Abstract

Dialogue understanding for low-resource languages such as Persian remains challenging due to limited annotated data, which constrains supervised training at scale. We propose a simple yet effective training-free method that combines machine translation, retrieval-based example selection, and prompting with a large language model (GPT-4o) to improve zero-shot cross-lingual performance. Given a Persian utterance translated into English, our method retrieves semantically and lexically similar English examples using a hybrid similarity function, translates them back into Persian, and constructs a few-shot prompt tailored to the input. This input-sensitive strategy enhances the quality of the examples, helping the model align more effectively with each instance. Experimental results on the Persian-ATIS dataset show that our approach improves intent detection and achieves competitive slot filling performance, outperforming state-of-the-art baselines without requiring any supervision in the target language. The modular pipeline is easy to reproduce and, in future work, can be extended to other low-resource languages, tasks, or retrieval configurations. The repository of our work is available at https://github.com/saedeht/Persian_Language_Understanding.

1. Introduction

Recent advances in natural language processing (NLP), driven by large language models (LLMs), have led to remarkable improvements in tasks such as machine translation, summarization, information extraction, and dialogue systems [1-3]. These developments have brought NLP technologies closer to real-world applications while enabling more natural human–computer interaction. Despite these advancements, the benefits are not equally distributed across languages. Although over 7,000 languages are spoken worldwide [4], nearly 50% of websites are in English. Spanish follows, but with a wide gap, accounting for only about 6% of web content (as of February 2025)[1], reflecting the unequal distribution of digital resources. Consequently, recent NLP progress has primarily favored high-resource languages like English, while low-resource languages remain underrepresented and continue to face significant challenges in developing effective language models [5].

Task-oriented dialogue is an important NLP application that helps users accomplish goals such as booking flights or setting reminders [6-8]. Natural language understanding (NLU) is a crucial component of these systems. As shown in Figure 1, NLU typically includes intent detection (ID), which identifies the user's goal from the utterance, and slot filling (SF), which assigns labels to tokens to extract relevant entities [9]. These tasks rely on

[1] https://www.statista.com/statistics/262946/most-common-languages-on-the-internet

labeled datasets, which are often unavailable for low-resource languages like Persian. Manual annotation is costly and time-consuming, limiting the deployment of robust NLU systems and highlighting the need for methods that generalize without requiring labeled data in the target language [10].

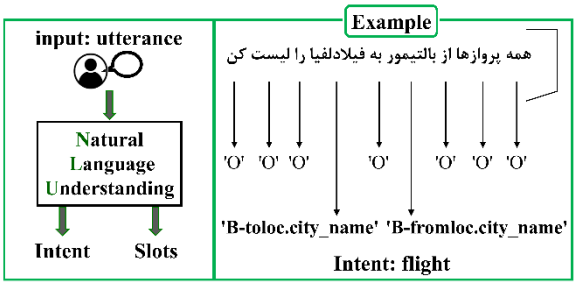

Figure 1. The Natural Language Understanding module consists of intent detection and slot filling, illustrated with an example in Persian.

To address this, studies have explored cross-lingual transfer techniques, leveraging labeled data from high-resource languages such as English to build models for low-resource languages [11]. A common approach is zero-shot cross-lingual transfer, where multilingual models trained on English are directly applied to other languages. While this yields reasonable performance, it may overlook language-specific context and cultural nuances [12].

Recent approaches have employed LLMs through prompt-based learning [13], constructing prompts that enable the model to perform tasks with zero or few informative examples, without extensive fine-tuning. Some studies further combine cross-lingual transfer with prompting to achieve both linguistic generalization and model adaptability [14].

The application of prompt engineering to cross-lingual NLU, despite promising results, remains relatively underexplored, particularly in low-resource contexts. A key factor in few-shot prompting is the selection of relevant examples, as their contextual alignment with the input greatly influences performance. This motivates us to move beyond static example selection and introduce an input-sensitive prompting approach, in which contextually appropriate examples are dynamically retrieved based on the input utterance.

In this paper, inspired by retrieval-augmented generation (RAG) [15], we propose a retrieval-based dynamic prompting approach for zero-shot cross-lingual Persian language understanding, comprising two main phases. We leverage machine translation to transfer supervision from English to Persian, a strategy widely used in cross-lingual NLP due to the robustness of current translation systems and the central role of English in multilingual resources [16, 17].

In the first phase, we perform input-sensitive example selection via retrieval in the English source space. Each Persian input is translated into English to retrieve the top-k relevant examples using a hybrid similarity function that combines contextual embeddings and lexical overlap (TF-IDF) [18] with an adjustable weight (α). This balances semantic relevance with surface-level matching, promoting the quality of selected examples. The retrieved English examples are translated back into Persian, with their intent and slot annotations preserved and automatically aligned to the source input using token-level alignment [19]. In the second phase, the retrieved examples are incorporated into a few-shot prompt and fed to an LLM (GPT-4o) to perform ID and SF on the original Persian input. By combining machine translation, cross-lingual retrieval, and prompt-based inference, our approach provides a flexible solution that dynamically adapts predictions for each input. Notably, our approach requires no training or fine-tuning in either the source or target language, making it practical and easily applicable to real-world scenarios.

Accordingly, we define three research questions to systematically investigate the proposed method throughout the paper:

- **RQ1**: How sensitive is the proposed method to the adjustable weight (α) and the number of examples in few-shot prompting (k)?
- **RQ2:** How well does the proposed method perform compared to its counterparts for zero-shot cross-lingual understanding?
- **RQ3:** What insights can be drawn from a fine-grained evaluation of the method's performance?

To answer these research questions, we evaluate our approach on the Persian-ATIS (Airline Travel Information Systems) dataset [20]. Experimental results demonstrate that our method outperforms state-of-the-art methods in ID and remains competitive in SF.

The key contributions of this work are as follows:

- Proposing a simple yet effective method with minimal computational overhead, featuring a modular pipeline that uses standard tools for easy adaptation without task- or language-specific training.
- Introducing dynamic retrieval of contextually relevant examples to improve few-shot prompting.
- Enhancing performance on Persian language understanding tasks without requiring any

labeled data in the target language.

The paper is organized as follows: Section 2 reviews task definitions and related work. Section 3 details the methodology, and Section 4 presents experiments and analysis. Finally, Section 5 concludes the paper.

2. Background and Related Work

This section first provides a formal definition of the NLU tasks (ID and SF) and then presents a summary of the most relevant related studies.

2.1. Language Understanding Tasks

The NLU module functions as the core component responsible for enabling meaningful interactions in task-oriented dialogues, typically structured within the dialogue act framework [8].

As shown in (1), given a user utterance $X = [x_1, x_2, ..., x_T]$, where T is the number of tokens, ID is framed as a classification task aiming to predict the most probable intent class $\hat{c}$ among a predefined set of intent classes C.

$$\hat{c} = \arg\max_{c \in C} p(c \mid X) \quad (1)$$

Similarly, SF is a sequence labeling task that assigns a semantic tag to each token x_t, identifying fine-grained semantic information such as departure city or date. This task is commonly modeled as provided in (2), which finds the most probable sequence of slot labels $\hat{Y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_T]$.

$$\hat{Y} = \arg\max_{Y} p(Y \mid X) \quad (2)$$

As supervised models heavily rely on annotated data, which is not feasible for low-resource languages, we resort to a zero-shot cross-lingual approach to circumvent this need for Persian. We next review existing approaches developed to cope with this limitation.

2.2. Related Work

In the pursuit of modern AI-native dialogue systems, particularly those capable of handling multilingual interactions, the design of robust language understanding modules remains a critical challenge. The availability of annotated data is an inevitable bottleneck in many NLP systems, especially in low-resource scenarios [21].

Early approaches to cross-lingual understanding relied on static multilingual embeddings such as context vectors [22, 23] or cross-lingual word alignments, which offered limited flexibility and generalization [24].

With the advent of multilingual pre-trained language models like multilingual BERT (mBERT) [25] and XLM-RoBERTa (XLM-R) [26], the field shifted toward more scalable solutions that leverage shared cross-lingual representations. For instance, Zadkamali et al. [27] conducted cross-lingual training for Persian using mBERT and XLM-R models. Several methods have been proposed to enhance these models, including translation-based approaches, data augmentation, and code-switching techniques, each aiming to mitigate the lack of supervision in target languages by enriching the input space or aligning the latent space across languages [28–31]. Qin et al. [30] proposed a data augmentation framework for fine-tuning mBERT with multilingual code-switched data, improving cross-lingual alignment without relying on bilingual training pairs. Safari and Shamsfard [32] introduced PerInfEx, a Persian chatbot designed to extract personal information through casual conversation, leveraging both manual and automated data augmentation to enhance NLU performance.

Another line of research focuses on improving cross-lingual transfer by learning language-independent representations [33, 34]. Tahery et al. [33] proposed an adversarial approach based on a Generative Adversarial Network (GAN) [35] to derive such representations from mBERT's contextual embeddings. Their model proved effective for language understanding tasks by minimizing language-specific information. However, despite strong results on Latin-based languages such as Spanish, its performance declined for typologically distant languages like Persian. Another GAN-based method was proposed in [36], which employed an encoder-decoder architecture. The task-specific fine-tuning was decoupled, and during adversarial training, the decoder was used to reconstruct input utterances. This approach mitigates language-specific information while preserving semantic content in contextual representations, improving performance for languages like Persian, although some room for further enhancement remains.

In recent years, LLMs have also opened new possibilities in this realm [37, 38]. These models, due to their extensive pre-training on diverse multilingual corpora, support two main paradigms: (i) fine-tuning the model for specific tasks in a supervised manner [39], and (ii) prompting the model to perform tasks in an inference mode, utilizing its emergent zero-shot capabilities [40]. While the former still requires labeled data and computational resources, the latter introduces new challenges, especially regarding whether such models can reliably perform structured tasks like

ID and SF with little or no task-specific supervision. Zhu et al. [41] proposed a two-stage framework leveraging ChatGPT for zero-shot language understanding, which empowers mutual verification between subtasks to improve performance. However, their experiments were exclusively conducted in English.

Yet, an important question arises regarding whether general-purpose LLMs such as ChatGPT can handle zero-shot cross-lingual NLU tasks, whether performance can be improved by providing a few demonstration examples, and how these examples should be selected.

A recent study tackled this problem by employing machine-generated examples [14]. Initially, an effective cross-lingual model was used to automatically generate labeled training instances. These examples were then filtered per domain, selecting a fixed subset based on criteria such as utterance length (favoring longer examples with fewer non-slot tokens), as well as diversity in intents and slot types. For each domain, the same static set of few-shot examples was used regardless of the input utterance, making the prompting strategy domain-aware but not input-sensitive. In contrast, this paper investigates an input-sensitive prompting strategy, which dynamically selects examples based on the input utterance and leads to improved NLU performance.

While other notable efforts have also contributed to advancing Persian language understanding [42, 43], it remains considerably underexplored compared to high-resource languages.

Continued research is required in areas such as adapting large-scale language models for Persian, improving the handling of code-switched inputs, and establishing standardized corpora [44].

3. Proposed Method

This section presents our approach to zero-shot cross-lingual Persian language understanding. In the absence of labeled data for the target language, we adopt machine translation, automatic alignment, and a retrieval-based method to identify and construct relevant few-shot examples for a given input utterance. These examples, formulated in the target language, are then used to prompt the LLM (GPT-4o) for the NLU tasks.

As illustrated in Figure 2, the proposed method consists of two main phases: dynamic example retrieval and LLM-based inference.

Given a Persian utterance, the first phase begins by translating it into English. Considering English as the source language with available annotations, we then retrieve the top-k most relevant utterances from the annotated English dataset. These examples are subsequently translated back into Persian and used to construct few-shot prompts. In the second phase, these prompts are provided to the LLM to perform the ID and SF tasks on the original Persian utterance. Further details of the proposed method are provided in the following sections.

3.1. Retrieval in the English Space

To retrieve relevant labeled English examples, we adopt a hybrid retrieval strategy that integrates both dense and sparse representations. We compute a weighted similarity score between the translated input utterance u and each candidate labeled utterance v in the English database as follows:

$$score(u,v) = \alpha.sim(emb(u), emb(v)) + (1-\alpha).sim(tfidf(u), tfid(v)) \quad (3)$$

where $emb(.)$ denotes the sentence embedding derived from a pre-trained transformer-based encoder. In our setup, we employ the BAAI/bge-large-en[2] model in inference mode, without any task-specific fine-tuning. This retrieval-oriented encoder was fine-tuned on large-scale English corpora using contrastive learning objectives and has demonstrated strong performance across dense retrieval benchmarks. Its ability to produce semantically meaningful and generalizable representations makes it well-suited for semantic similarity in our hybrid retrieval pipeline.

Moreover, $tfidf(.)$ in (3) represents the sparse vector obtained via TF-IDF weighting, sim denotes the cosine similarity [45], and $\alpha \in [0,1]$ is an adjustable weight that controls the relative contribution of each similarity component.

This hybrid retrieval strategy is motivated by the complementary strengths of dense and sparse retrieval methods. Embedding-based similarity captures high-level semantic relatedness, while TF-IDF provides token-level lexical matching, which is particularly useful in short-form texts such as user utterances, where keywords often directly signal intent or slot boundaries. TF-IDF also emphasizes rare yet informative words through higher weighting, which is especially beneficial in low-resource scenarios. We explore the effect of each similarity component in Section 4.3.1.

[2] https://huggingface.co/BAAI/bge-large-en

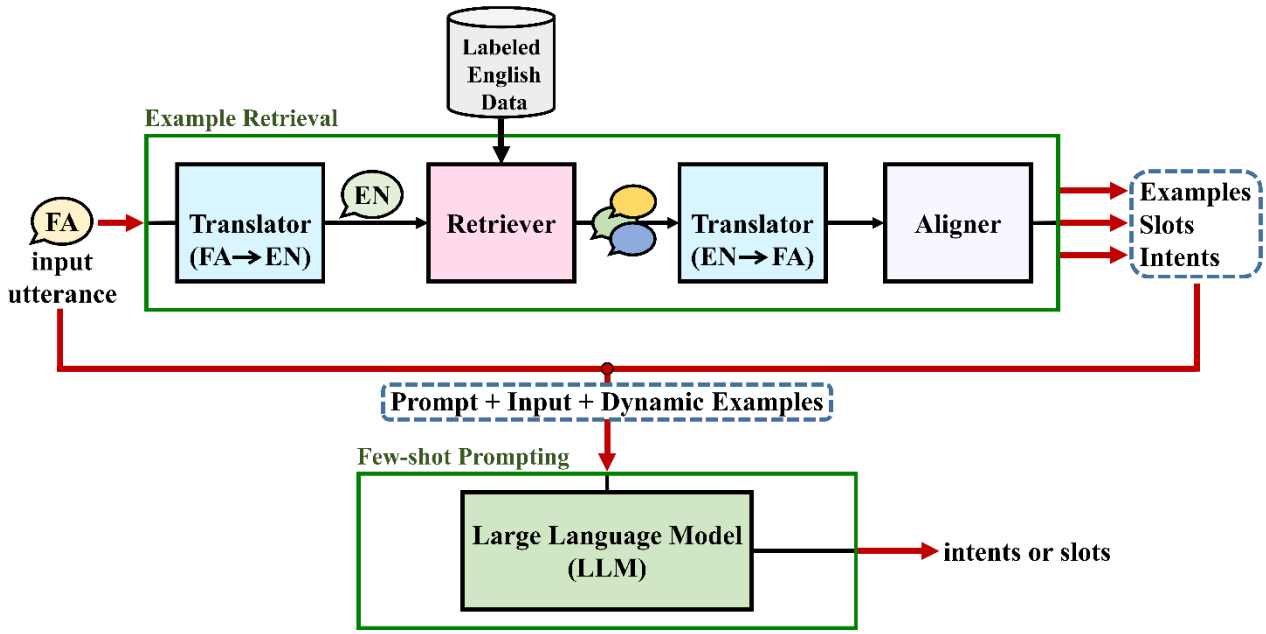

Figure 2. Overview of the proposed method, which consists of two main phases: dynamic example retrieval and LLM-based inference.

3.2. Translation and Alignment

Each retrieved English example is associated with its corresponding intent label and slot annotations. After selecting the top-*k* English utterances, we translate them back into Persian to maintain language consistency during the few-shot prompting phase. For translation in both directions, we employ Google Translate[3] due to its well-established performance across a wide range of language pairs, including low-resource ones.[4]

While intent labels can be directly reused after translation, slot filling requires additional processing. Since slot filling is a sequence labeling task, it is essential to ensure that slot labels are correctly aligned with the translated Persian tokens. To this end, we first normalize and tokenize the Persian utterances using Hazm[5] and then apply SimAlign with the IterMax algorithm [19] to perform token-level alignment between the English and Persian sequences. The slot tags are subsequently projected from the English utterances to the aligned Persian tokens based on the alignment links. For instance, Figure 3 shows a sample alignment in which the slot labels assigned to the English tokens *pittsburgh* and *philadelphia* are correctly mapped to their Persian equivalents through alignment.

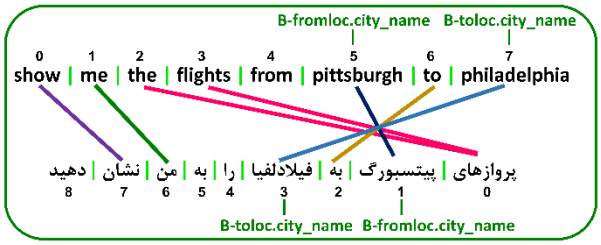

Figure 3. An example of word alignment between English and Persian utterances.

Note that we choose the tools Hazm and SimAlign based on their strong performance reported in the literature [46]. Hazm is a widely used NLP library tailored to the linguistic characteristics of Persian. It incorporates rules that handle morphological variations, pseudo-spaces, and common affixation patterns specific to the Persian language.

Furthermore, SimAlign is a lightweight alignment tool that leverages multilingual contextual embeddings to align tokens between language pairs without requiring parallel corpora, making it particularly suitable for low-resource settings like Persian.

We acknowledge that human translation and annotation of Persian utterances would offer higher reliability; however, it is impractical in our setup due to the time and cost involved, especially given

[3] Translation was performed using the Python Google Translate API (googletrans==3.1.0a0).

[4] To verify translation quality, we randomly selected 30 samples in each direction and had them manually reviewed by a human annotator. The resulting BLEU scores were 37.31 for FA→EN and 50.49 for EN→FA, which indicate satisfactory translation quality for the purposes of our cross-lingual experiments.

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

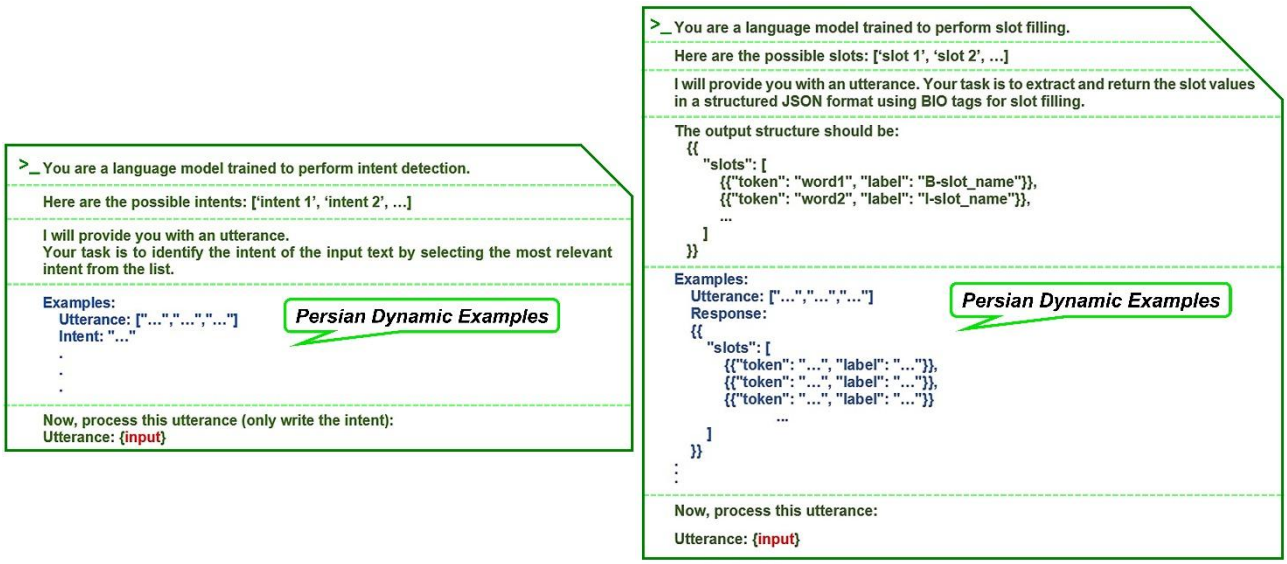

Figure 4. The prompt structure for intent detection (left) and slot filling (right) as used in [14], with the main difference being the use of input-sensitive examples.

the low-resource cross-lingual context. Therefore, we resort to automatic tools to generate such annotated examples. While these tools may introduce errors in translation and annotation, they still provide a feasible and efficient means of generating few-shot examples in the target language. In practice, we find that even imperfect projected labels are sufficient to construct effective prompts for LLM-based language understanding tasks. A few samples are provided in Appendix A.

3.3. LLM-based inference

In the second phase of our proposed method, the few-shot examples constructed during the retrieval phase are used to prompt GPT-4o to perform the ID and SF tasks. By presenting the LLM with informative Persian examples alongside the original utterance, we enable it to infer intent or slot labels without requiring any direct supervision in the target language. By informative examples, we mean Persian examples whose corresponding English versions have higher hybrid similarity scores. In other words, the higher the similarity score of an English example, the more informative the resulting Persian example is.

This prompt-based inference leverages the model's pre-trained knowledge and its capacity to generalize from limited examples, which makes it particularly effective in zero-shot cross-lingual settings.

4. Experimental Study

We begin by outlining the overall setup and prompt structure, followed by a description of the dataset and a presentation of the experimental results.

4.1. Setup

All experiments were conducted using a specific version of ChatGPT (GPT-4o-2024-08-06) via the official API[6]. All parameters, such as temperature, were left at their default settings.

To ensure a fair and consistent evaluation, we employ two structured prompts for the ID and SF tasks, similar to the setup in [14], as illustrated in Figure 4.

For ID, the model is instructed to select the most relevant intent label from a predefined list given an input utterance. The prompt begins by specifying the task description and the set of possible intents, followed by several illustrative examples demonstrating the expected mapping between utterances and intents. For SF, the prompt similarly introduces the task, defines the list of possible slots, and specifies the required output format. The examples illustrate how tokens should be labeled with the corresponding slots.

Note that although the format remains the same, the retrieved examples are not static as in [14]; rather, they are input-sensitive, vary with each test utterance, and rely on a fundamentally different example construction strategy.

4.2. Data

We conduct experiments on the Persian-ATIS dataset [20], a benchmark in the flight domain

[6] https://openai.com/

covering both intent detection and slot filling tasks. It includes 26 intent classes and 84 slot types, with utterances available in both English and Persian. The dataset contains 3,982 utterances for training, 996 for validation, and 893 for testing in each language.

For consistency with prior work and due to computational constraints, we use the same data subset as in [14], which consists of 500 utterances. This subset was selected in a stratified manner to preserve the distribution of both intent and slot labels, ensuring that the relative frequencies of each class remain similar to those in the full dataset. This approach maintains the representativeness of the evaluation set for both tasks while accommodating limited computational resources.

4.3. Evaluation Results

This section aims to answer the research questions presented in the Introduction.

4.3.1. Sensitivity Analysis

To address **RQ1**, we conduct a sensitivity analysis over two key variables: the adjustable weight (α) and the number of examples (k). We evaluate micro-averaged accuracy for ID and F1 score for SF across different values of $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$ and $k \in \{1, 3, 5\}$.

Intent Detection. As shown in Figure 5, the overall trend indicates that increasing k generally improves performance, with the best results achieved at $k = 5$ across all α values.

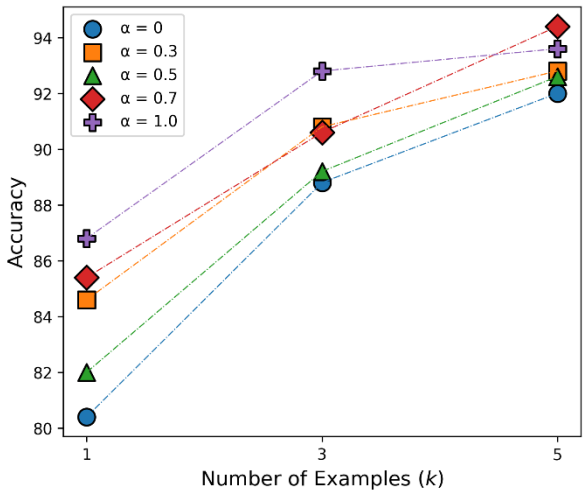

Figure 5. The impact of parameters α and k on intent detection accuracy.

The effect of α is also pronounced in this task: performance consistently improves with higher α values, peaking at 94.40% accuracy when $\alpha = 0.7$ and $k = 5$. This suggests that ID benefits more from dense similarity signals, likely because the semantic-level information captured by embeddings is essential for matching intents that may not share surface-level lexical overlap. Interestingly, performance slightly drops to 93.60% accuracy when α increases to 1.0 (fully dense similarity) at $k = 5$. This may be due to retrieving examples that, despite their high embedding-based similarity, still include irrelevant tokens.

These findings highlight the effectiveness of a hybrid retrieval strategy at moderate α values (e.g., 0.7), which balances semantic relevance with lexical-level control. In general, tuning α to emphasize dense similarity ($\alpha \geq 0.5$) and increasing k leads to consistent gains in ID performance.

Slot Filling. Compared to ID, the performance of SF exhibits less stability across values of α and k, as illustrated in Figure 6. Overall, the results are more scattered and suggest greater sensitivity to the interplay between retrieval parameters. The best performance is achieved at $\alpha = 0.3$ and $k = 5$, yielding an F1 score of 82.19%, whereas the worst result occurs at $\alpha = 1.0$ and $k = 1$, with a score of 79.95%. Notably, in contrast to the ID task, increasing the number of examples, particularly at higher α values, does not necessarily lead to performance gain. This suggests that relying solely on embedding-based similarity is not sufficient when including more examples.

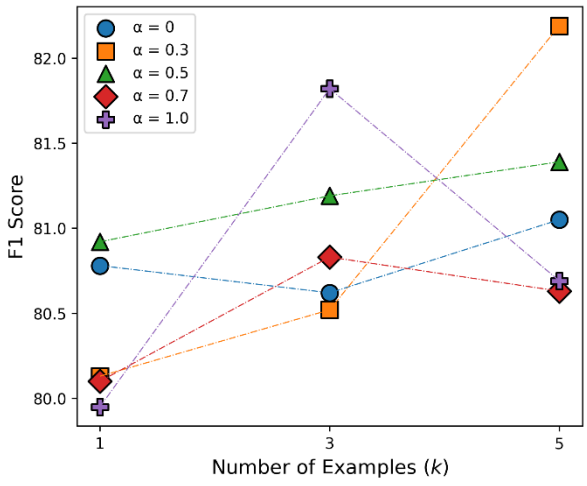

Figure 6. The impact of parameters α and k on slot filling F1 score.

The irregular pattern observed reflects the nature of sequence tagging, which depends more heavily on token-level alignment and exact lexical cues. These findings suggest a trade-off between the informativeness of examples and their quality.

Table 1. Main results comparing different zero-shot cross-lingual methods for Persian. The best scores are shown in bold.

Methods	Intent Detection (ID) Accuracy	Slot Filling (SF) micro-F1 score
GAN-based model (mBERT + BiLSTM) [33]	76.15	13.00
GAN-based model (encoder-decoder) + Multi-task Learning [36]	86.45	59.60
Adapted Few-shot Prompting ($k = 1$) [14]	63.80	74.61
Adapted Few-shot Prompting ($k = 3$) [14]	65.40	80.60
Adapted Few-shot Prompting ($k = 5$) [14]	84.40	**82.50**
Zero-shot ChatGPT (GPT-4o) [14]	60.40	75.46
Proposed Method: Dynamic Few-Shot Prompting ($k = 1$)	86.80 ($\alpha = 1.0$)	80.92 ($\alpha = 0.5$)
Proposed Method: Dynamic Few-Shot Prompting ($k = 3$)	92.80 ($\alpha = 1.0$)	81.82 ($\alpha = 1.0$)
Proposed Method: Dynamic Few-Shot Prompting ($k = 5$)	**94.40** ($\alpha = 0.7$)	**82.19** ($\alpha = 0.3$)

Table 2. Performance of our method on the intent detection task across different parameter configurations. The best values are in bold.

		Micro Average	Macro Average			Weighted Average		
α	k	Precision/Recall/F1	Precision	Recall	F1	Precision	Recall	F1
0	1	80.40	34.81	36.20	32.95	96.89	80.40	87.43
0	3	88.80	48.45	53.04	49.02	96.51	88.80	92.22
0	5	92.00	52.02	53.61	52.33	97.56	92.00	94.55
0.3	1	84.60	45.55	50.32	44.92	96.48	84.60	89.76
0.3	3	90.80	56.39	56.66	55.24	97.54	90.80	93.80
0.3	5	92.80	58.66	58.86	58.56	97.15	92.80	94.87
0.5	1	82.00	37.66	41.25	36.72	97.02	82.00	88.33
0.5	3	89.20	53.55	56.56	53.56	96.83	89.20	92.55
0.5	5	92.60	59.65	62.78	60.52	97.76	92.60	94.95
0.7	1	85.40	49.27	50.76	48.67	97.35	85.40	90.64
0.7	3	90.60	57.53	64.54	58.50	97.58	90.60	93.64
0.7	5	**94.40**	**69.08**	**71.01**	**69.69**	**97.98**	**94.40**	**96.08**
1.0	1	86.80	45.35	50.57	45.29	96.94	86.80	91.11
1.0	3	92.80	67.87	70.76	68.79	97.51	92.80	94.99
1.0	5	93.60	56.86	60.58	57.84	97.47	93.60	95.34

4.3.2. Comparison of Different Models

To address **RQ2**, we compare the performance of our proposed method (Dynamic Few-Shot Prompting) against various baselines, including GAN-based models [33, 36], a zero-shot ChatGPT setting, and an adapted few-shot prompting method [14] across different values of k. Table 1 reports the results for both ID and SF.

The GAN-based baseline (mBERT + BiLSTM) [33], which employs an adversarial learning approach to generate language-independent representations, achieved reasonably good results for ID but performed poorly on SF, primarily due to its limited capacity to capture fine-grained, token-level dependencies in Persian. In contrast, the other GAN-based baseline (encoder-decoder), which utilizes the multilingual BART (mBART) architecture and emphasizes preserving semantic content, achieved significantly better results for both the ID and SF tasks.

Zero-shot ChatGPT, relying solely on its internal knowledge, produced moderate results for both tasks, with 60.40% accuracy on ID and an F1 score of 75.46% for SF. This highlights the limitations of general-purpose LLMs in zero-shot settings, especially for structured language understanding tasks. Nevertheless, it serves as a useful baseline that offers insight into performance without task-specific supervision.

Comparing our method with the Adapted Few-shot Prompting approach, which is the most similar baseline to ours, under various values of k (each using its optimal α), we observe substantial improvements for ID across all k values. This underscores the importance of sentence-level semantic similarity in ID and suggests that dynamically retrieved examples offer greater utility than static ones, even when length and diversity are considered.

Furthermore, for SF, while our method achieves performance on par with the adapted approach at $k = 5$, it outperforms it at $k = 1$ and $k = 3$. These findings indicate that even feeding a small number of dynamic examples to an LLM can be more informative than statically selected ones.

Table 3. Performance of our method on the slot filling task across different parameter configurations. The best values are in bold.

		Micro Average			Macro Average			Weighted Average		
α	k	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	1	79.99	81.58	80.78	59.08	59.11	57.11	82.09	81.58	81.46
0	3	78.86	82.46	80.62	55.28	58.22	55.25	81.46	82.46	81.62
0	5	79.51	82.64	81.05	61.22	63.43	60.61	81.77	82.64	81.89
0.3	1	79.39	80.88	80.13	56.89	60.00	56.04	80.92	80.88	83.06
0.3	3	78.82	82.29	80.52	63.45	65.75	62.97	81.35	82.29	81.52
0.3	5	**80.53**	**83.93**	**82.19**	**66.68**	62.23	**65.73**	**82.75**	**83.93**	**83.06**
0.5	1	79.92	81.94	80.92	62.49	64.98	61.59	81.87	81.94	81.51
0.5	3	79.67	82.76	81.19	65.02	**67.61**	64.85	81.64	82.76	81.91
0.5	5	79.77	82.87	81.39	62.28	66.67	62.57	82.06	82.87	82.11
0.7	1	79.12	81.11	80.10	61.84	64.14	60.99	81.18	81.11	80.82
0.7	3	79.04	82.70	80.83	60.19	63.77	60.06	81.61	82.70	81.79
0.7	5	78.72	82.64	80.63	60.02	64.37	60.30	81.34	82.64	81.62
1.0	1	78.55	81.41	79.95	57.04	60.01	56.19	81.06	81.41	80.87
1.0	3	80.41	83.28	81.82	63.75	66.30	63.34	82.43	83.28	82.54
1.0	5	78.66	81.41	80.69	61.23	62.94	60.73	81.46	82.82	81.78

The statistical reliability of the reported results is assessed in Appendix B.

4.3.3. Fine-Grained Performance Analysis

In response to **RQ3**, we conduct a fine-grained analysis to examine how performance varies across different α and k configurations in terms of precision, recall, and F1.

***Intent Detection*.** Table 2 summarizes the results of our method on the ID task under various configurations. Micro-level metrics, which are identical to accuracy, indicate strong performance across settings, with the best results achieved at $\alpha = 0.7$ and $\alpha = 1.0$. Macro-level scores show more variation, reflecting differences across intent types. However, the consistently high weighted scores suggest that the model performs well when accounting for class frequency. Taken together, the results affirm that increasing the number of dynamically selected support examples enhances intent detection performance.

Slot Filling. The detailed results in Table 3 show that micro and weighted F1 scores for SF remain consistently high across all configurations, indicating strong overall performance at the token level. Conversely, macro F1 fluctuates more, reflecting variation in handling less frequent slot types. Nevertheless, the best configuration, which aligns across all three metrics, suggests good generalization even to underrepresented classes.

Notably, even with $k = 1$, our method remains competitive (micro $F1 \geq 80\%$ for most α values), demonstrating the value of dynamic prompting even with few support examples. The stability across α further confirms the robustness of our method.

In general, the model maintains a good balance between precision and recall, and gains in macro F1 also indicate its ability to handle label imbalance through dynamic selection.

5. Conclusion

In this work, we introduced a zero-shot cross-lingual method for Persian language understanding, leveraging machine translation, dynamic example retrieval, and GPT-4o prompting. Without requiring labeled data in the target language, our approach constructs input-sensitive prompts from semantically and lexically similar examples retrieved from the source language through a hybrid similarity strategy. Experiments on the Persian-ATIS dataset demonstrate that our method improves intent detection and achieves competitive slot filling performance, outperforming strong baselines, including zero-shot LLM inference and prompting techniques with fixed examples. Moreover, it remains effective even with a single retrieved example, underscoring its practicality for low-resource scenarios. In future work, we plan to extend this modular pipeline to other domains and languages, and explore prompt optimization under broader cross-lingual settings.

References

[1] Y. Wang, J. Zhang, T. Shi, D. Deng, Y. Tian, and T. Matsumoto, "Recent advances in interactive machine translation with large language models," *IEEE Access*, vol. 12, pp. 179353-179382, 2024.

[2] H. Zhang, P. S. Yu, and J. Zhang, "A systematic survey of text summarization: From statistical methods to large language models," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1-41, 2025.

[3] A. Algherairy and M. Ahmed, "Prompting large language models for user simulation in task-oriented dialogue systems," *Computer Speech & Language*, vol. 89, no. 101697, 2025.

[4] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, and M. Fazel-Zarandi, "Scaling speech technology to 1,000+

languages," *Journal of Machine Learning Research*, vol. 25, no. 97, pp. 1-52, 2024.

[5] P. Pakray, A. Gelbukh, and S. Bandyopadhyay, "Natural language processing applications for low-resource languages," *Natural Language Processing*, vol. 31, no. 2, pp. 183-197, 2025.

[6] M. Firdaus, H. Golchha, A. Ekbal, and P. Bhattacharyya, "A deep multi-task model for dialogue act classification, intent detection and slot filling," *Cognitive Computation*, vol. 13, no. 3, pp. 626-645, 2021.

[7] A. Algherairy and M. Ahmed, "A review of dialogue systems: current trends and future directions," *Neural Computing and Applications*, vol. 36, no. 12, pp. 6325-6351, 2024.

[8] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, and X. Zhu, "Recent advances and challenges in task-oriented dialog systems," *Science China Technological Sciences*, vol. 63, no. 10, pp. 2011-2027, 2020.

[9] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, vol. 2 (Short Papers), pp. 753-757.

[10] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A survey on recent approaches for natural language processing in low-resource scenarios," *arXiv preprint arXiv:2010.12309*, 2020.

[11] T. Adimulam, S. Chinta, and S. K. Pattanayak, "Transfer learning in natural language processing: Overcoming low-resource challenges," *International Journal of Enhanced Research in Science Technology & Engineering*, vol. 11, no.12, pp. 65-79, 2022.

[12] Y. Fu, N. Lin, B. Chen, Z. Yang, and S. Jiang, "Cross-lingual named entity recognition for heterogenous languages," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 371-382, 2022.

[13] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024.

[14] S. Tahery and S. Farzi, "An Adapted Few-Shot Prompting Technique Using ChatGPT to Advance Low-Resource Languages Understanding," *IEEE Access*, vol. 13, pp. 93614-93628, 2025.

[15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, and T. Rocktäschel, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems 33, Annual Conference on Neural Information Processing Systems 2020*, *NeurIPS* 2020, pp. 9459-9474.

[16] A. Conneau and G. Lample, "Cross-lingual language model pretraining," *Advances in neural information processing systems 32*, *Annual Conference on Neural Information Processing Systems 2019*, *NeurIPS* 2019, pp. 7057–7067.

[17] A. Siddhant, M. Johnson, H. Tsai, N. Ari, J. Riesa, A. Bapna, O. Firat, and K. Raman, "Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 05, pp. 8854-8861.

[18] C. D. Manning, P. Raghavan, and H. Schütze, An introduction to information retrieval, Cambridge University Press, 2009.

[19] M. J. Sabet, P. Dufter, F. Yvon, and H. Schütze, "SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings," in *Findings of the Association for Computational Linguistics: EMNLP 2020, 2020: Association for Computational Linguistics*, pp. 1627-1643.

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

[21] E. Razumovskaia, G. Glavaš, O. Majewska, A. Korhonen, and I. Vulić, "Crossing the Conversational Chasm: A Primer on Multilingual Task-Oriented Dialogue Systems," *Journal of Artificial Intelligence Research*, vol. 24, pp. 351-1402, 2022.

[22] K. Yu, H. Li, and B. Oguz, "Multilingual seq2seq training with similarity loss for cross-lingual document classification," in *Proceedings of the third workshop on representation learning for NLP*, 2018, pp. 175-179.

[23] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," *Advances in neural information processing systems 30, Annual Conference on Neural Information Processing Systems 2017, NeurIPS* 2017, pp. 6294-6305.

[24] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 3795–3805.

[25] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019: Association for Computational Linguistics*, pp. 4996–5001.

[26] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, 2019: Association for Computational Linguistics, pp. 8440–8451.

[27] R. Zadkamali, S. Momtazi, and H. Zeinali, "Intent detection and slot filling for Persian: Cross-lingual training for low-resource languages," *Natural Language Processing*, vol. 31, no. 2, pp. 559-574, 2025.

[28] Z. Li, C. Hu, J. Chen, Z. Chen, X. Guo, and R. Zhang, "Improving Zero-Shot Cross-Lingual Transfer via Progressive Code-Switching," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, 2024, pp. 6388-6396.

[29] L. Qin, Q. Chen, T. Xie, Q. Li, J.-G. Lou, W. Che, and M.-Y. Kan, "GL-CLeF: A global-local contrastive learning framework for cross-lingual spoken language understanding," in *Proceedings of the 60th annual meeting of the association for computational linguistics*, vol. 1 (Long Papers), 2022, pp. 2677–2686.

[30] L. Qin, M. Ni, Y. Zhang, and W. Che, "Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020, pp. 3853–3860.

[31] Z. Liu, G. I. Winata, Z. Lin, P. Xu, and P. Fung, "Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems," in *Proceedings of the AAAI Conference on Artificial Intelligence,* 2020, vol. 34, no. 05, pp. 8433-8440.

[32] P. Safari and M. Shamsfard, "Data augmentation and preparation process of PerInfEx: a Persian chatbot with the ability of information extraction," *IEEE Access*, vol. 12, pp. 19158-19180, 2024.

[33] S. Tahery, S. Kianian, and S. Farzi, "Cross-Lingual NLU: Mitigating Language-Specific Impact in Embeddings Leveraging Adversarial Learning," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 4158-4163.

[34] W. U. Ahmad, Z. Zhang, X. Ma, K.-W. Chang, and N. Peng, "Cross-lingual dependency parsing with unlabeled auxiliary languages," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019, pp. 372-382.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems 27, Annual Conference on Neural Information Processing Systems 2014, NeurIPS* 2014, pp. 2672-2680.

[36] S. Tahery and S. Farzi, "An Invasive Embedding Model in Favor of Low-Resource Languages Understanding," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 24, no. 12, pp. 1-24, 2025.

[37] J. Pei, G. Yan, M. De Rijke, and P. Ren, "Mixture-of-Languages Routing for Multilingual Dialogues," *ACM Transactions on Information Systems*, vol. 42, no. 6, pp. 1-33, 2024.

[38] T. Labruna, S. Brenna, and B. Magnini, "Dynamic Task-Oriented Dialogue: A Comparative Study of Llama-2 and Bert in Slot Value Generation," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2024, pp. 358-368.

[39] A. R. Ghasemi and J. Salimi Sartakhti, "Multilingual Language Models in Persian NLP Tasks: A Performance Comparison of Fine-Tuning Techniques," *Journal of AI and Data Mining*, vol. 13, no. 1, pp. 107-117, 2025.

[40] W. Pan, Q. Chen, X. Xu, W. Che, and L. Qin, "A preliminary evaluation of chatgpt for zero-shot dialogue understanding*," arXiv preprint arXiv:2304.04256*, 2023.

[41] Z. Zhu, X. Cheng, H. An, Z. Wang, D. Chen, and Z. Huang, "Zero-shot spoken language understanding via large language models: A preliminary study," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 17877-17883.

[42] Z. Borhanifard, H. Basafa, S. Z. Razavi, and H. Faili, "Persian language understanding in task-oriented dialogue system for online shopping," in *2020 11th International Conference on Information and Knowledge Technology (IKT)*, 2020: IEEE, pp. 79-84.

[43] M. Akbari, A. Mohades, and M. H. Shirali-Shahreza, "A hybrid architecture for out of domain intent detection and intent discovery," in *2025 11th International Conference on Web Research (ICWR)*, 2025: IEEE, pp. 137-144.

[44] E. A. Abyaneh, R. Zolfaghari, and A. A. Abyaneh, "User Intent Detection in Persian Text-Based Chatbots: A Comprehensive Review of Methods and Challenges," *in 2025 11th International Conference on Web Research (ICWR)*, 2025: IEEE, pp. 243-249.

[45] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35-43, 2001.

[46] D. Kamali, B. Janfada, M. E. Shenasa, and B. Minaei-Bidgoli, "Evaluating Persian Tokenizers," *arXiv preprint arXiv:2202.10879*, 2022.

Appendix A

To better interpret the quantitative results, we conducted a manual qualitative inspection of representative samples, as shown in Table 4. The analysis reveals how translation errors, tokenization mismatches, and alignment deviations interact and affect the ID and SF tasks.

Sample 1 illustrates a case where the overall translation quality is strong.

Table 4. Samples from the proposed pipeline showing translation and alignment.

Sample 1 ($\alpha = 0.7$)	
Input utterance	**من یک پرواز صبح از نیوارک به لس آنجلس می‌خواهم** ["من"، "یک"، "پرواز"، "صبح"، "از"، "نیوارک"، "به"، "لس"،"آنجلس"، "می‌خواهم"]
Translation (FA→EN)	I want a morning flight from New York to Los Angeles
Top-k English Examples ($k = 3$)	**1.** what flights from new york to los angeles • ["what", "flights", "from", "new", "york", "to", "los", "angeles"] • flight • ['O', 'O', 'O', 'B-fromloc.city_name', 'I-fromloc.city_name', 'O', 'B-toloc.city_name', 'I-toloc.city_name'] **2.** please list the flights from new york to los angeles • ["please", "list", "the", "flights", "from", "new", "york", "to", "los", "angeles"] • flight • ['O', 'O', 'O', 'O', 'O', 'B-fromloc.city_name', 'I-fromloc.city_name', 'O', 'B-toloc.city_name', 'I-toloc.city_name'] **3.** i 'd like a flight from kansas city to los angeles that arrives in los angeles in the late afternoon • ['i', "'d", 'like', 'a', 'flight', 'from', 'kansas', 'city', 'to', 'los', 'angeles', 'that', 'arrives', 'in', 'los', 'angeles', 'in', 'the', 'late', 'afternoon'] • flight • ['O', 'O', 'O', 'O', 'O', 'O', 'B-fromloc.city_name', 'I-fromloc.city_name', 'O', 'B-toloc.city_name', 'I-toloc.city_name', 'O', 'O', 'O', 'B-toloc.city_name', 'I-toloc.city_name', 'O', 'O', 'B-arrive_time.period_of_day', 'I-arrive_time.period_of_day']
Translation (EN→FA)	**1.** چه پروازهایی از نیویورک به لس آنجلس **2.** لطفا پروازهای نیویورک به لس آنجلس را لیست کنید **3.** من مثل پرواز از کانزاس سیتی به لس آنجلس هستم که اواخر بعد از ظهر وارد لس آنجلس می شود
Normalization & Tokenization	**1.** ["چه"، "پروازهایی"، "از"، "نیویورک"، "به"، "لس‌آنجلس"] **2.** ["لطفا"، "پروازهای"، "نیویورک"، "به"، "لس‌آنجلس"، "را"، "لیست"، "کنید"] **3.** ["من"، "مثل"، "پرواز"، "از"، "کانزاس"، "سیتی"، "به"، "لس‌آنجلس"، "هستم"، "که"، "اواخر"، "بعد"، "از"، "ظهر"، "وارد"، "لس‌آنجلس"، "می-شود"]
Alignment	**1.** [[0, 0], [1, 1], [2, 2], [3, 3], [4, 5], [5, 4], [6, 5], [7, 5]] ➔ ['O', 'O', 'O', 'B-fromloc.city_name', 'O', 'I-toloc.city_name'] **2.** [[0, 0], [1, 6], [2, 1], [3, 1], [5, 2], [6, 4], [7, 3], [9, 7]] ➔ ['O', 'O', 'B-fromloc.city_name', 'O', 'I-fromloc.city_name', 'O', 'O', 'I-toloc.city_name'] **3.** [[0, 0], [2, 1], [3, 1], [4, 2], [5, 3], [6, 4], [7, 5], [8, 6], [9, 7], [10, 7], [11, 9], [12, 16], [13, 14], [14, 15], [15, 15], [16, 11], [18, 10], [19, 13]] ➔ ['O', 'O', 'O', 'O', 'B-fromloc.city_name', 'I-fromloc.city_name', 'O', 'I-toloc.city_name', 'O', 'O', 'B-arrive_time.period_of_day', 'O', 'O', 'I-arrive_time.period_of_day', 'O', 'I-toloc.city_name', 'O']
Output	➢ Intent: flight ✓ ➢ ['O', 'O', 'O', 'depart_time.period_of_day', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name', 'I-toloc.city_name', 'O']
Sample 2 ($\alpha = 0.3$)	
Input utterance	**ارزان‌ترین پروازهای یک طرفه از هیوستون به بوستون** ["ارزان‌ترین"، "پروازهای"، "یک"، "طرفه"، "از"، "هیوستون"، "به"، "بوستون"]
Translation (FA→EN)	The cheapest one -way flights from Houston to Boston
Top-k English Examples ($k = 3$)	**1.** what 's the cheapest one way flight from oakland to boston • ['what', "'s", 'the', 'cheapest', 'one', 'way', 'flight', 'from', 'oakland', 'to', 'boston'] • flight • ['O', 'O', 'O', 'B-cost_relative', 'B-round_trip', 'I-round_trip', 'O', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name'] **2.** what is the cheapest one way flight from atlanta to boston • ['what', 'is', 'the', 'cheapest', 'one', 'way', 'flight', 'from', 'atlanta', 'to', 'boston'] • flight • ['O', 'O', 'O', 'B-cost_relative', 'B-round_trip', 'I-round_trip', 'O', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name']

Table 4. *Continued.*

	3. what are the cheapest one way flights from denver to Atlanta • ['what', 'are', 'the', 'cheapest', 'one', 'way', 'flights', 'from', 'denver', 'to', 'atlanta'] • flight • ['O', 'O', 'O', 'B-cost_relative', 'B-round_trip', 'I-round_trip', 'O', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name']
Translation (EN→FA)	**1.** چه چیزی ارزانترین پرواز از یک طرفه از اوکلند به بوستون است **2.** ارزانترین پرواز یک طرفه از آتلانتا به بوستون چیست **3.** ارزانترین پروازهای یک طرفه از دنور به آتلانتا چیست؟
Normalization & Tokenization	**1.** ["چه"، "چیزی"، "ارزانترین"، "پرواز"، "از"، "یک‌طرفه"، "از"، "اوکلند"، "به"، "بوستون"، "است"] **2.** ["ارزانترین"، "پرواز"، "یک‌طرفه"، "از"، "آتلانتا"، "به"، "بوستون"، "چیست"] **3.** ["ارزانترین"، "پروازهای"، "یک‌طرفه"، "از"، "دنور"، "به"، "آتلانتا"، "چیست"، "؟"]
Alignment	**1.** [[0, 0], [1, 1], [3, 2], [4, 5], [5, 5], [6, 3], [7, 4], [7, 6], [8, 7], [9, 8], [10, 9]] ➔ ['O', 'O', 'B-cost_relative', 'O', 'O', 'I-round_trip', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name', 'O'] **2.** [[0, 0], [3, 0], [4, 2], [5, 2], [6, 1], [7, 3], [8, 4], [9, 5], [10, 6], [10, 7]] ➔ ['B-cost_relative', 'O', 'I-round_trip', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name', 'B-toloc.city_name'] **3.** [[0, 0], [2, 1], [3, 0], [4, 2], [5, 2], [6, 1], [7, 3], [8, 4], [9, 5], [10, 6]] ➔ ['B-cost_relative', 'O', 'I-round_trip', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name', 'O', 'O']
Output	➢ Intent: flight ✖ ➢ ['B-cost_relative', 'O', 'I-round_trip', 'I-round_trip', 'O', 'B-fromloc.city_name', 'O', 'B-toloc.city_name']
Each pair [i, j] shows that the i^{th} English token is aligned with the j^{th} Persian token.	

The minor issues observed mainly concern sentence fluency rather than semantic accuracy. Consequently, the translated utterance preserves the intended meaning and accurately renders all slot-bearing entities (e.g., fromloc.city_name, toloc.city_name). Although the city name "نیوارک" ("Newark") was translated as "New York," and this substitution led to the retrieval of semantically different English examples, the retrieved utterances still belonged to the same intent category (flight), and therefore did not affect the model's final prediction.

Moreover, Persian tokenization and normalization introduce certain systematic mismatches. Due to the use of half-spaces, compounds such as "می‌خواهم" must not be split into "می" and "خواهم" for consistent processing. We used the Hazm toolkit to normalize tokens, which standardizes the text according to Persian orthographic rules. However, this process sometimes produces discrepancies with the gold data. For example, proper names like "لس‌آنجلس" may appear as two separate tokens ("لس"، "آنجلس") in the gold annotation. Such inconsistencies can misalign tokens during alignment evaluation.

Nevertheless, when the alignment correctly identifies the slot types (even if the B–I tags are not perfectly assigned), the LLM's contextual understanding often compensates for these inconsistencies. In this sample, despite minor alignment errors and tokenization mismatches, the model correctly predicted both the intent (flight) and all slot labels, including the proper B–I structure for "Los Angeles." This suggests that the overall translation–alignment pipeline preserves essential slot semantics even under noisy conditions.

Sample 2, in contrast, highlights a more challenging case where the intent boundary is semantically ambiguous. The correct intent should be airfare, but all top-*k* retrieved examples were labeled as flight, introducing a bias toward that intent during prompting. While the translation correctly rendered most slot-related tokens, the expression "یک طرفه" ("one-way") proved difficult for the alignment model to handle. The system interpreted it as a round_trip tag but failed to produce the correct B–I structure.

Given that the evaluation was performed at the span level, such boundary-level mismatches can lead to substantial penalties. Even a small misrecognition around compound slots such as "یک طرفه" may propagate considerable error into the span-based metrics. In contrast, similar issues were handled more effectively for city and country names, suggesting that the model can better cope with such entities.

In a nutshell, these qualitative cases demonstrate that translation errors in our setup rarely distort slot

semantics, while normalization and tokenization in Persian introduce systematic yet predictable mismatches. The results confirm that the translation–alignment pipeline remains sufficiently reliable for cross-lingual intent detection and slot filling. At the same time, targeted improvements addressing specific sources of error could further strengthen the model's stability and consistency in future work.

Appendix B

To assess the statistical reliability of the reported results, we estimate confidence intervals for both ID accuracy and SF F1 using non-parametric bootstrap resampling, which allows uncertainty estimation directly from the empirical distribution of predictions on the test set.

We perform 10,000 bootstrap resamples, each drawn with replacement from the 500 test instances, using a fixed random seed (42) for reproducibility. For each resample, ID accuracy and SF F1 are recomputed, and 95% confidence intervals are derived from the resulting empirical score distribution. Given the limited size of the test set, these intervals are approximate, but they provide a meaningful indication of the stability of the reported metrics.

As shown in Table 5, the reference scores (as reported in Table 1) fall well within their corresponding 95% bootstrap confidence intervals, demonstrating that the results are stable with respect to test set variability. These findings indicate that the reported metrics exhibit stable behavior under resampling-driven uncertainty estimation.

Table 5. 95% bootstrap confidence intervals for the proposed method's performance.

Methods	Intent Detection (ID) Accuracy	Slot Filling (SF) micro-F1 score
Dynamic Few-Shot Prompting ($k = 1$)	[84.00, 89.80] ($\alpha = 1.0$)	[78.90, 82.90] ($\alpha = 0.5$)
Dynamic Few-Shot Prompting ($k = 3$)	[90.80, 95.20] ($\alpha = 1.0$)	[79.80, 83.80] ($\alpha = 1.0$)
Dynamic Few-Shot Prompting ($k = 5$)	[92.40, 96.40] ($\alpha = 0.7$)	[80.10, 84.30] ($\alpha = 0.3$)

پرامپت‌دهی پویا مبتنی بر بازیابی برای درک گفتگوی بین‌زبانی در فارسی

صاعده طاهری* و سعید فرضی

گروه هوش مصنوعی دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیر الدین طوسی، تهران، ایران.

ارسال ۲۰۲۵/۰۷/۲۷؛ بازنگری ۲۰۲۵/۱۱/۱۳؛ پذیرش ۲۰۲۵/۱۱/۲۴

چکیده:

درک گفتگو برای زبان‌های کم‌منبع از جمله فارسی همچنان چالشی اساسی به شمار می‌آید، چرا که کمبود دادگان برچسب‌خورده، امکان آموزش نظارت‌شده در مقیاس وسیع را محدود می‌کند. در این پژوهش، روشی ساده اما کارآمد و بی‌نیاز از آموزش ارائه می‌دهیم که با بهره‌گیری از ترجمه ماشینی، انتخاب مثال مبتنی بر بازیابی و پرامپت‌دهی به یک مدل زبانی بزرگ (GPT-4o)، عملکرد مدل را در شرایط بین‌زبانی بهبود می‌بخشد. در این روش، ابتدا گفته فارسی به انگلیسی ترجمه می‌شود. سپس، با استفاده از یک تابع شباهت ترکیبی، مثال‌هایی از پایگاه داده انگلیسی که از نظر معنایی و واژگانی با ورودی مشابه‌اند بازیابی شده و مجدد به فارسی برگردانده می‌شوند. این مثال‌های پویا که بر اساس گفته ورودی تنظیم می‌شوند، در قالب یک پرامپت چند-نمونه‌ای به مدل زبانی بزرگ داده می‌شوند. این راهبرد حساس به ورودی، موجب افزایش غنای اطلاعاتی مثال‌ها شده و به مدل کمک می‌کند تا با هر نمونه دقیق‌تر هم‌راستا شود. نتایج آزمایش‌ها بر روی مجموعه داده Persian-ATIS نشان می‌دهند که روش پیشنهادی ما در تشخیص مقصود کاربر عملکرد بهتری را ارائه می‌نماید و در وظیفه پرکردن شیارها نیز نتایجی رقابتی نسبت به قوی‌ترین روش‌های پایه به دست می‌آورد، بدون آن که به نظارت در زبان هدف نیازمند باشد. روش پیشنهاد شده با خط لوله پیمانه‌ای، بازتولیدپذیر بوده و می‌تواند در آینده به سایر زبان‌های کم‌منبع، وظایف متنوع دیگر، یا پیکربندی‌های مختلف بازیابی تعمیم یابد. منابع مربوط به این پژوهش، از طریق نشانی https://github.com/saedeht/Persian_Language_Understanding در دسترس است.

کلمات کلیدی: تطبیق بین‌زبانی، درک زبان طبیعی، زبان فارسی، مدل‌های زبانی بزرگ، ChatGPT.