



Stairway to AI: Ease the Engagement of Low-Tech users to the AI-on-Demand platform through AI, H2020

StairwAI Chatbot Prototype

Deliverable information	
Deliverable number	D4.2
WP number and title	WP4: Multi-lingual interaction with the platform
Lead beneficiary	Tilde
Dissemination level	Public
Due date	30.12.2022
Actual date of delivery	30.12.2022
Author(s)	Inguna Skadiņa, Miquel Buxons Vives, Javier Vázquez Salceda, Ilhan Aslan, Michelle Xiao Lin Foo, Jurgita Kapočūtė-Dzikienė, Michele Lombardi
Contributors	Tilde, UNIBO, UPC, HUAWEI
Deliverable reviewers	Roberta Calegari (UNIBO), Marco Rorro (EGI)



Document Control Sheet

Version	Date	Summary of changes	Author(s)
0.1	20.11.2022	First Table of Contents	Inguna Skadiņa
0.2.	10.12.2022	Initial content from partners	Miquel Buxons Vives, Javier Vázquez Salceda, Ilhan Aslan, Michelle Xiao Lin Foo, Jurgita Kapočičūtė-Dzikienė, Michele Lombardi
0.3	15.12.2022	First draft ready for internal review	Inguna Skadiņa, Miquel Buxons Vives, Javier Vázquez Salceda, Ilhan Aslan, Michelle Xiao Lin Foo, Jurgita Kapočičūtė-Dzikienė, Michele Lombardi
0.4	23.12.2022	Internal review	Roberta Calegari, Marco Rorro
1.0	30.12.2022	Final version, comments from the internal review addressed	Inguna Skadiņa, Miquel Buxons Vives, Javier Vázquez Salceda, Ilhan Aslan, Michelle Xiao Lin Foo, Jurgita Kapočičūtė-Dzikienė, Michele Lombardi



Table of contents

1	Executive Summary	6
2	Introduction.....	6
2.1	Purpose and scope of the document	7
2.2	Structure of the document.....	7
3	Overview of StairwAI platform architecture and chatbot implementation platform.....	7
3.1	StairwAI platform and role of chatbot	8
3.2	Overview of Tilde.AI platform	9
4	Dialogue Types and Scenarios	9
4.1	Starting conversation with the chatbot and multilinguality.....	10
4.2	Assessment of user’s organization AI readiness.....	11
4.3	Frequently Asked Questions (FAQ).....	13
4.4	Guided dialogue that allows user to find AI assets for a specific use case	13
5	NLU/ NLP tools for multilingual human-computer interaction.....	16
5.1	Text labelling for matchmaking and keyword extraction.....	16
5.2	Multilingual and cross-lingual intent detection	16
5.2.1	A formal definition of the solving task	17
5.2.2	Methodology	17
5.2.3	The data	18
5.2.4	Experiments and results	20
5.2.5	Discussion and conclusions	23
5.3	Knowledge datasets	24
5.3.1	The StairwAI AI Asset Conceptual Model	24
5.3.2	The StairwAI AI Asset Management System.....	24
6	Validation of NLP models	25
6.1	Telegram Rasa	25
6.1.1	Chat Summary	26
6.1.2	Document Question-Answering	27
6.2	Paprr Readr Bot demonstrator	29
7	Summary and next steps	30
7.1	Summary of results.....	30
7.2	Next steps.....	30



Bibliography.....	31
8 Appendices	32
Chatbot integration in StairwAI Website	32



Acronyms

Acronym	Explanation
AI	Artificial Intelligence
AMS	StairwAI's Asset Management System
API	Application Programming Interface
CPU	Central Processing Unit
FAQ	Frequently asked questions
GPU	Graphics Processing Unit
HUA	Huawei
ML	Machine Learning
MT	Machine Translation
MVP	Minimum viable product
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NPU	Neural network Processing Unit
VA	Virtual Assistant
W3C	World Wide Web Consortium
WP	Work Package
XML	eXtensive Markup Language



1 Executive Summary

Deliverable 4.2 introduces to the public demonstrator - StairwAI chatbot prototype, a multilingual interface enabling users' interaction with the StairwAI platform in their native language. The prototype support communication in six StairwAI project languages - English, French, German, Italian, Latvian and Spanish. Deliverable presents implementation of user scenarios defined in WP5 and corresponding dialogues flows. We also present tools, services and interfaces to support natural language understanding and human-computer interaction.

The StairwAI chatbot prototype is based on StairwAI Chatbot MVP, released in M18 and described in D4.1 (Blaudez et al., 2022). This is the first public version of the StairwAI chatbot prototype, available for use and testing at: <https://va.tilde.com/api/prodk8sbotstair1/chat/default.htm>. After user tests, evaluation and elaboration the prototype will be updated to the StairwAI Chatbot Final Version (D4.3) in M30.

2 Introduction

The main goal of the StairwAI project is to enhance the AI-on-demand platform services by means of a service layer that provides *Horizontal Matchmaking* (an AI-based automatic mapping between user requirements and assets of the AI-on-demand platform to meet end users' needs) and *Vertical Matchmaking* (AI-based automatic mechanisms for hardware resource dimensioning and hardware resource provider discovery to satisfy end users' needs). The goal of the StairwAI Chatbot is to ease use of both services supporting user's interaction with the system in her/his native language. Figure 1 shows the planned interactions between the StairwAI (Tenhunen et al., 2022): the core modules of the StairwAI system - the Horizontal Matchmaking module and the Vertical Matchmaking module – are being reached by user through *Multi-Lingual Virtual Assistant* communicating with user in her/his native language.

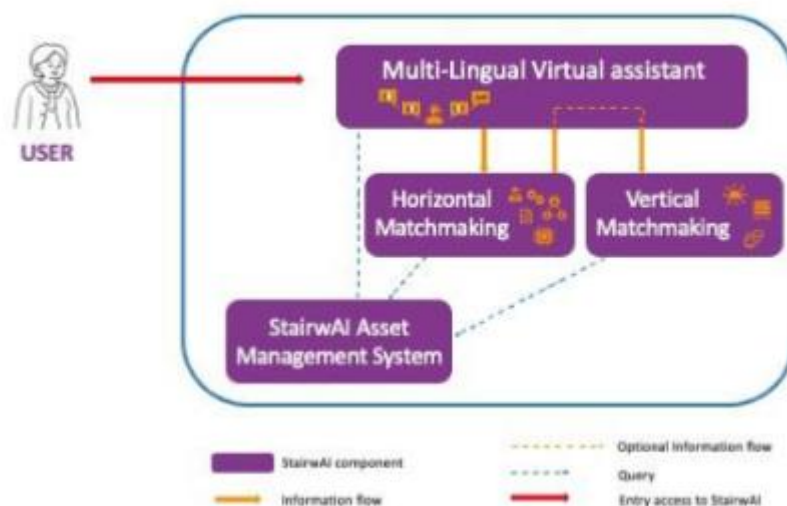


Figure 1. Interactions between StairwAI's main components (Tenhunen et al., 2022).



2.1 Purpose and scope of the document

Deliverable D4.2 is a Demonstrator - set of services and tools that jointly form the *StairwAI Chatbot Prototype*. The prototype is created on Tilde.AI conversational AI platform¹ (presented in detail in Deliverable 4.1 (Blaudez et al., 2022) and introduced also in Subsection 3.2 of this deliverable), it integrates several tools and assets developed by partners of WP4, WP5 and WP3. The deliverable also provides initial scenarios/demonstration of interaction with tools that will be integrated or released at the later stage of this project (Section 5 of this document).

This document is a companion to the Demonstrator of the *StairwAI Chatbot Prototype* available at <https://va.tilde.com/api/prodk8sbotstair1/chat/default.htm>. The main purpose of this document is to briefly introduce to the chatbot and its components.

2.2 Structure of the document

This deliverable extends and updates Deliverable 4.1 StairwAI Chatbot MVP (Blaudez et al., 2022) with actual information about current version of the StairwAI chatbot and corresponding NLP tools, as well as includes summarized related information from D4.1.

Section 3 “Overview of StairwAI platform architecture and chatbot implementation platform” introduces to StairwAI platform architecture in general, the place of the chatbot in this architecture, as well as, provides short overview of Tilde.ai platform, where the chatbot is implemented and available for further integration into StairwAI platform.

Section 4 “Dialogue Types and Scenarios” provides overview of general dialog types and scenarios, describes first steps and multilinguality (Subsection 4.1) and currently implemented dialogs: assessment of AI readiness (Subsection 4.2), frequently asked questions (Subsection 4.3) and matchmaking (Subsection 4.4).

Section 5 “NLU/ NLP tools for multilingual human-computer interaction” introduces to different NLP tools that are already used or will be integrated into chatbot. We present several tools for text classification task: text labelling (Subsection 5.1) which will be included into chatbot as a bridge between user’s input in natural language and matchmaking performed by tools developed in WP5 and multilingual and cross-lingual intent detection (Subsection 5.2) allowing easily switch between languages and extend chatbot to new languages outside the scope of the StairwAI project. Finally, we introduce services developed by StairwAI partners to support interaction between the chatbot platform and knowledge datasets (Subsection 5.3). In particular, StairwAI AI asset conceptual model and Asset management system developed within WP3.

Section 6 “Validation of NLP models” explores and validates applications of NLP models that would be useful for the StairwAI chatbot use case on mobile devices.

Finally, in the Appendices section we include additional materials related to the implementation of the chatbot.

3 Overview of StairwAI platform architecture and chatbot implementation platform

To ease the access to AI assets, the StairwAI platform should enable users to interact with the platform using natural (human) language. To do so, a chatbot service, implementing natural language processing, is needed

¹ <https://va.tilde.com/>



to extract from the interactions (in form of use cases, job offers, profile description, etc.) the user's needs and support user's access to the StairwAI platform.

3.1 StairwAI platform and role of chatbot

Figure 2 provides a general overview of StairwAI platform architecture and the role of the chatbot in human interaction with the platform. The main role of the chatbot is to provide an interface, where the user can communicate and describe her/his needs in natural language. In particular:

- user describes his/her needs in natural language through the dialogue with the chatbot;
- user's input is analysed and additional questions are asked, if necessary, through the dialogue;
- user's input is analysed to extract necessary information for the horizontal matchmaking;
- response to the user's request is prepared from the response of the horizontal matchmaking.

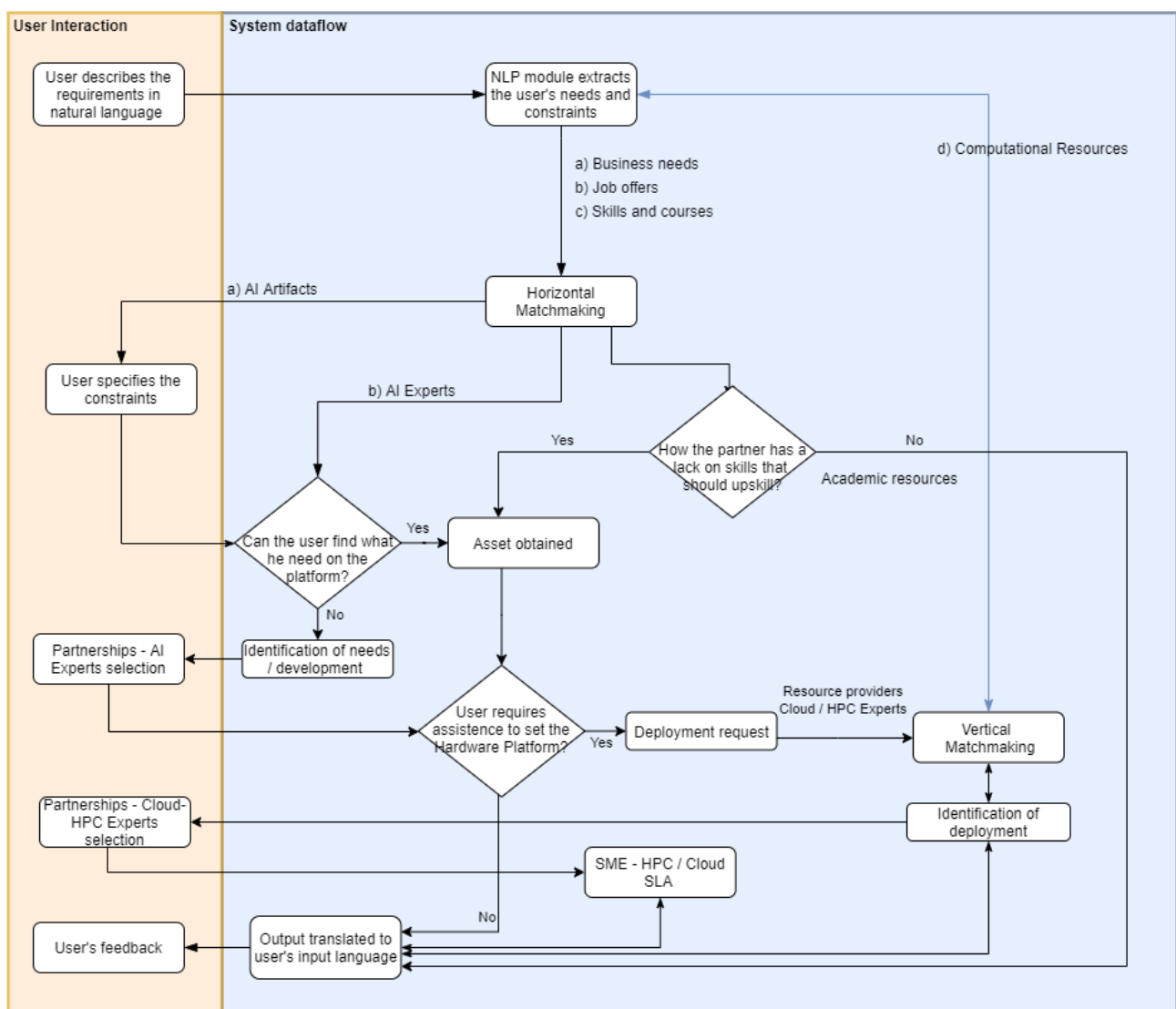


Figure 2. General architecture of the StairwAI platform.



In addition to the clarifying user's needs through the dialogue, two additional communication scenarios are provided: with the help from the chatbot users can assess AI readiness of their organization (see Subsection 4.2) and chatbot can answer different questions about AI and StariwAI project (see Subsection 4.3).

3.2 Overview of Tilde.AI platform

The Tilde.AI conversational platform allows building guided and free form dialogue scenarios to create an intuitive and human-like experience. Virtual Assistants (VAs) can be trained to answer questions and perform various tasks. The most common tasks relevant to StairwAI project include: information search, frequently asked questions and user support.

Typical development process of the VA includes a sequence of actions to be implemented: dialogue scenario development (overview of StairwAI chatbot dialogue scenarios is provided in Chapter 4 "Dialogue Types and Scenarios"), data preparation and training (initial dataset for StairwAI chatbot has been collected in Task 3.3 "Data collection and datasets generation for Multi-lingual NLP", delivered in M17), integration with external tools and services (information on external services developed by partners is provided in Subsection 5.1 "Text labelling for matchmaking and keyword extraction" and Subsection 5.3 "Knowledge datasets") and the testing step (verifying that the assistant's performance is compatible with the set goals and requirements).

Tilde.AI virtual assistants are multilingual and use machine learning algorithms to detect customers' intent and provide the most suitable answers. Tilde.AI supports a variety of communication channels – text, voice or 3D avatar (more details about Tilde.ai platform are provided in Deliverable 4.1).

4 Dialogue Types and Scenarios

According to the structure, three types of dialogue are distinguished: guided, free form and the combined dialogue. In **guided dialogue** the chatbot is the main driver of the conversation, while the user can only select one of the answers on offer. The StairwAI chatbot uses guided dialog to test the AI readiness level of the user's organization. This dialog is described in Subsection 4.2 "Assessment of user's organization AI readiness".

A **free form dialogue** does not entail a specific sequence of questions and answers. Each action from the chatbot depends on the input of the user. The user is the active conversation leader and mover. Free dialog is implemented in StairwAI chatbot for communication on frequently asked questions. This dialog type is illustrated in Subsection 4.3 "Frequently Asked Questions (FAQ)".

The combined dialogue is a combination of a guided dialogue and a free dialogue. The user can be guided by a prearranged conversation plan, prompting him to click on a button, but it is possible to deviate from the conversation provided for in the scenario at any moment by asking for other information in free form, which is not thematically related to the guided dialogue, or by accessing the required state of the guided dialogue.

The StairwAI chatbot prototype is implemented as a combined dialog system demonstrating three different scenarios of human-computer interaction:

- Guided dialogue, that allows to assess user's organization AI readiness level;
- Free dialogue for Frequently Asked Questions (FAQ): in this scenario chatbot that can answer most common questions from user about the StairwAI project, different AI terms, etc.;
- Guided dialogue that helps users to find necessary assets for described use case.



The overall chatbot dialogue diagram is shown in Figure 3.

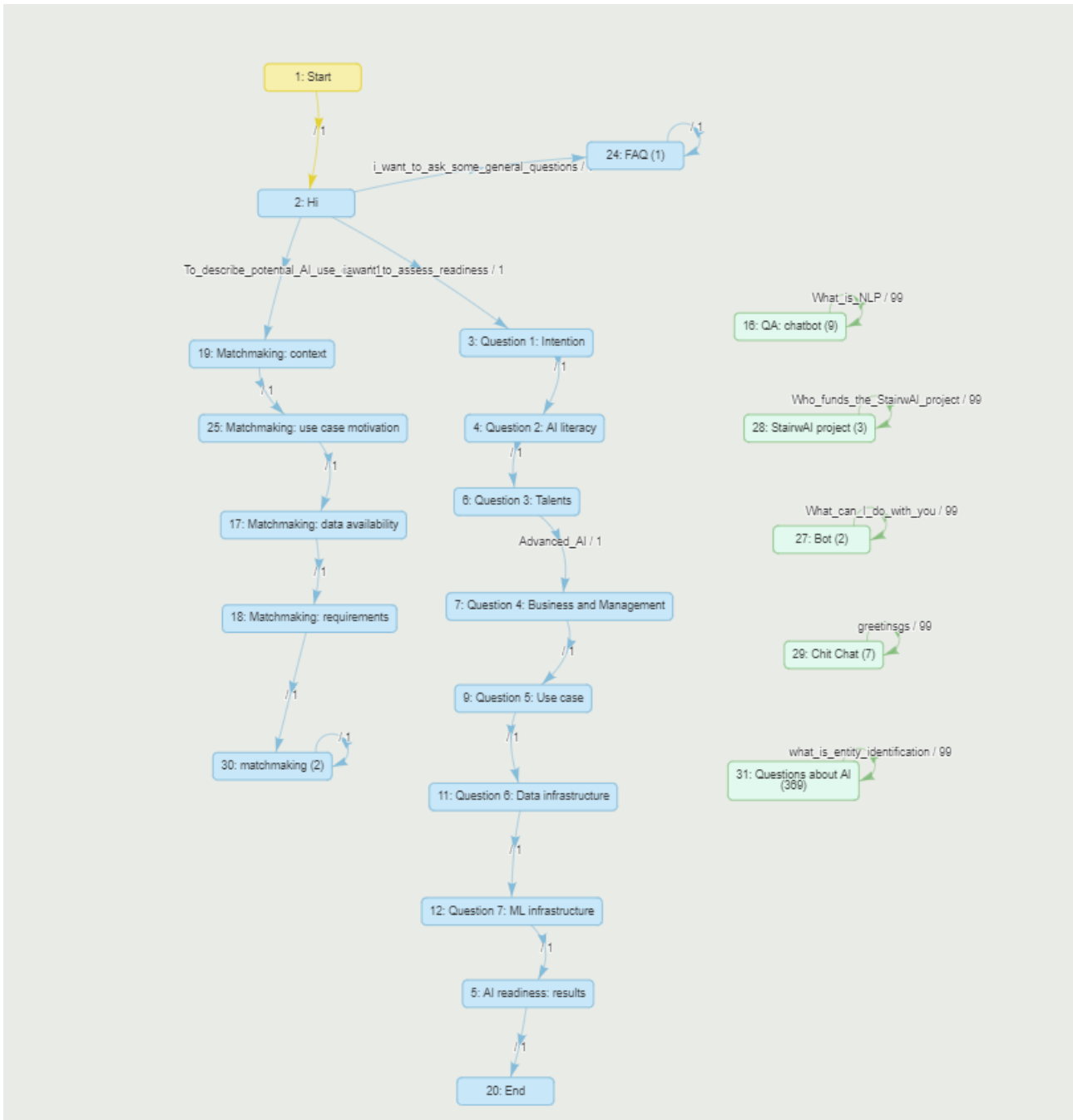


Figure 3. Overview of StairwAI chatbot dialogue scenarios: left – matchmaking, right – AI readiness, green -FAQ.

4.1 Starting conversation with the chatbot and multilinguality

The dialog with the StairwAI chatbot starts with greetings from the chatbot and explanation of which tasks the chatbots can perform. A user can provide her/his answer in free text form or by clicking on the item in a list (Figure 4). Depending on the user’s answer the chatbot helps user to find necessary assets (Subsection 4.4), answers his/her questions (Subsection 4.3) or assess organization’s AI readiness level (Subsection 4.2).



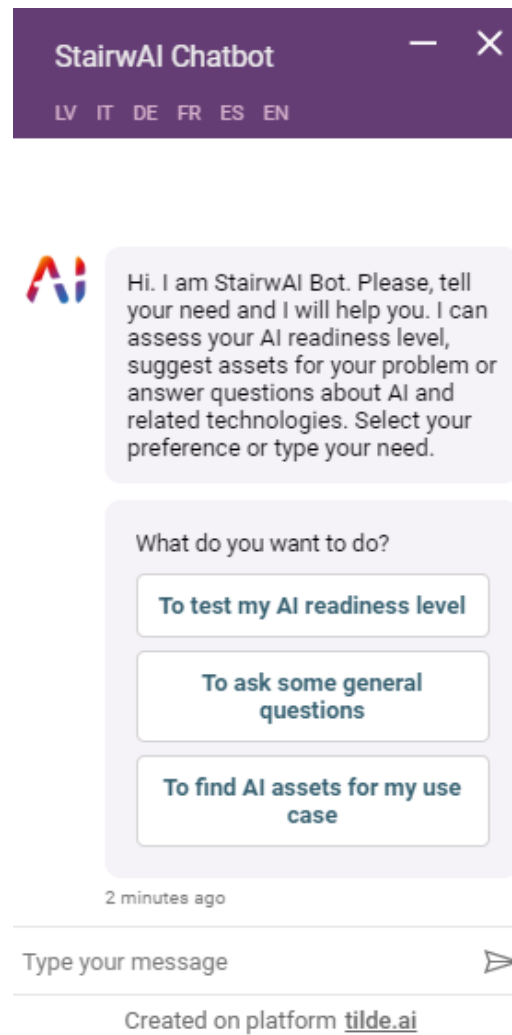


Figure 4. Introductory screen of conversation with the chatbot.

The interaction between chatbot and user always starts in English. The user can change language of interaction in two ways:

- 1) by clicking on the corresponding language button (LV - Latvian, IT - Italian, DE - German, FR- French, ES – Spanish and EN- English),
- 2) by entering utterance in the user’s native language. In this case the language identification module will detect the user’s language and switch dialogue to the requested language. The current version of the chatbot requires that the utterance contains at least 8 words.

4.2 Assessment of user’s organization AI readiness

For the assessment of organization’s AI Readiness level through human-computer interaction, we adapted methodology proposed by WP7 in the Deliverable 7.2. “First Call Announcement and Guide for Applicants” (Sobek et al, 2022) that is derived from the survey developed by AI SINGAPORE how to evaluate AI Readiness of SMEs².

² <https://ai4sme.aisingapore.org/learn-about-gamis/>



Figure 5 shows a guided dialog scenario for AI readiness and an example of the dialogue. The dialogue starts with clarifying the user’s intention (Question 1), followed by six questions (Question 2-Question 7) about organization’s AI skills, management’s readiness, possible use cases, data and IT infrastructures. The result of this assessment - the radar graph - shows the organization’s strengths and weaknesses.

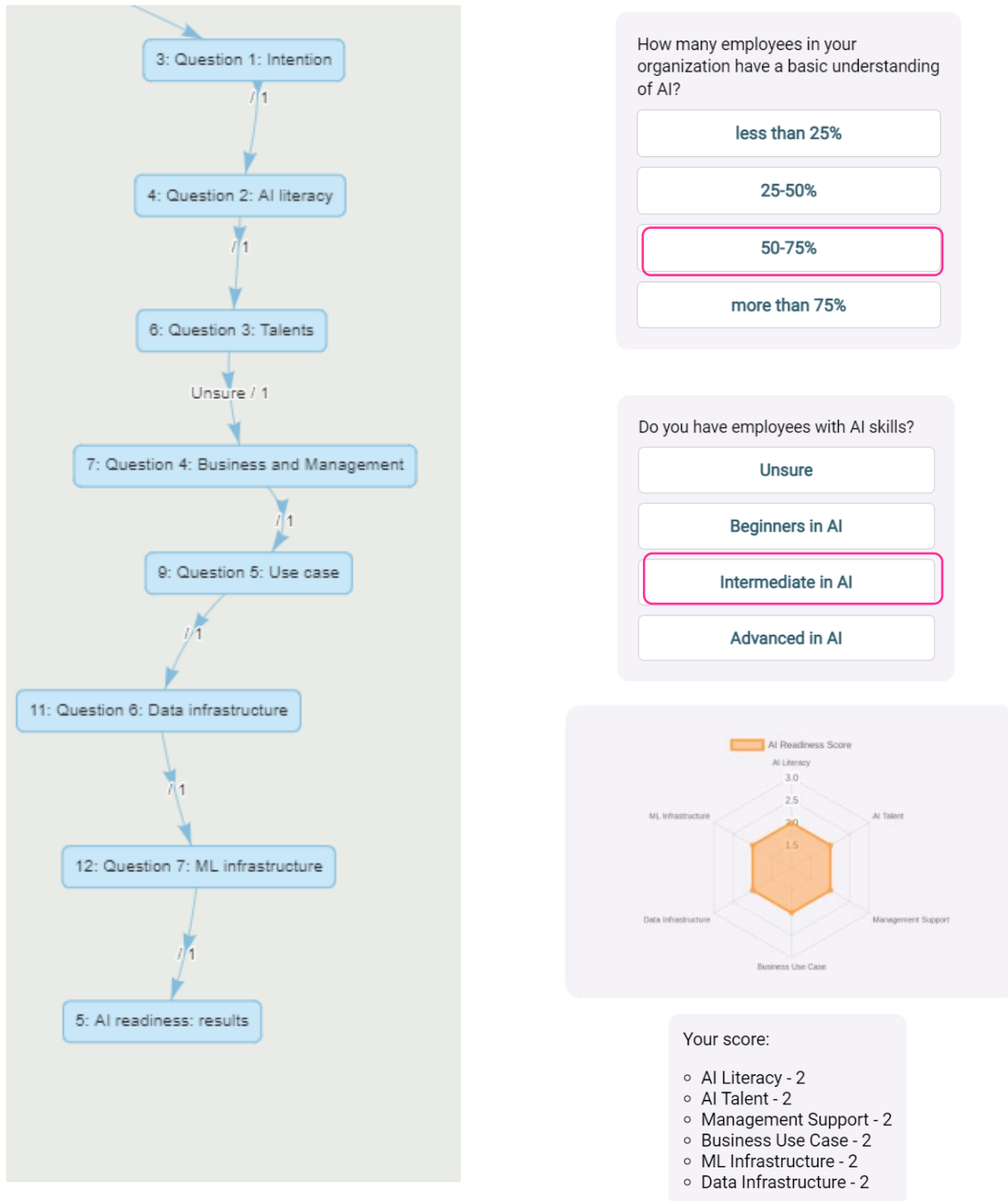


Figure 5. Dialogue for AI readiness evaluation: overall design (on left), fragment of the dialog (top right) and visualisation of AI readiness assessment results (bottom right).



4.3 Frequently Asked Questions (FAQ)

The Frequently Asked Questions constituent allows user to get answers to most common questions on AI, StairwAI project and matchmaking. The initial dataset of FAQ question-answer pairs was provided by AI4EU project partner Institute of Computer Science at FORTH-ICS. This dataset is being adapted and extended for the project needs as part of the WP3 Deliverable 3.6 “Datasets for the StairwAI Chatbot” (initial version submitted in M17, final version is planned in M30) in all project languages. Figure 6 illustrates use of FAQ module in StairwAI prototype.



Figure 6. Examples of FAQ dialogues for English, German, Italian and Spanish languages.

4.4 Guided dialogue that allows user to find AI assets for a specific use case

This version of the chatbot includes a simple guided dialogue to support users that are willing to find AI assets for their use cases. These use case descriptions represent the main “queries” for the Horizontal Matchmaking system (Lombardi et al, 2022). The chatbot can trigger the prototype of the StairwAI matchmaking service by first asking the user whether they are interested in searching for AI assets for a specific use case (currently,



only a fixed pool of AI experts is supported), as depicted in Figure 4. Then, the chatbot collects information about the use case to be used as a query by asking the user about the context and objective for the problem, data availability, and additional requirements (e.g. interpretability).

Figure 7 demonstrates the user interface for this scenario implemented in the prototype. The dialogue consists of four questions, aimed at collecting information about basic aspects of potential AI application. At first the chatbot collects information about the context of the use case (State 19: Matchmaking: context). Then, chatbot asks objectives for the use case (State 25: Matchmaking: use case motivation) and data availability (State 17: data availability) and requirements. Finally, chatbot asks about additional requirements (State 18: Matchmaking: requirements).

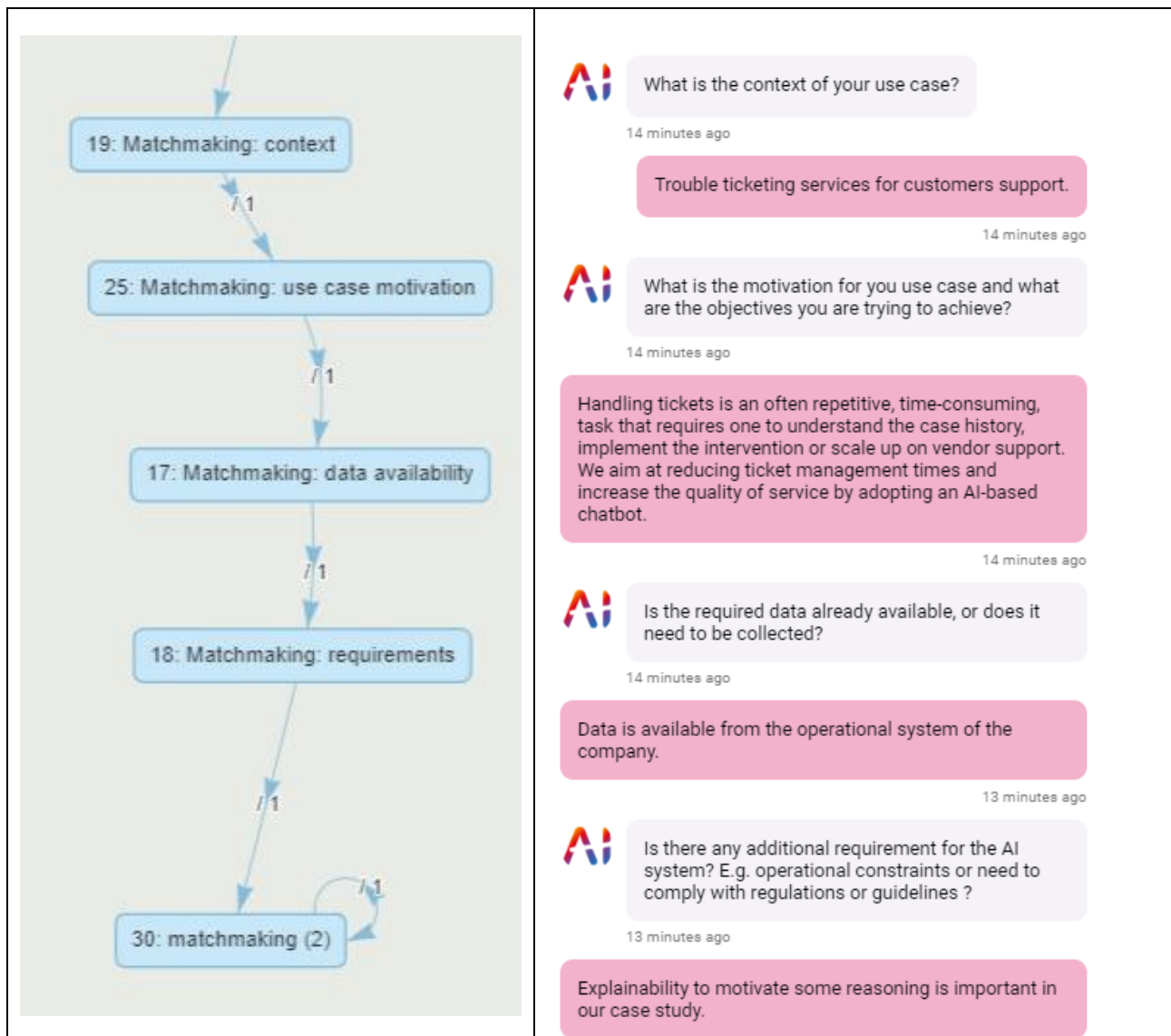


Figure 7. Human-chatbot interaction dialogue graph and demonstration example for the matchmaking scenario.



The information collected by chatbot in these four steps is sent to the matchmaking service developed in WP5, currently as a single piece of text³. Output of the matchmaking service (currently a list of experts) is visualised to the user (State 30: matchmaking). User can look for additional information about expert by clicking on his/her name, or, look through the list of candidates by asking for more results as demonstrated in Figure 8.

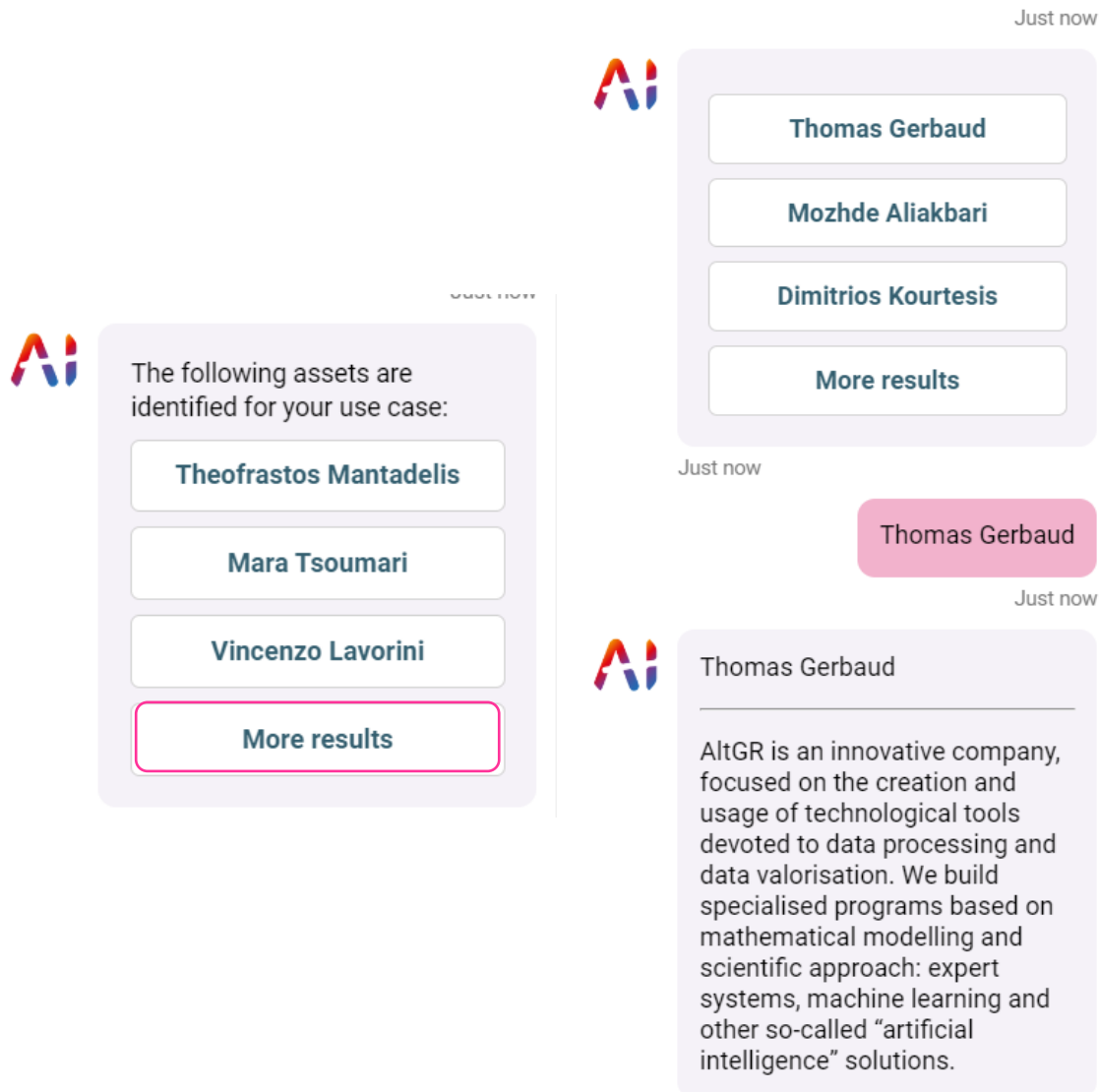


Figure 8. Visualised output from the matchmaking service.

³ Forthcoming versions of the service will be capable of accepting richer input, including multiple sentences and symbolic information (e.g. preferred language or the AI readiness index). Planned extensions also include a system for query refinement with help of labeller presented in Subsection 5.1.



5 NLU/ NLP tools for multilingual human-computer interaction

5.1 Text labelling for matchmaking and keyword extraction

The main purpose of the keyword extraction module (labeler) is to reveal the labels (topics, keywords) related to the content incoming from the user via the conversation with the chatbot. Predicting relevance of these labels is critical for the design of the StairwAI matchmaking algorithm, and it can increase conversation productivity by leading the user toward the correct sources of information (e.g., related job descriptions) via the guided dialog.

Since the set of possible labels (together with their brief definitions to avoid semantical ambiguity) is known, the task for the matchmaking and keyword extraction module is to determine which of these labels can be attached to the textual content incoming from the user. The task can be defined as a multi-class classification problem, but to solve it effectively, we would ideally need access extensive training data. Unfortunately, such large data collections cannot be realistically acquired within the framework of the project, which makes our problem even more challenging. However, there are two promising research directions in NLP for this problem:

1. Zero-shot learning. The zero-shot model is already pre-trained on various texts and can observe even these class labels that were not observed during its training. In our experiments, we have used the transformer model able to return probabilities for each possible label for the input text.
2. The semantical similarity between texts and their labels. This approach consists of two important parts. The first part vectorizes all sentences and all labels (or their definitions) by projecting them into the multi-dimensional semantical space. The second part simply evaluates similarities. The transformer model was used for the vectorization and cosine similarity measure for the evaluation.

We evaluated these two solutions with the small dataset of 2 class labels and 40 instances. Both approaches seem accurate enough to continue working in this direction. Additional testing was performed on a larger dataset (~160 queries, 12 labels), which however lacks ground truth information. More comprehensive investigation and conclusions will be drawn after we evaluate both approaches on a larger testing dataset, which is in the process of being acquired as part of the second round of the open call process. The comprehensive research on matchmaking and keyword extraction was presented in D4.1 (Blaudez et al., 2022). The NLP tool for text labeling is available through one of the web services developed as part of WP5 (queries can be passed via POST over HTTP), and via a simple Graphical User Interface.

5.2 Multilingual and cross-lingual intent detection

This research focuses on the creation of the effective Natural Language Understanding (NLU) module (the key component in chatbot systems), which is responsible for the comprehension of the user's intents. Usually, NLU modules are monolingual: they are trained separately for different target languages, which in turn, require the training data for each such language separately. However, parallel datasets are often unavailable, their preparation is an expensive process, and the number of target languages may still expand in the future. As an alternative, this research aims to investigate multilingual or cross-lingual solutions to this problem with English (*en*), German (*de*), Spanish (*es*), French (*fr*), Italian (*it*), and Latvian (*lv*) languages.



The initial research on multilingual and cross-lingual intent detection was initiated in the frame of the Horizon 2020 COMPRISE project⁴ (Kapočiūtė-Dzikiene et al., 2021). We continue our investigation with the new dataset (new intents, new list of target languages).

5.2.1 A formal definition of the solving task

The intent detection problem is a typical example of a supervised text classification task. Formally, such a task is determined as follows:

Let $D = \{d_1, d_2, \dots, d_n\}$ be a set of documents (questions/statements as input from a user). Let $C = \{c_1, c_2, \dots, c_m\}$ be a set of intents (classes). We have a closed-set classification problem where m is limited, and each c_j is defined in advance. Besides, we solve a single-label classification problem because each $d_i \in D$ can be attached to only one $c_j \in C$.

Let function η be a classification function that maps d_i into their correct classes: $D \rightarrow C$. Let $D^L \subset D$ be a training set of labeled instances (pairs of documents and their correct intents $\langle d_i, c_j \rangle$) used to train a model.

Let Γ be a classification method that, from labeled instances, can learn a model (which is the approximation of η).

Our solving intent detection task aims to offer a classification method Γ that can find the best approximation of η , achieving as high an intent detection accuracy as possible on unseen instances ($D - D^L$) also.

5.2.2 Methodology

The majority of the intent detection research is focused on monolingual problems (e.g., 82 research papers with code with 5 benchmark datasets⁵ and proves the superiority of different transformer-based models. In our research, we investigate transformer-based approaches (denoted as Γ), which were also experimentally proved to be very promising in multilingual and cross-lingual settings:

- **LaBSE + FFNN** approach is composed of two parts: vectorization and classification. For the vectorization, we use the LaBSE⁶ (Language Agnostic BERT sentence transformer) sentence embedding model, supporting 109 languages (including all target languages used in this project) (Feng et al., 2020). This cross-lingual transformer model has mechanisms that allow projecting semantically similar sentences closer to each other in the multi-dimensional vectorization space. Such vectorization output is then mapped into classes via the fully connected dense layer whose weights are adjusted during the training process.
- **LaBSE + COS** is a memory-based approach that simply stores all training instances. During the vectorization, instances are projected into the multi-dimensional space with the LaBSE model. During the testing phase, similarities between the testing instance and all stored training instances are calculated. For it, we use the cosine similarity measure. The testing instance is labeled with the class of the training instance to which it is the most similar. From the theoretical point of view, this approach has to be the most accurate, because it does not perform any generalizations (due to which important information can be lost); however, it is not always suitable due to slower performance (that directly depends on the size of the training dataset) and sensitivity to noise.

These methods were evaluated under different experimental settings:

⁴ <https://www.compriseh2020.eu/>

⁵ Papers with code. Intent Detection. Natural Language Processing. <https://paperswithcode.com/task/intent-detection/>. Accessed on 7th of December, 2022.

⁶ <https://huggingface.co/sentence-transformers/LaBSE>.



- **Monolingual.** The model training and testing were performed with each language (*en, de, es, fr, it, lv*) separately. In these experiments only manually prepared datasets were used.
- **MT-based.** The manually prepared *en* dataset was machine-translated into other target languages (*de, es, fr, it, lv*) and the *googletrans*⁷ python library was used for this purpose. Similarly, as in the monolingual setting, we have trained separate monolingual models for separate languages, but both training and testing were performed on the machine-translated data.
- **Cross-lingual.** This approach uses manually prepared data both for training and testing. But for training, it uses *en* dataset and for testing any other target language.
- **Combined.** In addition to what is already used under the cross-lingual conditions, the machine-translated training dataset for the target language complements the training split. Thus, the training is performed on the manually prepared *en* training data and machine-translated target language training data, and testing is performed on the manually prepared target language data.
- **Train all.** The training is performed on manually prepared *en* training data and machine-translated data to various languages except for the target one. The testing is performed on the manually prepared testing data for the target language.

5.2.3 The data

In our experiments, we have used 6 manually prepared intent detection datasets for *en, de, es, fr, it, and lv* target languages, described in D3.6 (Skadiņa, 2022). All languages contain the same 37 classes (intents). All datasets were randomly split into training and testing subsets keeping 80% and 20% of instances within each class, respectively. The training split of the *en* dataset was translated into other target languages.

The dataset statistics used in different experimental conditions are presented in Table 1. So that we could correctly interpret and evaluate the results, we also calculated these baselines:

majority baseline = $P(c_j)$, where $P(c_j)$ is a probability of a class.

$$\text{random baseline} = \sum P^2(c_j)$$

The method can be considered suitable for the solving task if the achieved *accuracy* = $\frac{\text{correct predictions}}{\text{all predictions}}$ exceed these baselines.

⁷ <https://pypi.org/project/googletrans/>.



Table 1. Datasets used in different experiments.

Experiments	Languages (train→test)	number of training instances	number of testing instances	majority baseline	random baseline
Monolingual	en	386	94	0.074	0.033
	de	192	47	0.064	0.032
	es	193	47	0.064	0.032
	fr	193	47	0.064	0.032
	it	193	47	0.064	0.032
	lv	183	46	0.065	0.031
MT-based	de	386	94	0.074	0.033
	es	386	94	0.074	0.033
	fr	386	94	0.074	0.033
	it	386	94	0.074	0.033
	lv	386	94	0.074	0.033
Cross-lingual	en→de	386	47	0.064	0.032
	en→es	386	47	0.064	0.032
	en→fr	386	47	0.064	0.032
	en→it	386	47	0.064	0.032
	en→lv	386	46	0.065	0.031
Combined	en,de→de	772	47	0.064	0.032
	en,es→es	772	47	0.064	0.032
	en,fr→fr	772	47	0.064	0.032
	en,it→it	772	47	0.064	0.032
	en,lv→lv	772	46	0.065	0.031
Train all	en,es,fr,it,lv→de	1,930	47	0.064	0.032
	en,de,fr,it,lv→es	1,930	47	0.064	0.032
	en,de,es,it,lv→fr	1,930	47	0.064	0.032
	en,de,es,fr,lv→it	1,930	47	0.064	0.032
	en,de,es,fr,it→lv	1,930	46	0.065	0.031



5.2.4 Experiments and results

As presented in Table 1, we tested *LaBSE + FFNN* and *LaBSE + COS* approaches under different experimental settings: *monolingual* (Figure 9), *MT-based* (Figure 10), *cross-lingual* (Figure 11), *combined* (Figure 12), *train all* (Figure 12). Since the initial weights for the *LaBSE + FFNN* method are set randomly and then adjusted during training, the *accuracy* values (along with confidence intervals) for this method were calculated from 5 runs and averaged. To see what works best for different languages, we summarized the results for each language separately: *German* (Figure 13), *Spanish* (Figure 14), *French* (Figure 16), *Italian* (Figure 17), and *Latvian* (Figure 18).

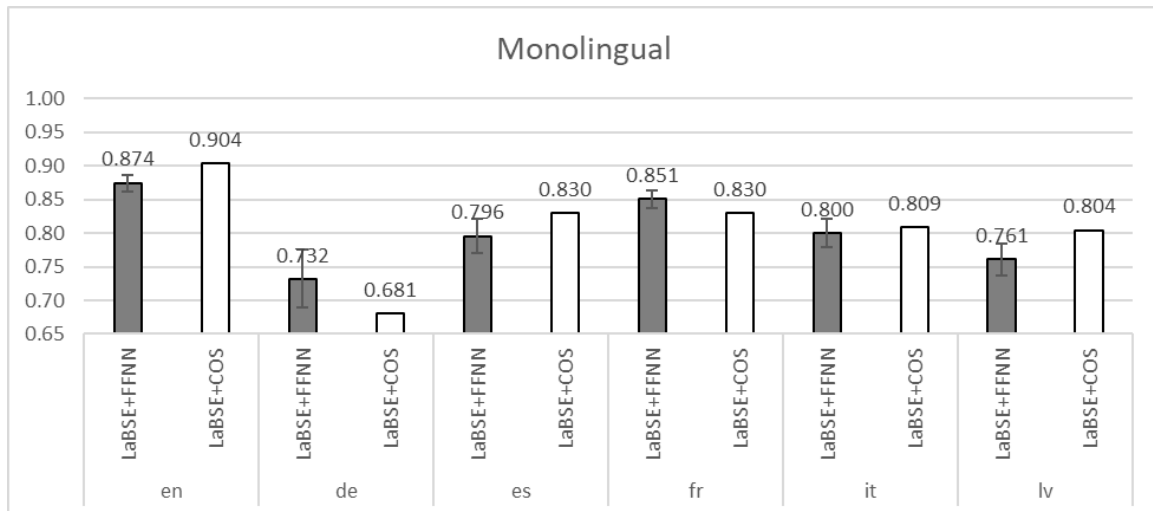


Figure 9. Accuracy values under the *monolingual* settings.

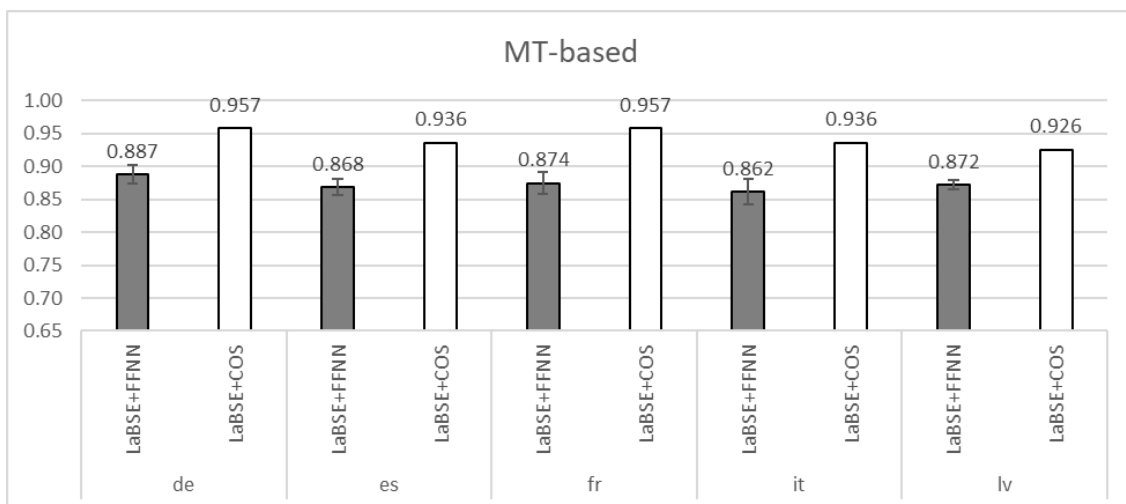


Figure 10. Accuracy values under the *MT-based* settings.



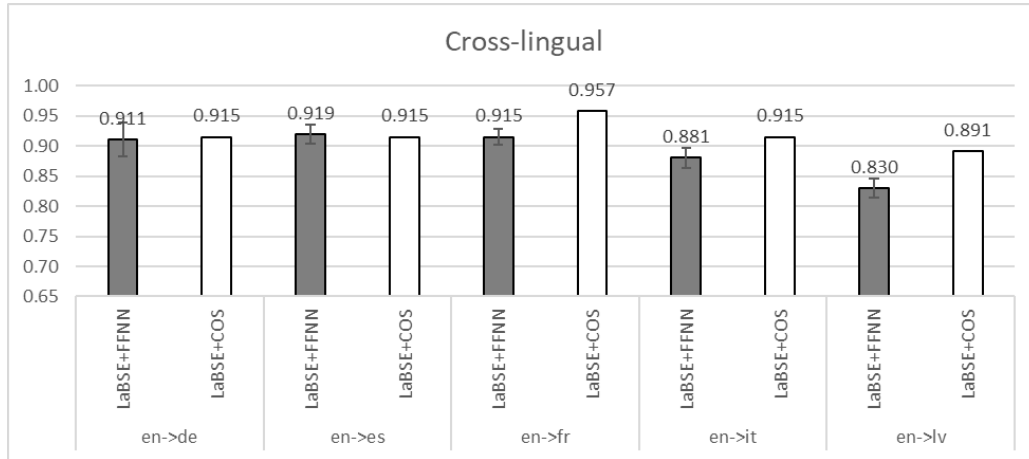


Figure 11. Accuracy values under the **cross-lingual** settings.

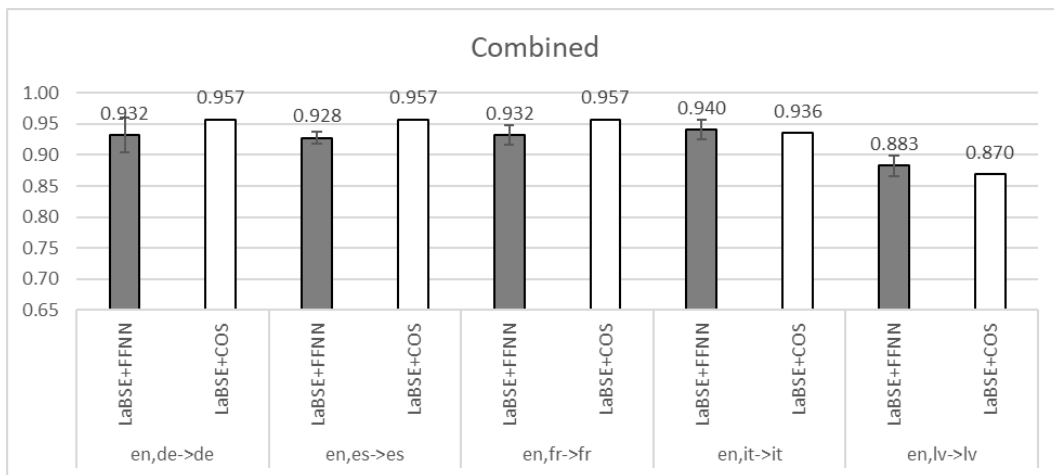


Figure 12. Accuracy values under the **combined** settings.

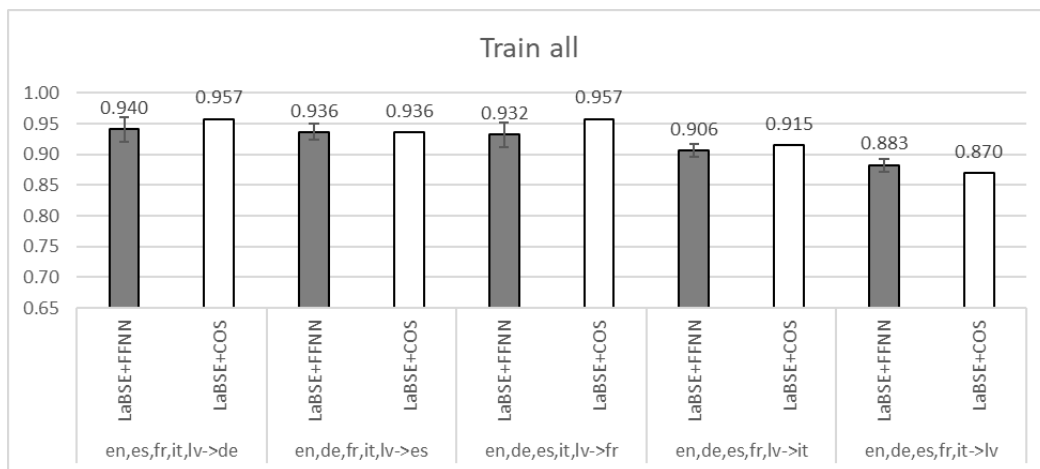


Figure 13. Accuracy values under the **train all** settings.



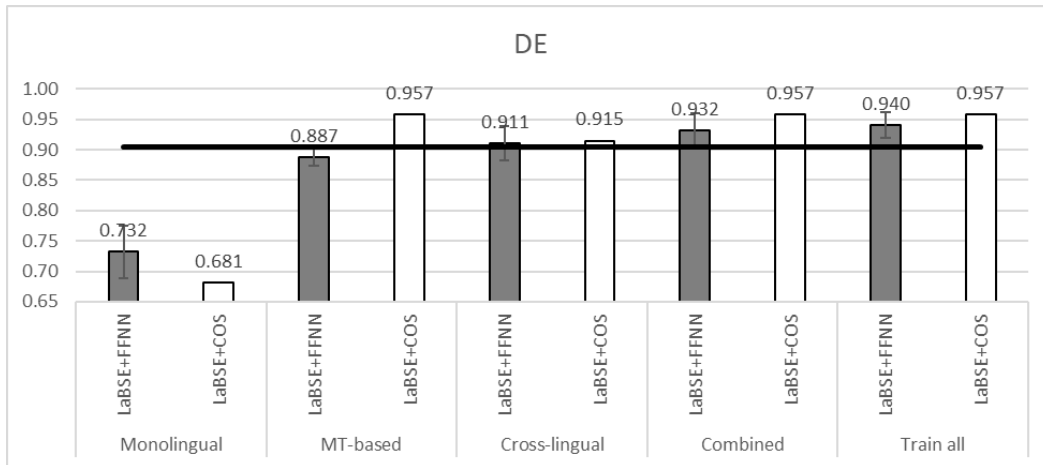


Figure 14. Accuracy values with different approaches for the German (DE) language. The black line is the top line (0.904) that represents the results with the manually prepared **English** dataset.

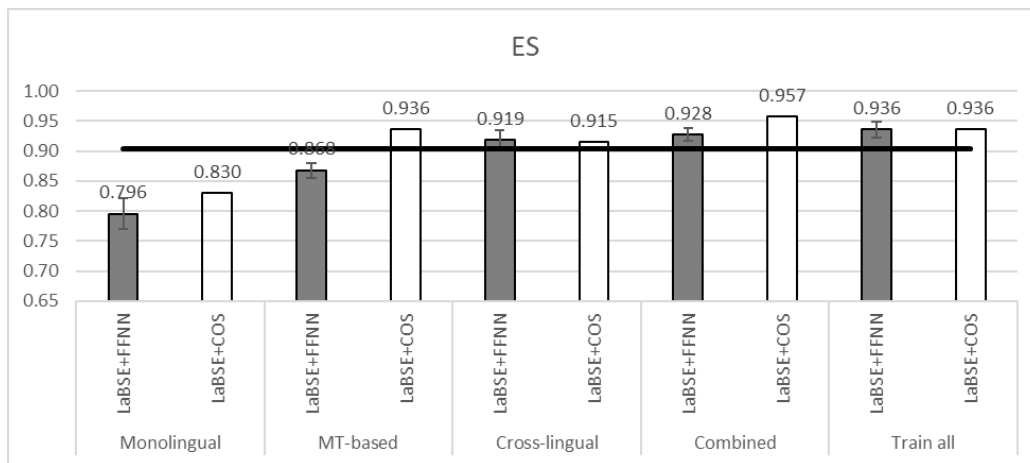


Figure 15. Accuracy values with different approaches for the Spanish (es) language. The black line is the top line (0.904) that represents the results with the manually prepared **English** dataset.

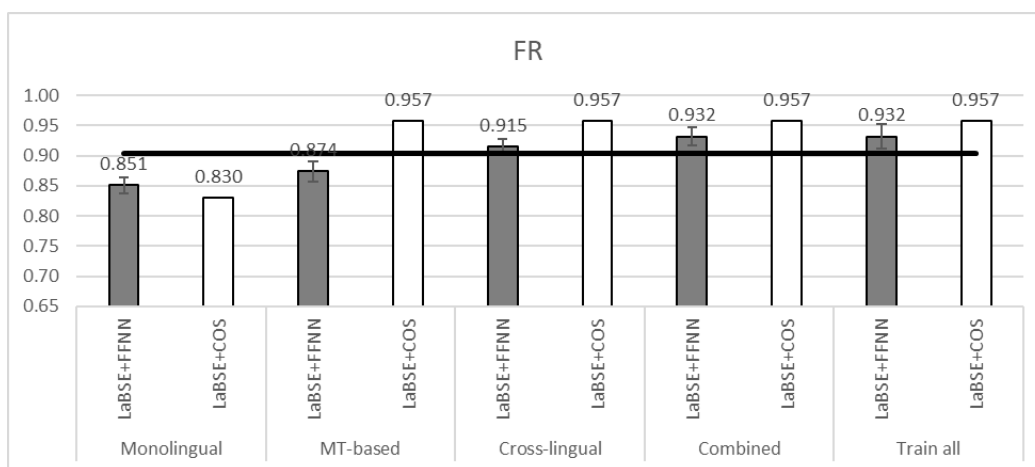


Figure 16. Accuracy values with different approaches for the French (fr) language. The black line is the top line (0.904) that represents the results with the manually prepared **English** dataset.



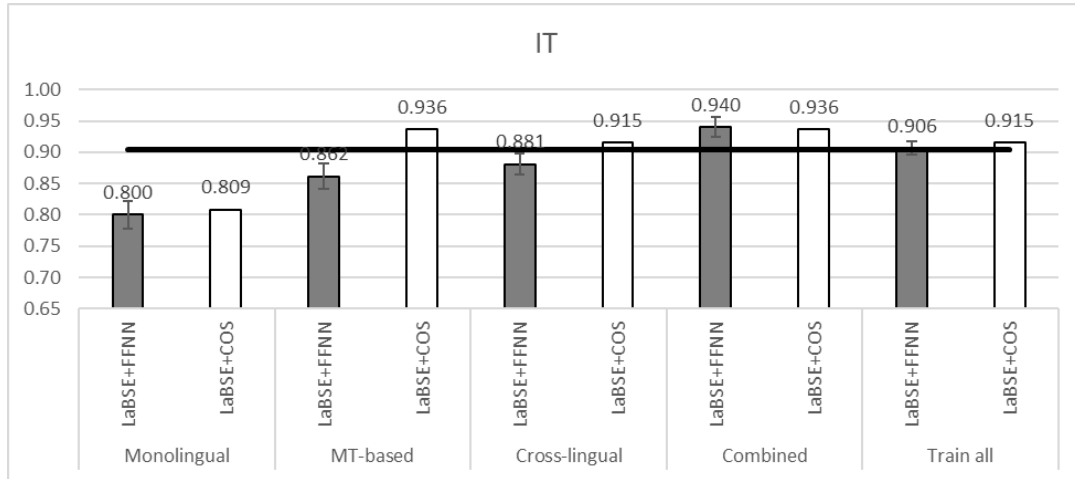


Figure 17. Accuracy values with different approaches for the *Italian (it)* language. The black line is the top line (0.904) that represents the results with the manually prepared *English* dataset.

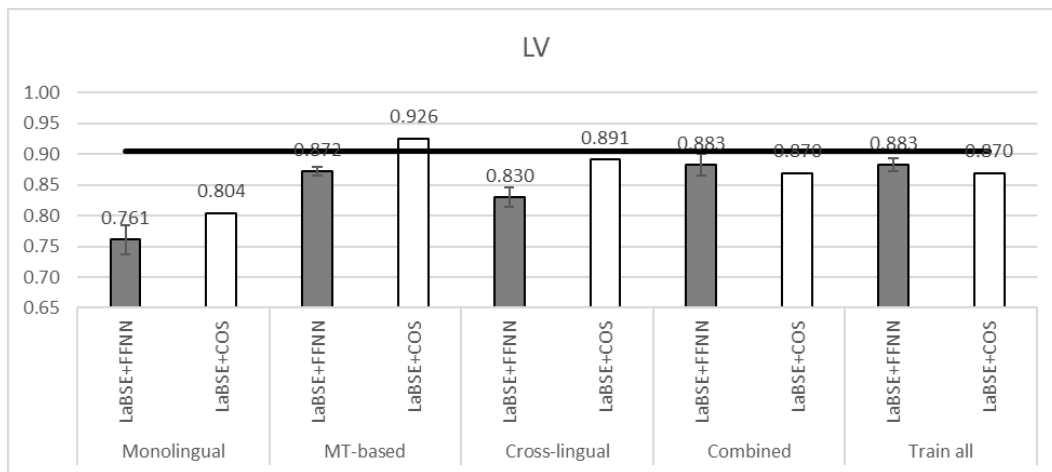


Figure 18. Accuracy values with different approaches for the *Latvian (lv)* language. The black line is the top line (0.904) that represents the results with the manually prepared *English* dataset.

5.2.5 Discussion and conclusions

Zooming into Table 1 and Figure 9 - Figure 18, allows us to make the following conclusions. All tested approaches are suitable for the solving intent detection problem, because exceed random and majority baselines.

In the *monolingual* experiments with the manually prepared dataset, the best results (*accuracy* = 0.904 with *LaBSE + COS* and 0.874 with *LaBSE + FFNN*) were achieved with the *en* language. There are two reasons for it: 1) the *en* language is the best covered in the LaBSE transformer model (see Feng et al., 2020), and 2) the *en* training dataset has twice more data compared to other languages. The second reason becomes even more apparent in Figure 10 with the machine-translated data: the translations are accurate and the accuracy above 0.9 is reached with all target languages. However, the results in Figure 9 and Figure 10 cannot be directly comparable due to the different test sets (in terms of their sizes and data).

The *cross-lingual*, *combined*, and *train all* experiments (Figure 11) allow achieving high accuracy levels above 0.9 except for the *lv* language. It is probably due to the first reason: the *lv* language is the least covered (of all target languages) in the LaBSE transformer model. It is rather difficult to make the conclusion about which



method (i.e., *LaBSE + FFNN* or *LaBSE + COS*) is more suitable for the solving problem (except for the *MT-based* where *LaBSE + COS* is always superior).

If looking at each language separately Figure 14 - Figure 18 we can see that with the right approach it is even possible to exceed the top line (the best accuracy equal to 0.904 achieved with the original manually prepared *en* dataset) except for the *lv* language. Since MT-based results are not directly comparable to all other approaches, we will not make any conclusions based on them. Thus, for the *de* language *combined* and *train all* approaches are the best; for *es – combined*, for *fr – combined* and *train all*, for *it – combined*, and for *lv – cross-lingual*. Hence, the *combined* approach seems to be the most suitable for the majority of languages. The *cross-lingual* and multilingual (*combined*, *train all*) approaches are always better (for all languages) compared to the *monolingual* one.

5.3 Knowledge datasets

The chatbot needs to interact with the AI Asset Knowledge representations developed in WP3. This subsection describes those representations.

5.3.1 The StairwAI AI Asset Conceptual Model

StairwAI's AI Asset Conceptual Model is an ontology (Buxons et al, 2021). It aims to act as the common knowledge representation for all the project concepts, establishing the classes (items), attributes of these classes and relationships between the classes, creating a common knowledge backbone for all other modules in the StairwAI platform. It also aims to ease interoperability with other platforms, such as AI4EU, BonsEyes or Open ML.

At the time of writing, the second version of the AI Asset Conceptual Model has been published (In RDF format). A complete documentation has been developed and can be consulted at <https://stairwai.gitlab.io/wp3/ontology/>. It can also be imported in ttl format using the (Permanent URL <https://stairwai.gitlab.io/wp3/ontology/>).

5.3.2 The StairwAI AI Asset Management System

StairwAI's Asset Management System (AMS for short) is a structured graph database that allows the insertion, modification and inference over the different assets that StairwAI supports, The aim of this graph database is to represent the relationships between the different StairwAI assets and their main attributes. These relationships correspond to the one defined in the StairwAI's AI Asset Conceptual Model, which is imported by the database, in order to use it as a backbone representation of the available classes, attributes and relations.

Capabilities

This second draft follows a similar structure as the first one, the user is able to find three main capabilities. A complete documentation was generated and can be consulted at <https://gitlab.com/stairwai/wp3/knowledge-representation/-/tree/main/serverhosted-version/stairwai-ams-api>. These three main capabilities are composed by:

- the *getters* which will extract all the different static data in the database, inside *getters'* category, we can distinguish between: (1) non-instantiable class getters, (2) general data getters (such as, general AMS classes, attributes or relations getters) and, (3) specific node information getters.



- Another StarwAI AMS API capability is the one composed by the *methods to insert, modify or delete* each one of the AI Asset instances of the platform.
- Finally, the last capability is the one composed by the implemented *queries*.

The StairwAI AMS has been wrapped in two different distributions (both available in the StairwAI GitLab repository⁸) one distribution has been wrapped in a docker container with all the dependences required. Therefore, it is quite easy to install in any machine following guidelines in the repository. This distribution was thought to be run locally, enabling local testing of AMS capabilities for all users of the service.

On the other hand, there are the server-designed distribution. This distribution was designed to be deployed in a distributed environment. We can distinguish between the Graph database docker container which starts the graph database and open two different ports (for http and bolt protocols) to be able the users to connect with it. Apart from the database container, this distribution has two different API, one for common users and one with administrator extra functionalities.

6 Validation of NLP models

This task aims to compile a variety of AI neural network architectures and models into dedicated AI instruction sequences for the HUA NPU/CPU/GPU in an offline setting, with data and weight rearrangement for optimized performance. The task includes creation of NLP inside terminal devices and MT combined to accelerate low-tech users input requests running on mobile devices.

Moreover, this task aims to explore and validate applications of NLP models that would be useful for the StairwAI chatbot use case on mobile devices. In addition, we investigate user engagement and trust towards the chatbot, chatbot usability, acceptability and also ways to improve the chatbot's abilities of conversational user interaction. To this end, prototypes of NLP based applications are produced and studied. In this section, the research findings are presented in detail.

6.1 Telegram Rasa

Since new NLP research results are being published continuously, the task started with studies of recent research in various NLP areas that could be important for users accessing a chatbot via a mobile user interface. To be able to validate new concepts made available in state-of-the-art research on NLP rapidly, a prototyping environment based on the open-source Rasa conversational AI framework⁹ was set up. Also Telegram modules and APIs are utilized to validate effects of NLP models when accessed via the Telegram app on a mobile device. Doing this allows us to test a broad spectrum of NLP models rapidly without interfering with the chatbot development in task T4.3 but sharing insights gained in task T4.4 within WP4. Figure 19 shows the guiding concept for our early activities in T4.4 whereas Figure 20 presents the interaction pipeline between Telegram and Rasa.

⁸ <https://gitlab.com/stairwai/wp3/knowledge-representation>

⁹ <https://rasa.com/product/rasa-platform/>



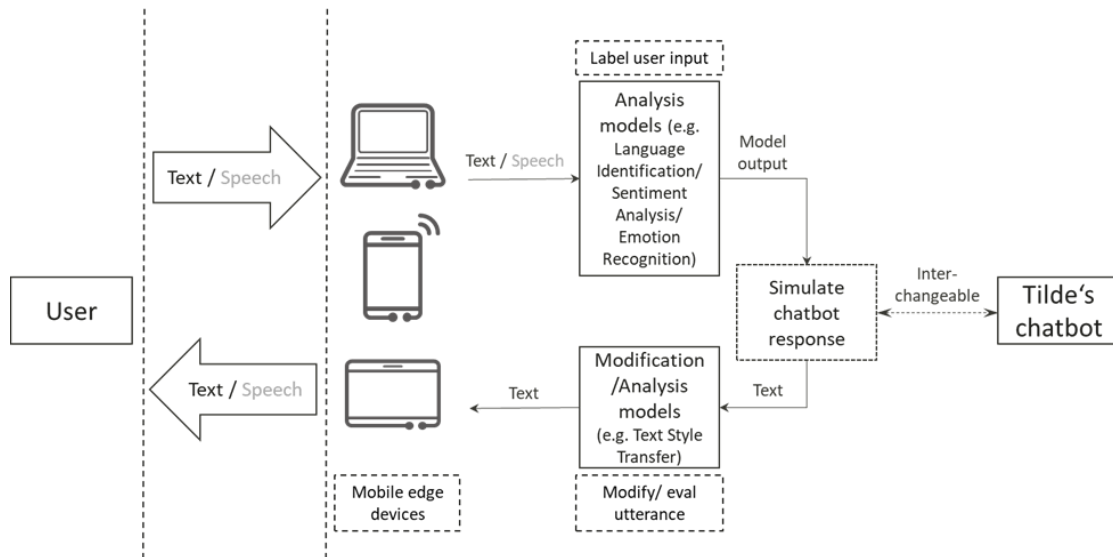


Figure 19. Guiding concept for our early activities in Task 4.4.

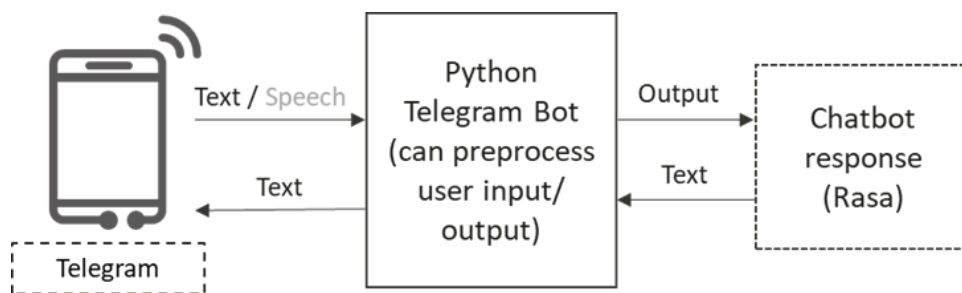


Figure 20. Interaction pipeline between Telegram and Rasa.

6.1.1 Chat Summary

The ability to summarize a conversation is proven to be a valuable active listening skill for chatbots to increase trust and engagement with the user. To test the chat summarization skill in StairwAI scenario, a simple replica of the StairwAI chatbot with limited functionality is produced using Rasa. This chatbot is used to demonstrate how the interaction would unfold with a chatbot possessing this skill. The Figure 21 below shows the chatlog and the resulting summary of the conversation between the user and the chatbot. The chat summarization model¹⁰ used in this implementation can be found on Hugging Face¹¹. The *philschmid/bart-large-cnn-samsum* model is trained using messenger-like conversations with summaries from the SAMSum¹² Corpus.

¹⁰ [philschmid/bart-large-cnn-samsum · Hugging Face](#)

¹¹ [Hugging Face – The AI community building the future.](#)

¹² [SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization - ACL Anthology](#)



Chatlog

You: hi

Bot: Hi! I am StairwAI bot, and I am here to connect you to AI resources.

You: What can you do

Bot: I can assess your AI readiness level, suggest assets for your problem or we can chat about AI and related technologies

You: who funds the project

Bot: The project has received funding from the European Union's Horizon 2020 research and innovation program.

You: when did the project started

Bot: In January 2021



Figure 21. Chat summarization.

For the StairwAI use case, the user can explicitly request the chatbot to show a summary of the chat whenever they need it or the chatbot can automatically produce the summary at the end of the interaction so that the user can have an overview of what was communicated to the chatbot.

6.1.2 Document Question-Answering

In line with the StairwAI scenario of the chatbot asking questions to extract information about the user's needs, we further explored an additional feature that could assist the users in their communication with the chatbot, which is document-based question-answering. Figure 22 below shows a screenshot of the user-chatbot interaction for document question-answering.



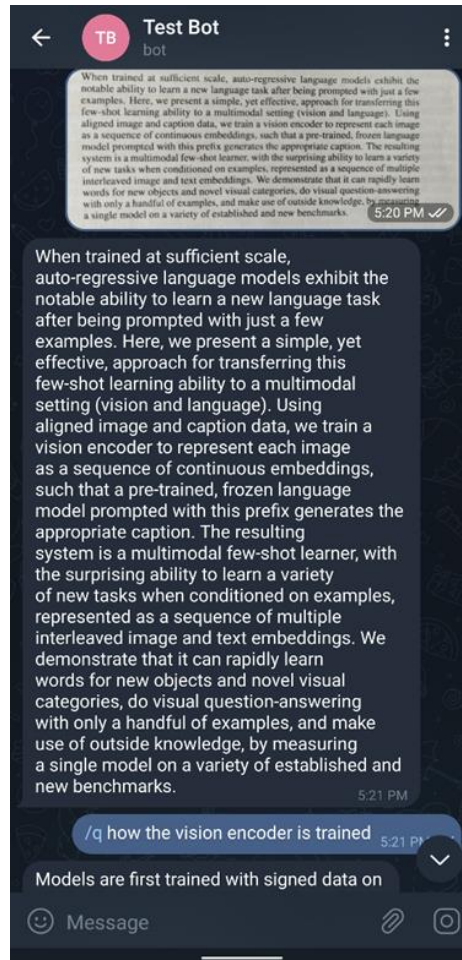


Figure 22. Document Question-Answering on mobile (Telegram).

First, the user uploads an image of the document, then the text embedded in the image is automatically extracted using Python-Tesseract¹³, an optical character recognition (OCR) tool for Python. It is a wrapper for Google’s Tesseract-OCR Engine. After the text is extracted, a Python library for processing textual data called TextBlob¹⁴ is used to correct potential spelling mistakes. The chatbot is then able to answer questions from the user based on the extracted text. The Question-Answering model¹⁵ used in this implementation can be found on Hugging Face. It is a distilled version of the *deepset/roberta-base-squad2* model called *deepset/tinyroberta-squad2*. This model was chosen due to its inference speed, it runs at twice the speed of the base model and has a comparable prediction quality to the base model.

For the StairwAI use case, this could be a feature for the chatbot to automatically extract information from technical documents that the user uploads to answer questions related to the user’s needs.

¹³ [pytesseract · PyPI](#)

¹⁴ [TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation](#)

¹⁵ [deepset/tinyroberta-squad2 · Hugging Face](#)



6.2 Papr Readr Bot demonstrator

The Papr Readr Bot demonstrator was developed as a proof of concept using the guiding concept from Figure 19, it encompasses many of the StairwAI general NLP tasks, such as intent recognition and question answering. A demonstration paper (Foo et al, 2022) presenting the Papr Readr Bot was published at the Conversational User Interfaces 2022 conference and showcased publicly during the conference. Feedbacks from the conference participants, mostly researchers and PhD students were positive. They liked the demonstrator and praised how useful it would be for assisting them in reading papers.

The code was also made available through a public Github repository¹⁶. The demonstrator showcased the utility of various NLP models in an assistive conversational user interface as shown in Figure 23, which helps users to access and filter information in (technical) research papers.

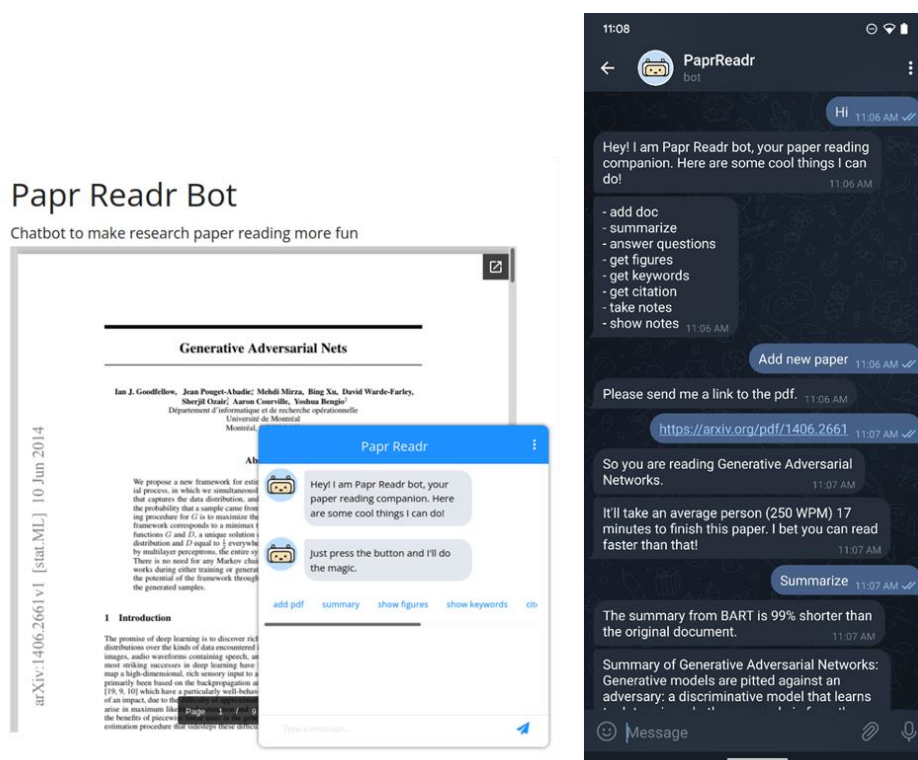


Figure 23. Web application interface and Telegram (mobile) chatbot interface.

¹⁶ [Papr Readr Bot code](#)



7 Summary and next steps

This deliverable presents StairwAI chatbot prototype – main scenarios and functionality. The deliverable introduces the architecture, main features and NLP tools that are or will be included in StairwAI chatbot prototype.

7.1 Summary of results

The main result described in this deliverable is a StairwAI chatbot prototype¹⁷ that demonstrates functionality for three scenarios:

- AI readiness assessment,
- Simple matchmaking through guided dialog using matchmaking service developed in WP5,
- Frequently asked questions.

The prototype demonstrates guided, free and combined dialogue scenarios. It includes functionality for multilinguality, intent detection, machine translation and access (querying) to external services (e.g., matchmaking).

Where it concerns interaction between the chatbot platform and knowledge datasets developed in WP3, the first prototype of the StairwAI's Asset Management System was published in Month 12, providing means the Chatbot to interact with its knowledge graph by using graph database queries. The current results summary includes the second version of StairwAI's AI Asset Conceptual Model (due at Month 18), and the second version of the AMS prototype (due at Month 24).

Finally, a prototyping environment based on Rasa and Telegram was set up to validate the applications of NLP models in StairwAI related scenarios on mobile devices. Implementations for chat summarization and document-based question answering are explored and tested through this setup. The Papr Readr Bot demonstrator that encompasses many of the StairwAI general NLP tasks was developed as a proof of concept. Furthermore, a conference paper on the developed chatbot was published and Papr Readr Bot was presented at the 2022 Conversational User Interfaces conference.

7.2 Next steps

The main goal of WP4 for the last year of the project is to deliver final version of StairwAI Chatbot Prototype by M30 (end of June, 2023). This includes further development of multi-lingual user interface and domain adaptation of NLU/NLP tools where it is required by use cases and scenarios. We will also continue refinement of the user scenarios and dialogue flows in close collaboration with WP5.

Finally, we are planning user studies to validate the research ideas for the StairwAI chatbot use case. Through the insights gathered from the user studies, we also expect to provide further guidance to improve the final demonstrator by M30.

¹⁷ The *StairwAI chatbot prototype* is publicly available at <https://va.tilde.com/api/prodk8sbotstair1/chat/default.htm>



Bibliography

1. Eric Blaudez, Miquel Buxons Vives, Jurgita Kapočiūtė-Dzikienė, Michele Lombardi, Inguna Skadiņa, Javier Vázquez-Salceda. 2022. D4.1. StairwAI Chatbot MVP.
2. Miquel Buxons Vives, Javier Vázquez-Salceda. 2022. D3.3. WP3 - Platform knowledge and community organisation.
3. Miquel Buxons, Javier Vázquez-Salceda, Javier Farreres. D3.4. 2022. Design of the knowledge representation in the StairwAI AI Asset Management System - 2nd version.
4. Jurgita Kapočiūtė-Dzikienė, Askars Salimbajevs and Raivis Skadiņš. 2021. Monolingual and Cross-Lingual Intent Detection without Training Data in Target Languages. *Electronics* 2021, 10(12), 1412.
5. Michele Lombardi, Eleonora Misino, Roberta Calegari, Ville Tenhunen, Begüm Genç. 2022. D5.1. Matchmaking Algorithms, 1st version.
6. Inguna Skadiņa. 2022. D3.6. Datasets for the StairwAI Chatbot.
7. Urszula Sobek, Marta Calero, Elena Japundzic. 2022. D7.2. First Call Announcement and Guide for Applicants.
8. Ville Tenhunen, Andrea Borghesi, Roberta Calegari, Malika Elkharraz, Michele Lombardi, Marco Rorro, Renato Santana. 2022. D2.2. Requirements for AI-on-demand platform.
9. Feng F.; Yang, Y.; Cer, D.; Arivazhagan, N.; Wang, W. Language-agnostic BERT Sentence Embedding. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2020*, pp. 878-891. <https://aclanthology.org/2022.acl-long.62>
10. Michelle Xiao-Lin Foo, Luca Della Libera and Ilhan Aslan. "Paper Readr Bot: A Conversational Agent to Read Research Papers". *International Conference on Conversational User Interfaces*, July 26–28, 2022, Glasgow, UK



8 Appendices

Chatbot integration in StairwAI Website

The code below allows easily to integrate the StairwAI bot prototype in any page where the virtual assistant is necessary.

```
<div id="webchat"></div>
<script>
function onChatOptions() {
  return {
    params : {
      // Optional parameters
      /*mainColor: '#643d74',
      botIntroEn: 'Intro',
      botIntroLv: 'Intro',
      showGreetingsBanner: 'true',
      botNameEn: 'StairwAI Chatbot',
      botNameLv: 'StairwAI Chatbot',
      botShowIntro: 'true',
      chatPlacementType: 'side-nav',
      chatWidth: 21,
      opened: 'false',*/
    },
    style : {
      // Optional parameters
      /*userTextColor: '#000000',
      botTextColor: '#000000',
      botAvatarUrl: 'https://va.tilde.com/api/prodk8sbotstair1/avatar/AR5MS2G8VFW4.png',
      botMessageColor: '#f5f2f8',
      userMessageColor: '#f2b2cc',*/
    }
  };
};
</script>
<script type="text/javascript"
src="https://va.tilde.com/api/prodk8sbotstair1/chat/botchat.min.js"></script>
```

