

Project Acronym: STAR
Grant Agreement number: 956573 (H2020-ICT-2020-1 – Research and Innovation Action)
Project Full Title: Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines
Project Coordinator: INTRASOFT International



Funded by the Horizon 2020
Framework Programme of the
European Union

DELIVERABLE

D4.5 – Active Learning Systems and Techniques Final Version

Dissemination level	PU -Public
Type of Document	Demonstrator
Contractual date of delivery	31/03/2023
Deliverable Leader	JSI
Status - version, date	Final – v1.0, 12/04/2023
WP / Task responsible	WP4
Keywords:	Active learning

This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956573. It is the property of the STAR consortium and shall not be distributed or reproduced without the formal approval of the STAR Management Committee. The content of this report reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

Executive Summary

This document provides an overview of the activities and the results achieved in WP4 Safe, Transparent, and Reliable Human-Robot Collaboration within Task 4.3 Human-Robot Interactions for Active Learning.

The main activities carried out during T4.3 are:

- 1) research in active learning to facilitate human robot interactions and accelerate knowledge acquisition;
- 2) developing prototype systems that enable human robot interactions;
- 3) implementing active learning techniques within STAR use cases;
- 4) implementing NLP techniques and conversational clients.

This document describes the developed active learning prototypes and approaches applicable to STAR Use cases.

The WP4 partners conducted an extensive scientific work presented in some publications.

In comparison to Deliverable D4.4 Active Learning Systems and Techniques – Initial version, WP4 partners conducted several significant improvements to the initial active learning prototypes, such as:

- further research and experiments of the benefits of the active learning approach,
- exploitation of the developed techniques in other STAR use cases,
- improvement of the NLP techniques in the industrial environment.

Within the scope of Task 4.3 in WP4 partners published scientific articles and a book chapter.

Deliverable Leaders:	JSI: Jože Rožanec, Klemen Kenda, Patrik Zajec, Inna Novalija
Contributors:	R2M: Rubén Alonso
Reviewers:	UBI, GFT
Approved by:	Charalampos Ipeksidis (INTRA)

Document History			
Version	Date	Contributor(s)	Description
0.1	20/02/2023	JSI	Table of contents
0.2	03/03/2023	JSI	First draft
0.3	10/03/2023	R2M, JSI	Second draft (integration of partners' input)
0.4	17/03/2023	JSI	Third draft (content refinement)
0.5	21/03/2023	JSI	Final draft
0.6-0.9	31/03/2023	Reviewers comments	Addressed
1.0	12/04/2023	INTRA	QA and creation of the final submitted version

Table of Contents

EXECUTIVE SUMMARY	2
TABLE OF CONTENTS.....	4
TABLE OF FIGURES.....	5
LIST OF TABLES.....	6
DEFINITIONS, ACRONYMS AND ABBREVIATIONS	7
1 INTRODUCTION.....	8
2 ACTIVE LEARNING STRATEGIES AND TECHNOLOGIES.....	9
3 STATE OF THE ART IN QUALITY INSPECTION.....	12
4 APPLICATION TO STAR USE CASES.....	13
4.1 PCL USE CASE.....	13
4.1.1 <i>Implementation</i>	13
4.1.2 <i>Experiments</i>	13
4.1.3 <i>Evaluations</i>	15
4.1.4 <i>Demos and proof of concept</i>	17
5 NLP FUNCTIONALITIES	20
5.1 EARLY PROOFS OF CONCEPT	20
5.2 FROM STT AND TSS TO CHATBOTS.....	21
5.3 CONVERSATIONAL AGENTS.....	22
5.4 DEMOS AND PROOF OF CONCEPTS.....	23
5.5 INTEGRATION AND FUTURE NLP ACTIVITIES	24
6 CONCLUSIONS.....	25
REFERENCES	26

Table of Figures

FIGURE 1: TAXONOMY OF ACTIVE LEARNING APPROACHES. THE IMAGE WAS TAKEN FROM [20]	9
FIGURE 2: SCHEMATIC DIAGRAM OF ACTIVE LEARNING.	10
FIGURE 3 THREE ORACLE SETTINGS. THE IMAGE WAS TAKEN FROM [19]	14
FIGURE 4 EQUATION 1: PLATT CLASSIFIER CALIBRATION LOGISTIC MODEL.	15
FIGURE 5 SNIPPET FROM A JUPYTER NOTEBOOK, WHERE ACTIVE LEARNING EXPERIMENTS WERE IMPLEMENTED.....	18
FIGURE 6 ACTIVE LEARNING REST API DEMO.	19
FIGURE 7 WEB SPEECH API DEMO 2.....	20
FIGURE 8 CONVERSATIONAL AGENTS MODULE.....	22
FIGURE 9 NLP DEMO: STAR INTERVIEWER.....	23
FIGURE 10 NLP DEMO: CONVERSATIONAL INTERFACE ON OCCUPATIONS	24

List of Tables

TABLE 1 PROPOSED EXPERIMENTS TO EVALUATE THE BEST ACTIVE LEARNING SETTING REGARDING HOW IT INFLUENCES THE MODELS’ LEARNING AND ITS IMPACT ON THE MANUAL REVISION WORKLOAD. THE TABLE WAS TAKEN FROM [19]......14

TABLE 2 THE TABLE DISPLAYS THE MEAN AUC ROC VALUES FOR FIVE MACHINE LEARNING MODELS, AVERAGED ACROSS TEN FOLDS. THE RESULTS HIGHLIGHT THE IMPACT OF VARIOUS ACTIVE LEARNING STRATEGIES ON THE MODELS’ PERFORMANCE OVER TIME, COMPARING THE FIRST QUANTILE (Q1) TO THE LAST QUANTILE (Q4) OF THE ACTIVE LEARNING POOL. SOFT LABELLING WAS APPLIED USING TWO PROBABILITY THRESHOLDS (0.95 AND 0.99). THE BEST RESULTS ARE BOLDED, AND THE SECOND-BEST RESULTS ARE DISPLAYED IN ITALICS. THE TABLE WAS TAKEN FROM [19]......15

TABLE 3 THE MEAN AUC ROC VALUES WERE COMPUTED ACROSS TEN TEST FOLDS FOR FIVE MACHINE LEARNING MODELS TO DETERMINE HOW THEY LEARN OVER TIME (Q1 VS. Q4) UNDER THE EXPERIMENT 2 SETTING. WE ALSO EXAMINED WHETHER THE DIFFERENCES WERE STATISTICALLY SIGNIFICANT AT A P-VALUE OF 0.95 (DS(p=0.95)). THE RESULTS ARE PRESENTED BELOW, WITH THE BEST OUTCOMES HIGHLIGHTED IN BOLD AND THE SECOND-BEST RESULTS DISPLAYED IN ITALICS. THE TABLE WAS TAKEN FROM [19]......16

TABLE 4 THE TABLE PRESENTS THE PROPORTION AND QUALITY OF SOFT LABELLING UNDER DIFFERENT SETTINGS, WITH A PREDICTED PROBABILITY CUT-OFF VALUE OF $p=0.95$. SL (%) SHOWS THE PERCENTAGE OF SOFT ANNOTATED DATA INSTANCES RELATIVE TO THE TOTAL, SL OK (%) INDICATES THE PERCENTAGE OF CORRECTLY SOFT ANNOTATED INSTANCES, ML SL OK (%) REPRESENTS THE PERCENTAGE OF SOFT ANNOTATED DATA INSTANCES THAT WOULD BE CORRECTLY ANNOTATED BASED ON THE ML MODEL SCORE, AND SSIM SL OK (%) INDICATES THE PERCENTAGE OF SOFT ANNOTATED DATA INSTANCES THAT WOULD BE CORRECTLY ANNOTATED BASED ON THE SSIM SCORE. THE TABLE WAS TAKEN FROM [19]......17

TABLE 5 THE TABLE PRESENTS A COLLECTION OF ENDPOINTS WE IMPLEMENTED FOR THE ACTIVE LEARNING USE CASE......19

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AI	Artificial Intelligence
AL	Active Learning
BoW	Bag of Words
GAN	Generative Adversarial Networks
JSGF	Java Speech Grammar Format
LR	Logistic Regression
MAP	Mean Average Precision
MLP	Multilayer Perceptron
NLP	Natural Language Processing
RF	Random Forest
ROC AUC	Receiver Operating Characteristic, Area Under Curve
SA	Sentiment Analysis
SOTA	State of the Art
STT	Speech-To-Text
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency
TTS	Text-to-Speech
WP	Work Package

1 Introduction

Task 4.3 Human-Robot Collaborations for Active Learning targets STAR's objective to explore and implement active learning techniques that can contribute to human robot collaboration and accelerate knowledge acquisition.

The main development of task 4.3 has been focused on PCL use case, on quality inspection of Phillips manufactured products (Philips Consumer Lifestyle BV). Quality control allows companies to verify the product's conformance to requirements and specifications and thus build customer satisfaction and the brand's reputation.

The initial version of active learning prototypes has been described in STAR deliverable D4.4 Active Learning Systems and Techniques Initial Version (M15).

Artificial Intelligence (AI) can automate visual inspections and reduce inspection times while ensuring consistent evaluation of all products. In the STAR project we aimed to address the automated visual inspection ensuring manual revision is only used to examine manufactured pieces for which the model is uncertain. We propose that active learning can be used to improve the classification models continuously by leveraging newly labelled data in the manufacturing industry.

Furthermore, this deliverable presents the advances with respect to Natural Language Processing demonstrators, describing the sum-ups the carried NLP activities, and providing the system prototype for PCL use case.

The document is structured in the following sections:

- **Section 1** provides an overview of the scope and the structure of this document.
- **Section 2** includes a review of active learning strategies and technologies.
- **Section 3** specifies the state of the art in the quality inspection that strongly relates to the developed Task 4.3 active learning prototype for STAR use cases.
- **Section 4** provides the implementation activities and application to STAR use cases.
- **Section 5** describes the natural languages processing aspects related to technologies developed for STAR use cases.
- **Section 6** summarizes the activities and concludes the document.

2 Active Learning Strategies and Technologies

Artificial Intelligence is a field of study that focuses on developing computers and machines that can imitate the decision-making and problem-solving abilities of intelligent beings. One sub-field of AI is Machine Learning, which involves using algorithms to learn from data. The effectiveness of a machine learning model depends on how informative the data is, and Active Learning is an approach that seeks to improve the quality of the data used for learning. Active Learning can be characterized based on four criteria, which determine the approach used to select informative instances for labelling.

Supervised machine learning is a popular approach where algorithms learn the mapping between input feature values and expected outcomes. Active Learning is based on the assumption that unlabelled data is abundant, labelling is expensive, and models' generalization errors can be minimized by carefully selecting new input instances to train the model. Strategies and criteria have been developed to find the best-unlabelled instances to achieve this goal. In a supervised Active Learning setting, an unlabelled data instance is obtained, and an oracle provides a label. The labelled data instance is then used to train the machine learning model.

The resulting model can be used to classify or predict new data, and Active Learning can be used to continuously improve the model's performance. Different Active Learning approaches exist, and they can be classified based on how the data is generated, processed, how many instances are queried at a time, and what machine learning problem is being solved. Active Learning can be applied to various problems, but supervised Active Learning has received more attention in the scientific community. We provide a taxonomy of Active Learning in Figure 1.

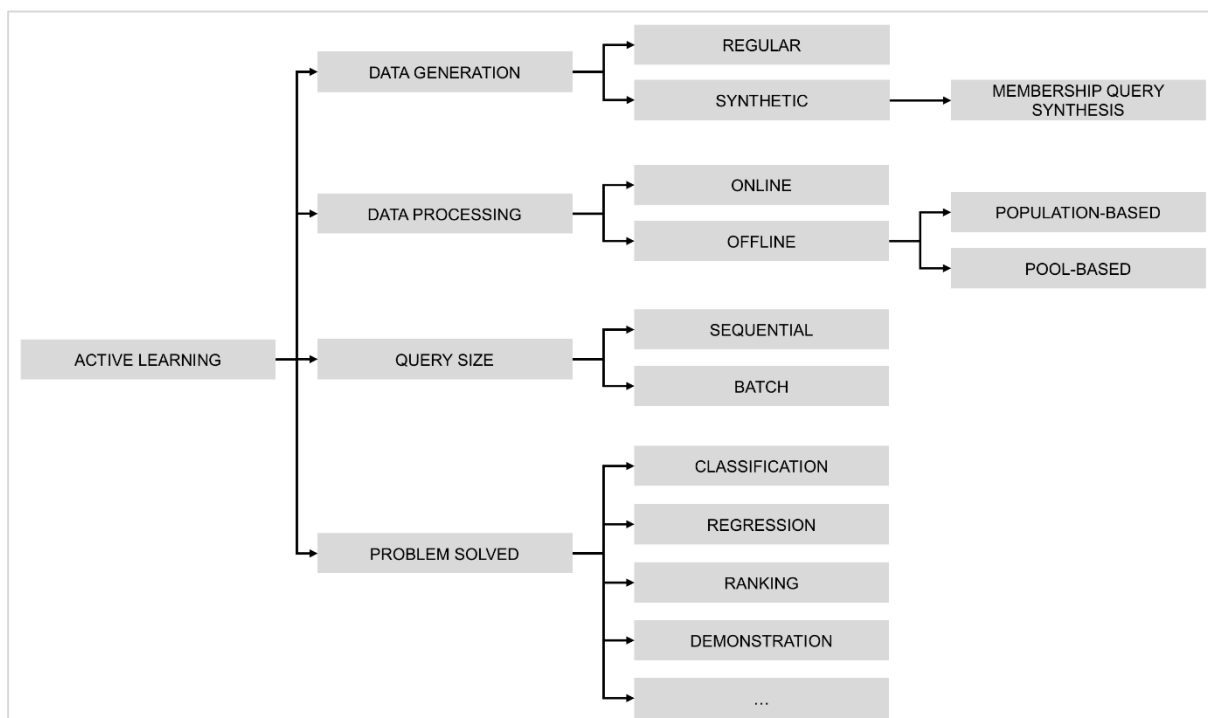


Figure 1: Taxonomy of active learning approaches. The image was taken from [20]

In a supervised Active Learning setting, the goal is to obtain a labelled dataset that can be used to train a machine learning model. To do this, the Active Learning system needs to select unlabelled data instances that are most informative to the model's learning process. These unlabelled instances are called queries, and the Active Learning system needs to decide which ones to request labels for from the oracle. Once the oracle provides the labels for the selected queries, the resulting labelled data instances can be used to train the model. The timing of model training depends on the scenario: in a streaming setting, the model is updated immediately after each new labelled instance is obtained, while in a batch setting, the model is retrained periodically after incorporating newly labelled instances into the labelled dataset. We depict the elements in Figure 2.

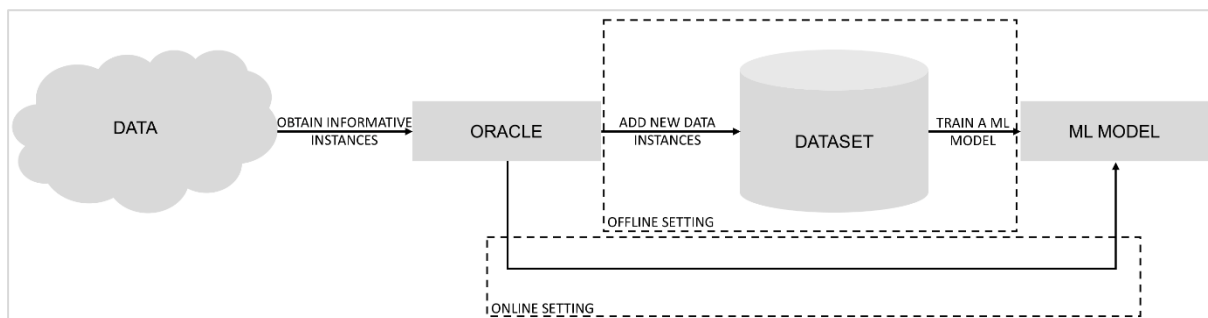


Figure 2: Schematic diagram of active learning.

The oracle provides the data directly to the ML model in an online setting. In contrast, in an offline setting, such instances are persisted into a dataset so that the model can leverage them when retrained. The diagram holds for most cases. An exception could be an oracle being directly asked for a particular kind of demonstration, and therefore data selection would not be needed. The image was taken from [20]

Three aspects must be considered when looking for the most valuable samples [1]: informativeness (contains rich information that would benefit the objective function), representativeness (how many other samples are similar to it), and diversity (the samples do not concentrate in a particular region, but rather are scattered across the whole space).

A simple baseline for selecting data instances is the Random Active Learning (RAL) [21], which proceeds in rounds and, in each round, randomly samples unlabelled data instances, assuming the data follows a uniform distribution. A popular active learning strategy is uncertainty sampling [21], which selects instances that are expected to have the highest uncertainty in their label predictions by the current model. This can be done using various measures of uncertainty, such as the least confidence criterion mentioned above, or the margin sampling criterion, which selects the sample whose two most probable labels are closest in probability. Another approach is to use diversity sampling, which selects instances that are dissimilar to those already in the labelled dataset. This can be done using clustering or other similarity measures.

Other active learning strategies focus on selecting instances that are informative for specific sub-tasks or concepts of interest, such as multi-label active learning or concept-based active learning. In multi-label active learning, the goal is to select instances that are informative for multiple labels, which can be useful in applications such as image tagging or document classification. In concept-based active learning, the goal is to select instances that are

informative for specific concepts or features of interest, which can be useful in applications such as medical diagnosis or fraud detection.

The abovementioned methods described different strategies to select existing data suitable for streaming and batch processing. The membership query synthesis active learning strategy considers no data is selected, but rather a data instance is created and presented to the oracle. In this line, [2] introduced the Generative Adversarial Active Learning (GAAL) technique, which leverages Generative Adversarial Networks (GANs) to generate informative instances based on a random sample of unlabelled instances close to the decision boundary. This concept was evolved by many authors, who aimed to develop variations of GAN architectures e.g., to create a specific instance leveraging additional data regarding the desired target label and therefore leading to faster convergence [3].

Overall, the choice of active learning strategy depends on the specific application and the characteristics of the data. It is often useful to compare the performance of different strategies on a small, labelled dataset before selecting the best one for active learning.

3 State of the Art in Quality Inspection

This deliverable provides a comprehensive overview of the state-of-the-art for quality inspection, as active learning approaches in WP4 are primarily focused on quality inspection tasks within STAR pilots. Quality control is crucial in ensuring that products meet specific requirements and specifications [4]. This is important for building customer trust, boosting loyalty, and reinforcing brand reputation. However, manual inspection can be challenging due to limited scalability and operator-to-operator inconsistency (see [5]). To address these issues, automated visual inspection systems that use artificial intelligence (AI) can be deployed to automate the visual inspection process. These models can reduce manual work, match the speed of production, and trace defect root causes to proactively solve issues in the production process [6,7]. Furthermore, they enable non-contact inspection that is not affected by the target type, surface, or ambient conditions like temperature [8]; and can perform multiple tasks simultaneously, including object, texture, or shape classification, and defect segmentation, among other inspections. Automated visual inspection approaches can be classified into three categories [9]: (a) classification, (b) background reconstruction and removal, and (c) template reference (comparing a template image with a test image). While for classification, much research was devoted to supervised approaches, unsupervised defect detection was explored by many authors. They explored using Fourier transforms to remove regularities and highlight irregularities (defects) [10] or employed autoencoders to find how a reference image differs from the expected pattern [11].

Among applications of visual inspection described in the scientific literature, we find the inspection of TFT-LCD panels and LCD colour filters [12], the detection of surface defects on cold-rolled strips [13], defect detection in weld images [14]. Although active learning has been successfully applied in manufacturing, the scientific literature on this domain remains scarce [15]. Some notable use cases of active learning in manufacturing include the automatic optical inspection of printed circuit boards [16], media news recommendation in a demand forecasting setting [17], and the identification of local displacement between two layers on a chip in the semiconductor industry.

4 Application to STAR Use Cases

4.1 PCL Use Case

The techniques developed in STAR WP4 have been effectively tested using STAR use cases, specifically the Philips Consumer Lifestyle BV use case. The company produces various products that feature their logo. One important task is to visually inspect the printed logo and identify any defects to ensure that only products with correctly printed logos are delivered. The company uses different setups for pad-printing when printing their logo on the products, and the printed products are later inspected. If a defective print is found, the product is removed from the production line.

Philips Consumer Lifestyle BV provided a dataset of 3,518 images, which were categorized into one of three possible classes: good print (no defects observed), double printing, or interrupted printing. The dataset is highly imbalanced, reflecting the company's commitment to high-quality standards.

4.1.1 Implementation

4.1.1.1 Supervised Classifiers

For the purpose of active learning research, we developed supervised classification models, and experimented the use of different kind of oracles (machine and human oracles) to label the data and eventually improve the models' outcomes.

4.1.2 Experiments

4.1.2.1 Experimenting with Active Learning Strategies

The use of different active learning strategies and oracle settings can have a significant impact on the performance and efficiency of the active learning process. In our research, two active learning settings were explored: pool-based and stream-based. Pool-based active learning involves selecting a subset of data instances from the entire pool of unlabelled data, whereas stream-based active learning selects data instances as they arrive in a sequential order. In our experiments, the train set was comprised of 1760 images, the unlabelled data was simulated considering 1056 images, and predictions tested on 352 images, using a ten-fold stratified cross-validation.

Two strategies were used for pool-based active learning: random sampling and selection of instances with the highest uncertainty. The uncertainty of the classification model was measured by considering the highest score for a given class and selecting the instance with the lowest score among the scores provided for the data instances in the active learning set. In stream-based active learning, a decision was made to keep or discard an instance with a probability threshold of 0.5 when random sampling was used. When selecting instances with the highest uncertainty, the prediction for each data instance was analyzed, and the oracle was requested to label the instance only if it fell below a certain confidence threshold ($p=0.95$ or $p=0.99$).

Three oracle settings were considered (see Figure 3), namely (a) a human labeller as the only source of truth, (b) a machine oracle for instances where the classifier had high certainty, and

a human labeller otherwise, and (c) machine oracle for instances where the classifier had high certainty, and another machine oracle was requested when uncertain about the outcome. The choice of oracle setting can affect the accuracy and efficiency of the active learning process, and the most appropriate setting may depend on various factors, such as the availability and cost of human labellers, the quality of the classification model, and the desired level of accuracy.

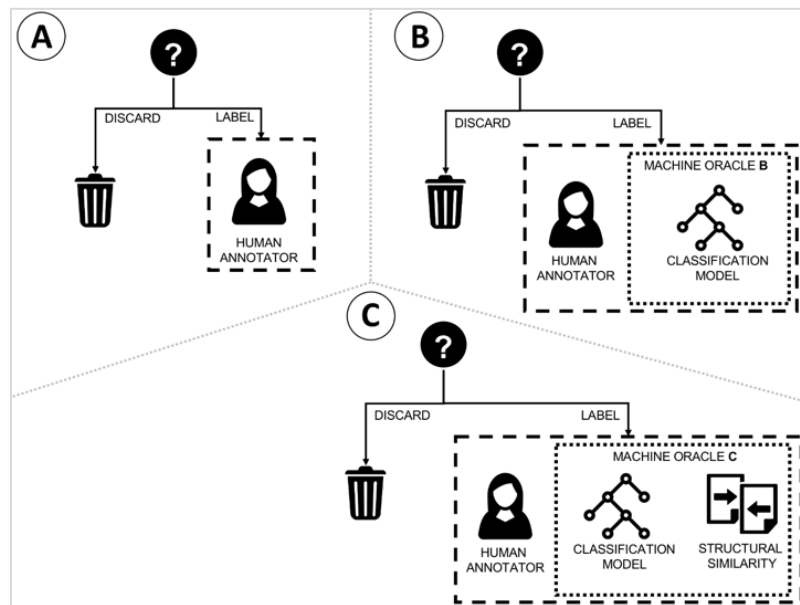


Figure 3 Three oracle settings. The image was taken from [19]

The second oracle was set up to query the closest labelled image from three randomly picked images (one per class). In (C), the machine oracle issued a label only when both machine oracles were unanimous on the label; otherwise, the instance labelling was delegated to a human labeller. The decision on which oracle to query relied on the model's confidence in the outcome and a probability threshold set based on manufacturing quality policies. It was assumed that the second machine oracle in (C) is accessible at a certain cost (e.g., paid external service) and, therefore, cannot be used for every prediction. We simulated such a service by computing the Structural Similarity Index Measure (SSIM) score over the queried image.

We considered eight scenarios (see Table 1) and experimented with five machine learning models. The machine learning models were calibrated using a sigmoid model based on Platt logistic model [18] (see Figure 4).

Table 1 Proposed experiments to evaluate the best active learning setting regarding how it influences the models' learning and its impact on the manual revision workload. The table was taken from [19]

Experiment	AL setting	AL data selection	Oracle
1	pool-based	Random sampling	Human labeler
2	pool-based	Highest uncertainty	Human labeler
3	pool-based	Highest uncertainty	Machine Oracle B + Human labeler

4	pool-based	Highest uncertainty	Machine Oracle B + Human labeler
5	stream-based	Random sampling	Human labeler
6	stream-based	Highest uncertainty	Human labeler
7	stream-based	Highest uncertainty	Machine Oracle B + Human labeler
8	stream-based	Highest uncertainty	Machine Oracle B + Human labeler

$$P(y_i = 1 | f_i) = \frac{1}{1 + \exp(Af_i + B)}$$

Figure 4 Equation 1: Platt classifier calibration logistic model.

4.1.3 Evaluations

4.1.3.1 Evaluation of Active Learning Strategies

The active learning strategies were analysed from two points of view. First, whether they contributed to better learning of the machine learning model. Second, how much manual work could be saved by adopting such strategies.

4.1.3.1.1 Impact of Active Learning strategies on ML model learning

To understand how much the active learning strategies contributed to a better learning of the model, we contrasted the models' performance variability observed when they consumed the data within the Q1 and Q4 quartiles of the active learning pool. The results showed that best outcomes were observed for Experiment 2 (highest uncertainty with human labeller) settings, while the second-best performance was observed for Experiment 8 (highest uncertainty, with the machine and human oracles). Interestingly, the streaming setting showed better performance than the pool-based experiments, even though the streaming experiments achieved only the second-best results with Platt scaling. We noticed that while soft labelling was detrimental for the pool-based active learning settings, it led to superior results in a streaming setting, achieving results close to the best ones obtained across all experiments.

Table 2 The table displays the mean AUC ROC values for five machine learning models, averaged across ten folds. The results highlight the impact of various active learning strategies on the models' performance over time, comparing the first quartile (Q1) to the last quartile (Q4) of the active learning pool. Soft labelling was applied using two probability thresholds (0.95 and 0.99). The best results are bolded, and the second-best results are displayed in italics. The table was taken from [19]

Setup	Experiment	p=0.95		p=0.99	
		Q1	Q4	Q1	Q4
Pool-based	1 - random, human oracle	0.8428	0.8612	0.8431	0.8623

	2 - uncertainty, human oracle	0.8594	0.8693	0.8594	0.8693
	3 - uncertainty, oracle (machine B + human)	0.8398	0.8396	0.8398	0.8398
	4 - uncertainty, oracle (machine C + human)	0.8349	0.8348	0.8358	0.8358
Streaming	5 - random, human oracle	0.8460	0.8559	0.8460	0.8559
	6 - uncertainty, human oracle	0.8525	0.8647	0.8529	0.8647
	7 - uncertainty, oracle (machine B + human)	0.8505	0.8608	0.8529	0.8647
	8 - uncertainty, oracle (machine C + human)	<i>0.8550</i>	<i>0.8665</i>	<i>0.8553</i>	<i>0.8668</i>

Table 3 The mean AUC ROC values were computed across ten test folds for five machine learning models to determine how they learn over time (Q1 vs. Q4) under the Experiment 2 setting. We also examined whether the differences were statistically significant at a p-value of 0.95 (DS(p=0.95)). The results are presented below, with the best outcomes highlighted in bold and the second-best results displayed in italics. The table was taken from [19]

Model	Q1	Q4	DS(p=0,95)
MLP	0.9309±0.0004	0.9448±0.0003	Yes
SVM	<i>0.8788±0.0007</i>	<i>0.8767±0.0007</i>	Yes
NB	0.8628±0.0005	0.8675±0.0005	Yes
KNN	0.8575±0.0006	0.8720±0.0006	Yes
CART	0.7669±0.0007	0.7854±0.0008	Yes

In Table 3 we report the results obtained for Experiment 2. We evaluated the performance of machine learning models and compared their results after being shown Q1 and Q4 of the active learning pool data. Our findings showed that MLP had the best performance, followed by SVM, with a difference of at least 0.05 AUC ROC points. While MLP's performance increased over time, SVM slightly decreased in Q4, and no other model had a performance decrease. Since Experiment 2 only involved a human oracle, and the annotations were accurate, we

ruled out mislabelling as the cause of the performance decrease. We also consider class imbalance as an improbable factor since other models could better discern among the classes over time. Finally, CART model had the worst performance, lagging by more than 0.16 AUC ROC points compared to the best-performing model.

4.1.3.1.2 Impact of Active Learning strategies on data annotation efforts

The preceding section highlighted the importance of assessing the potential of active learning strategies to enhance the machine learning models' performance. In this section, we present insights obtained on how active learning strategies can impact the manual annotation burden. In particular, we were interested in how machine oracles could reduce the required amount of effort for human labelling.

Table 4 presents the results for a cut-off value of $p=0.95$. Through the research conducted, we found that when using a cut-off value of 0.99, no instances were given to the machine oracles for analysis, so no analysis was performed. However, when using a cut-off value of 0.95, the number of soft-labelled instances was negligible for each experiment. This suggests that using machine oracles would not significantly reduce the manual labelling effort, even though the quality of the annotations was high. The experiments with streaming settings (Experiment 7 and Experiment 8) had the highest number of soft-labelled instances, but 96% of samples were correctly labelled in both cases, meeting the quality threshold of $p=0.95$. The best machine labelling quality was achieved when using Oracle C (unanimous vote of two machine oracles). Contrasting these results with the AUC ROC values obtained for each experiment, it was observed that while Experiments 3 and 4 showed a slight decrease in discriminative power, Experiments 7 and 8 showed an increase in performance of at least 0.01 AUC ROC points.

Table 4 The table presents the proportion and quality of soft labelling under different settings, with a predicted probability cut-off value of $p=0.95$. SL (%) shows the percentage of soft annotated data instances relative to the total, SL OK (%) indicates the percentage of correctly soft annotated instances, ML SL OK (%) represents the percentage of soft annotated data instances that would be correctly annotated based on the ML model score, and SSIM SL OK (%) indicates the percentage of soft annotated data instances that would be correctly annotated based on the SSIM score. The table was taken from [19]

Experiment	$p=0.95$			
	SL (%)	SL OK (%)	ML SL OK (%)	SSIM SL OK (%)
3	0.0077	0.9684	0.0075	NA
4	0.0033	0.9756	0.0050	0.0033
7	0.0413	0.9685	0.0400	NA
8	0.0343	0.9692	0.0483	0.0334

4.1.4 Demos and proof of concept

For the purpose of the research conducted regarding Active Learning, we developed multiple models and experiments at a prototype level, with the data provided by Philips Consumer

Lifestyle BV. Many of those experiments were developed in Jupyter Notebooks, as shown in Figure 5.

```
def train_for_learner_alpool_no_soft_labeling(classifier_name, calibration_strategy, repetitions, threshold):
    fold_file_path = "./data/philips/embeddings-original-rep-{}-fold-{}.csv"

    folds = sorted([x for x in range(0, 10)])

    with tqdm(total=3519*repetitions) as pbar:
        performance_measurements = {}
        for repetition in range(0, repetitions):
            performance_measurements[repetition]={}
            for fold in range(0, 10):
                tdigest = TDigest()
                print("Pool fold: {}".format(fold))
                poolfold = folds[fold]
                valfold = folds[fold+1-10]
                testfold = folds[fold+2-10]

                trainfolds = set(folds).difference(set([poolfold, valfold, testfold]))

                train_frames = []
                for trainfold in trainfolds:
                    train_frames.append(pd.read_csv(fold_file_path.format(repetition, trainfold)))
                df_train = pd.concat(train_frames)
                df_pool = pd.read_csv(fold_file_path.format(repetition, poolfold))
                df_val = pd.read_csv(fold_file_path.format(repetition, valfold))
                df_test = pd.read_csv(fold_file_path.format(repetition, testfold))

                selected_features = select_features(df_train[features], df_train['target'], math.floor(math.sqrt(len(df

                X_pool = df_pool[selected_features].values
                y_pool = df_pool['target'].values

                if calibration_strategy == "sigmoid":
                    learner = train_and_calibrate(get_classifier(classifier_name), df_train[selected_features].values,
```

Figure 5 Snippet from a Jupyter Notebook, where active learning experiments were implemented.

Nevertheless, we also created a REST API (see Figure 6) and implemented six active learning policies: three pool-based active learning policies (random sampling, classifier entropy, and classifier margin), and three stream-based active learning policies (random sampling, classifier entropy, and classifier margin). The abovementioned policies can be used by any service to perform active learning. Furthermore, we implemented a variety of endpoints supporting pool-based active learning. These endpoints enable to (a) register feature vectors, (b) register their labels when provided by the oracles, (c) retrieve feature vectors that are labelled (for training supervised models) or unlabelled (to perform a custom selection among them), (d) retrieve a particular feature vector for inspection, and (e) register model predictions for a given feature vector. We detail the endpoint requirements in Table 5.

Active Learning Module		^
POST	/active-learning-module/feature-vectors	Add Feature Vector Data
GET	/active-learning-module/feature-vectors-labeled	Get Feature Vectors
GET	/active-learning-module/feature-vectors-unlabeled	Get Unlabeled Feature Vectors
GET	/active-learning-module/feature-vectors/{id}	Get Feature Vector Data
PUT	/active-learning-module/feature-vectors/{id}/{label}	Update Item
POST	/active-learning-module/model-predictions	Add Model Prediction Data
GET	/active-learning-module/pool-based/random-sampling/{model_id}	Pool Random Sampling
GET	/active-learning-module/pool-based/classifier_entropy/{model_id}	Pool Classifier Entropy
GET	/active-learning-module/pool-based/classifier_margin/{model_id}	Pool Classifier Margin
GET	/active-learning-module/stream-based/random-sampling	Streaming Random Sampling
POST	/active-learning-module/stream-based/classifier_entropy	Streaming Classifier Entropy
POST	/active-learning-module/stream-based/classifier_margin	Streaming Classifier Margin

Figure 6 Active Learning REST API demo.

Table 5 The table presents a collection of endpoints we implemented for the active learning use case.

HTTP VERB	ENDPOINT	REQUEST BODY / QUERY PARAMS
POST	/active-learning-module/feature-vectors	{"feature_vector": [0.8, 0.05, 0.05, 0.1], "label": -1}
GET	/active-learning-module/feature-vectors-labeled	
GET	/active-learning-module/feature-vectors-unlabeled	
GET	/active-learning-module/feature-vectors/{id}	
PUT	/active-learning-module/feature-vectors/{id}/{label}	
POST	/active-learning-module/model-predictions	{"feature_vector_id": 2, "model_id": 1, "predictions": [0.23, 0.77]}
GET	/active-learning-module/pool-based/random-sampling/{model_id}	query parameter: items_count
GET	/active-learning-module/pool-based/classifier_entropy/{model_id}	query parameter: items_count
GET	/active-learning-module/pool-based/classifier_margin/{model_id}	query parameter: items_count
GET	/active-learning-module/stream-based/random-sampling	
GET	/active-learning-module/stream-based/classifier_entropy	{"prediction_vector": [0.23, 0.77]}
GET	/active-learning-module/stream-based/classifier_margin	{"prediction_vector": [0.23, 0.77]}

5 NLP Functionalities

We began the NLP section of the previous deliverable, detailing how NLP comprises a wide spectrum of technologies and research topics, and detailing how we had started research in Speech-To-Text (STT) and Text-to-Speech (TTS) technologies, which then evolved into conversational agents. In the initial document NLP was encompassed in an architecture for active learning systems such as the one detailed in this document and its previous version, detailing the sub-architecture for browser-based NLP module.

In this deliverable, we will detail that evolution, in which we have progressed from the initial interaction module to a module based on chatbots and conversational agents.

5.1 Early proofs of concept

Early in the project a proof-of-concept (PoC) was presented. In that PoC different voices could be evaluated in different browsers and the recognition of various demand forecast related sentences could be tested. The implementation helped us to research on grammar and grammar-less recognitions and paved the way for the grammar submodule of the Chatbot approach, which we are presenting in this section.

Web Speech API Demo 2

Speech Synthesizer:

Input text

Voice Microsoft Aria Online (Natural) - English (United States) (en-US) ▼ Synthesize

Recognition language English ▼

Utterances

- 1) Change demand [NUM] to [VALUE]
- 2) Explain demand [NUM]
- 3) Mark feature [NUM]
- 4) Start plan for demand [NUM]
- 5) Select option [NUM]
- 6) Show other options
- 7) Select feedback [NUM]
- 8) Utterance: Feedback [TEXT]

with [NUM]: [1, 2, 3], [VALUE]: [1000, 2000, 3000] and [TEXT]: *

Voice interaction

Activation words: "Ehi computer" | "Computer"

Start listening

Recognition results:

The user said:

Most likely utterance:

Figure 7 Web Speech API Demo 2

The image above shows the Web Speech and Interaction demo, developed as part of the collaboration between R2M, JSI and Qlector. The top speech Synthesizer part allows the test of different voices, while the bottom part was a multilanguage recognition system based on grammars related to demand forecasting.

5.2 From STT and TSS to Chatbots

In the conclusions of the initial version of this deliverable (D4.4), we detailed how we were evaluating different possibilities to demonstrate NLP functionality in the project. In addition, we studied different possibilities on how NLP can be beneficial for the manufacturing sector. Among them, the following cases were considered, as they are mentioned in the book chapter *Multimodal Human Machine Interactions in Industrial Environments* [22] (written in collaboration between R2M and University of Cagliari).

- **Process Automation:** The use of NLP technologies in the manufacturing process allows the automatic processing of information in natural language and the execution of repetitive tasks like paperwork and report analysis.
- **Inventory Management:** Analysing data about the stock, sales and user reports of certain products is essential to assess the correct decisions for a company to optimise and maximise profits. By leveraging NLP technologies, the resulting benefits are: 1) the entire process becomes more comprehensive; 2) it is more difficult to incur errors related to the analysis of sales; 3) it is easier to analyse the manufactured products and discard those with low quality without affecting the supply chain and sales.
- **Emotional Mapping:** Sentiment analysis and emotion detection are some of the most exciting features of NLP. Early NLP systems allowed organisations to collect speech-to-text communication without accurately determining its full meaning. NLP approaches can sort and understand the nuances and emotions in human voices and text, giving organisations unparalleled insight. Learning customer expectations and operators' viewpoints is a very important element in manufacturing. NLP technologies permit to identify emotions and the polarity of the opinions of customers and operators and provide actions to improve products and different processes.
- **Operation Optimisation:** NLP technologies can be employed to trace the performance of equipment and improve the interaction with machines. This simplifies the operation of complex systems and can enable Human Machine Interaction where the operator and the machine collaborate in order to optimise processes.
- **Worker Upskilling:** Use NLP techniques to enable employees to improve their knowledge of their occupation, to detect possible training needs and to receive recommendations.

From the potential use cases in the project, the last one (Worker Upskilling) was a real case of interest for integrating the NLP module, for several reasons: 1) Perfect fit and immediate benefits of using NLP; 2) Because the case was general and allowed for post-project walkthrough; 3) Because the operating environment did not have the limitations that can be found in other use cases (e.g. noisy environments, too simple grammars...).

For this reason, we focused the integration and development of the proofs of concept in the period associated with this deliverable on NLP modules related to chatbots.

5.3 Conversational Agents

Natural Language Processing (NLP) is crucial to perform recommendations of items that can be only described by natural language. However, NLP usage within recommendation modules is difficult and usually requires a relevant initial effort, thus limiting its widespread adoption. To overcome this limitation, we worked on a novel architecture that can be instantiated with NLP and Machine Learning (ML) modules to perform recommendations of items that are described by natural language features. Furthermore, providing users with an interactive agent on a clear user interface in which they can converse, and the fact that the system itself knows how to reason about the information that the user is sharing in order to make efficient recommendations is also a key element, especially if we want to use the tool in manufacturing environments where users do not necessarily have to have a highly technical background.

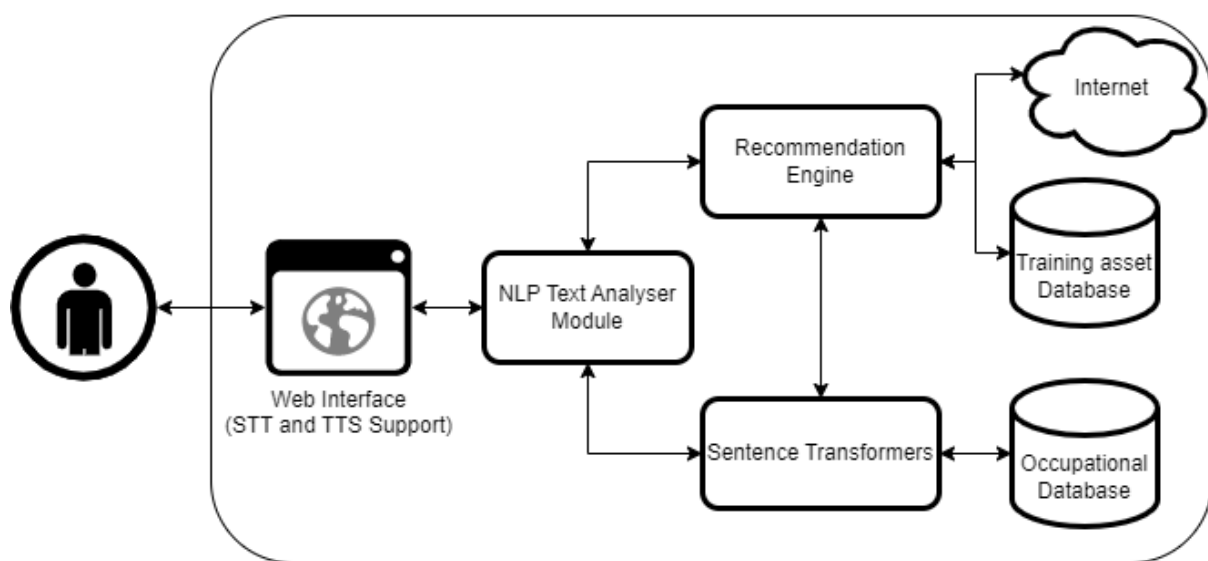


Figure 8 Conversational Agents Module

To implement this, we have developed a module that includes three main functionalities, being the NLP Text analyser and the UI, the key elements developed within WP4:

- NLP Text Analyser: In charge of the text processing and the general logic of the conversation and coordination with the rest of the components.
- Sentence Transformers: Allows us to generate text embeddings and find similarities between texts, so we can associate a conversation to a specific occupation, or know related areas for recommendations.
- Recommendation Engine: Engine for recommendations on training materials, which makes use of both public databases and information generated in the project to recommend courses and other training assets.

5.4 Demos and Proof of Concepts

The image below shows the concept of the Virtual Star Interviewer, a conversational agent that can make an analysis of the conversation to compare it with the necessary requirements for an occupation and shows recommendations for potential improvements and training materials.

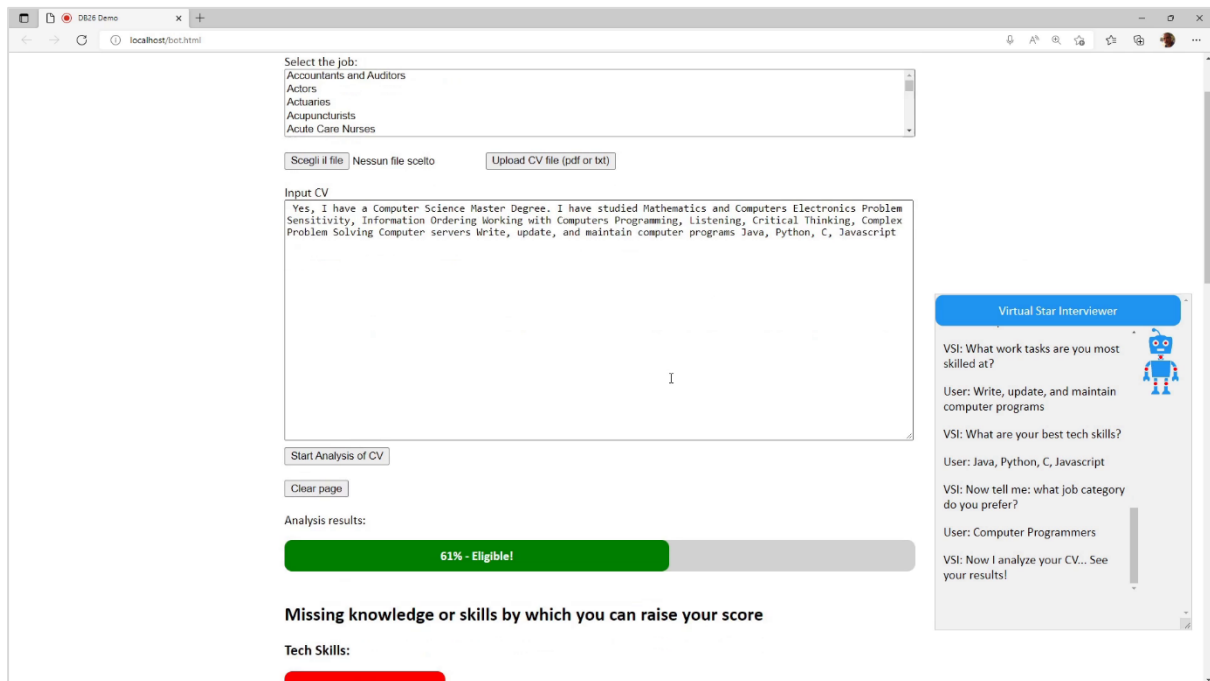


Figure 9 NLP Demo: Star Interviewer

As shown in the image, the chatbot is located at the bottom right of the UI, it can be interacted with in a multimodal way (voice and text) and asks a series of questions with which, after processing the information received from the user, it can analyse this information with respect to an occupation.

The following image also shows a detail of a chatbot made for another WP5 functionality. This particular chatbot is a proof of concept for another conversational interface on occupations that allows for a broader dialogue, and even answers questions in Open Domain by making use of web search APIs.

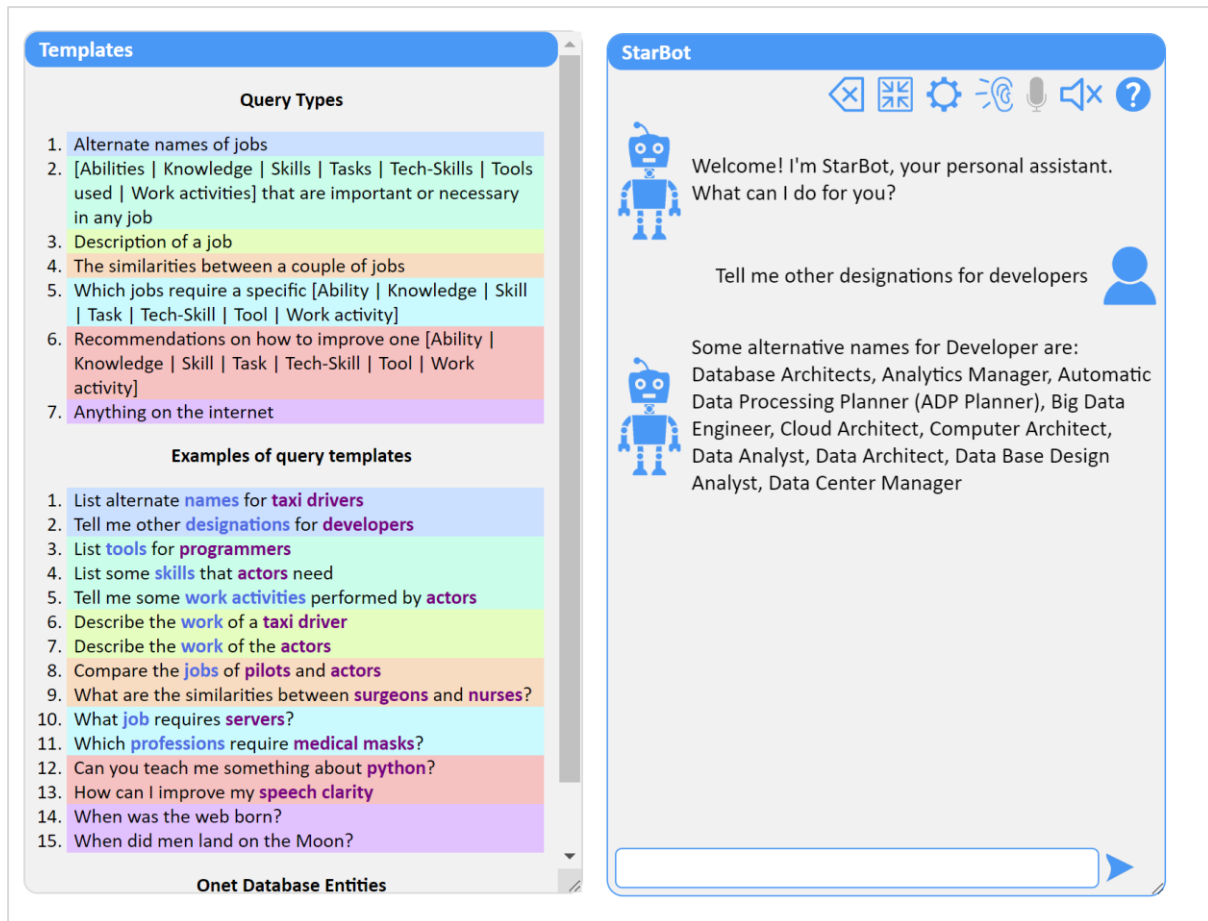


Figure 10 NLP Demo: Conversational Interface on Occupations

5.5 Integration and future NLP activities

On the one hand, these conversational interfaces are being used to access the content elaborated in the WP5 work package, especially in everything related to Worker Training Platforms and Worker Continuous Learning.

On the other hand, we are seeing how chatbots based on Large Language Models, such as ChatGPT are a good option for general conversation, although for specific conversation they are still not quite accurate and give dissimilar answers (e.g. the same questions about what the main tasks of a robot operator are, give different results in different moments).

The work related to NLP in WP4 has generated valid demonstrators and materials for its integration. This research is being carried out with WP5 and it is intended that the chatbot functionality will be included in a section of the public page of the project (WP7/WP8) so that anyone interested in AI in manufacturing can test the functionality.

6 Conclusions

In this deliverable we present the recent developments for STAR WP4 task 4.3 Human-Robot Interactions for Active Learning. The main development of task 4.3 has been focused on the PCL use case, on quality inspection of Phillips manufactured products (Philips Consumer Lifestyle BV). Quality control allows companies to verify the products' conformance to requirements and specifications and thus build customer satisfaction and the brand's reputation.

The work carried out in Task 4.3 proves that active learning can be used to improve the classification models continuously by leveraging newly labelled data in the manufacturing industry.

As planned in Task 4.3, deliverable 4.5 presents the advances with respect to Natural Language Processing demonstrators, describing the sum-ups the carried NLP activities and providing the system prototype for PCL use case.

The research and development outcomes of Task 4.3 resulted in the preparation of the following publications:

- Active Learning and Novel Model Calibration Measurements for Automated Visual Inspection in Manufacturing:
 - o Rožanec, J. M., Bizjak, L., Trajkova, E., Zajec, P., Keizer, J., Fortuna, B., Mladenić, D. Active Learning and Novel Model Calibration Measurements for Automated Visual Inspection in Manufacturing, arXiv:2209.05486, <https://doi.org/10.48550/arXiv.2209.05486>.
- CHAPTER *Active Learning* in The future of data mining:
 - o Rožanec, J. M., Fortuna, B., Mladenić, D. Active learning. In: BAYTAR, Cem Ufuk (ed.). The future of data mining. New York: Nova Science Publishers, 2022. p. 1-14, ilustr. Research Methodology and Data Analysis. ISBN 979-8-88697-250-4.

The outcomes of Task 4.3 Human-Robot Interactions for Active Learning are intended to influence not only project partners from STAR use cases but become the basis for future development of active learning techniques in manufacturing.

References

- [REF-01] Wu D. Pool-based sequential active learning for regression. *IEEE transactions on neural networks and learning systems*, 30(5):1348–1359, 2018.
- [REF-02] Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M. and He, X. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2019.
- [REF-03] Sinha, S., Ebrahimi, S., and Darrell, T. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.
- [REF-04] Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., and Tang, S. (2020). Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges. *Materials*, 13(24), 5755.
- [REF-05] See, J.E. (2012). *Visual inspection: a review of the literature*. Sandia Report SAND2012-8590, Sandia National Laboratories, Albuquerque, New Mexico.
- [REF-06] Chin, R.T. and Harlow, C.A. (1982). Automated visual inspection: A survey. *IEEE transactions on pattern analysis and machine intelligence*, (6), 557–573.
- [REF-07] Chouchene, A., Carvalho, A., Lima, T.M., Charrua-Santos, F., Osorio, G.J., and Barhoumi, W. (2020). Artificial intelligence for product quality inspection toward smart industries: quality control of vehicle non-conformities. In *2020 9th international conference on industrial technology and management (ICITM)*, 127–131. IEEE.
- [REF-08] Park, J.-K., Kwon, B.-K., Park, J.-H., Kang, D.-J. (2016). Machine learning-based imaging system for surface defect inspection. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 3 (3), 303–310.
- [REF-09] Jian, C., Gao, J., Ao, Y. (2017). Automatic surface defect detection for mobile phone screen glass based on machine vision. *Applied Soft Computing*, 52, 348–358.
- [REF-10] Aiger, D., & Talbot, H. (2012). The phase only transforms for unsupervised surface defect detection. *Emerging topics in computer vision and its applications* (pp. 215–232). World Scientific. [https://doi.org/10.1109.CVPR.2010.5540198](https://doi.org/10.1109/CVPR.2010.5540198)
- [REF-11] Mujeeb, A., Dai, W., Erdt, M., Sourin, A. (2018). Unsupervised surface defect detection using deep autoencoders and data augmentation. *2018 international conference on cyberworlds (cw)* (pp. 391–398). <https://doi.org/10.1109/CW.2018.0007>
- [REF-12] Tsai, D.-M., & Lai, S.-C. (2008). Defect detection in periodically patterned surfaces using independent component analysis. *Pattern Recognition*, 41 (9), 2812–2832.
- [REF-13] Kang, G.-W., & Liu, H.-B. (2005). Surface defects inspection of cold rolled strips based on neural network. *2005 international conference on machine learning and cybernetics* (Vol. 8, pp. 5034–5037). <https://doi.org/10.1109/ICMLC.2005.152783>
- [REF-14] Valavanis, I., & Kosmopoulos, D. (2010). Multiclass defect detection and classification in weld radiographic images using geometric and texture features. *Expert Systems with Applications*, 37 (12), 7606–7614.
- [REF-15] Meng, L., McWilliams, B., Jarosinski, W., Park, H.-Y., Jung, Y.-G., Lee, J., Zhang, J. (2020). Machine learning in additive manufacturing: A review. *Jom*, 72 (6), 2363–2377.
- [REF-16] Dai, W., Mujeeb, A., Erdt, M., Sourin, A. (2018). Towards automatic optical inspection of soldering defects. *2018 international conference on cyberworlds (cw)* (pp. 375–382). <https://doi.org/10.1109/CW.2018.0007>
- [REF-17] Zajec, P., Rožanec, J.M., Novalija, I., Fortuna, B., Mladenčić, D., Kenda, K. (2021). Towards active learning based smart assistant for manufacturing. *Ifip international*

conference on advances in production management systems (pp. 295–302).

https://doi.org/10.1007/978-3-030-85910-7_3

[REF-18] Platt, J., et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10 (3), 61–74.

[REF-19] Rožanec, J. M., Bizjak, L., Trajkova, E., Zajec, P., Keizer, J., Fortuna, B., Mladenić, D. Active Learning and Novel Model Calibration Measurements for Automated Visual Inspection in Manufacturing, arXiv:2209.05486, <https://doi.org/10.48550/arXiv.2209.05486>.

[REF-20] Rožanec, J. M., Fortuna, B., Mladenić, D. Active learning. In: BAYTAR, Cem Ufuk (ed.). *The future of data mining*. New York: Nova Science Publishers, 2022. p. 1-14, ilustr. *Research Methodology and Data Analysis*. ISBN 979-8-88697-250-4.

[REF-21] Settles, Burr. "Active learning literature survey." (2009)

[REF-22] Alonso, R., Cauli, N., Reforgiato, D. (2021). Multimodal Human Machine Interactions in Industrial Environments. In *Trusted Artificial Intelligence in Manufacturing: A Review of the Emerging Wave of Ethical and Human Centric AI Technologies for Smart Production* (<http://dx.doi.org/10.1561/9781680838770>).

[REF-23] Rožanec, J. M., Zajec, P., Trajkova, E., Šircelj, B., Breclj, B., Novalija, I., Dam, P., Fortuna, B., Mladenić, D. Towards a comprehensive visual quality inspection for industry 4.0.*. In: BERNARD, Alain (ed.). *10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022 Nantes, France, 22-24 June 2022*. New York: International Federation of Automatic Control, 2022. Str. 690-695, ilustr. *IFAC papersOnline*, vol. 55, iss.10. ISSN 2405-8963.

[REF-24] <https://www.sciencedirect.com/science/article/pii/S2405896322017827>, DOI: 10.1016/j.ifacol.2022.09.486.