UNIVERSITAT DE BARCELONA

Grau de Lingüística

Treball de Fi de Grau

Curs 2022-2023

TÍTOL: Enhancing the DVIL Chatbot through Linguistic Expertise

NOM DE L'ESTUDIANT: Francesca Wright

NOM DEL TUTOR: Mariona Taulé

Barcelona, 16 junio 2023

UNIVERSITAT DE BARCELONA

Coordinació d'Estudis
Facultat de Filologia

Gran Via
de les Corts Catalanes, 585
08007 Barcelona

Tel. +34 934 035 594
fil-coord@ub.edu
www.ub.edu

Amb aquest escrit declaro que sóc l'autor/autora original d'aquest treball i que no he emprat per a la seva
elaboració cap altra font, incloses fonts d'Internet i altres mitjans electrònics, a part de les indicades. En el
treball he assenyalat com a tals totes les citacions, literals o de contingut, que procedeixen d'altres obres.
Tinc coneixement que d'altra manera, i segons el que s'indica a l'article 18, del capítol 5 de les Normes
reguladores de l'avaluació i de la qualificació dels aprenentatges de la UB, l'avaluació comporta la
qualificació de "Suspens".

Barcelona, a 16 junio 2023

Signatura:

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my tutor, Mariona Taulé, for her guidance and invaluable assistance throughout the entire process of this project. Her expertise, insightful feedback, and encouragement have greatly influenced the outcome of this work.

I am immensely grateful to Ecem Kavaz for her indispensable assistance in handling the technical aspects of the chatbot development and carrying out the exploratory study. Without her, this work would not have come this far.

I extend my sincere appreciation to Anna Puig and Inmaculada Rodríguez for their contributions to the design of the exploratory study. Their input, suggestions, and expertise have greatly enriched the research process of the evaluation of the chatbot.

I am also thankful to all the participants of the exploratory study who generously dedicated their time and shared their valuable opinions on the current state of the chatbot.

Finally, I could not have undertaken this journey without my family and friends. Their consistent support, encouragement and understanding throughout the development of this work has been a constant source of motivation and strength, helping me overcome challenges and reach this significant milestone.

ABSTRACT

This dissertation focuses on the development of DVIL chatbot integrated in the Data Visualisation in Linguistics (DVIL) platform, a resource that offers visualisation and analytical tools to enhance the analysis of toxicity in the NewsCom-TOX corpus. The chatbot facilitates natural language interaction with the interface by incorporating natural-language understanding techniques and machine learning algorithms to enable dialogue between the users and interface using queries, to carry out a comprehensive analysis of the corpus. The involvement of linguist has been fundamental in refining the chatbot's capabilities through the enhancement of natural language understanding (NLU) and improvement of responses provided by the chatbot. This work presents the current state of the DVIL chatbot and highlights its potential for facilitating analysis within the DVIL interface, while acknowledging ongoing efforts for future improvements.

Keywords: chatbot, visualisation interface, natural-language understanding

RESUMEN

Este trabajo se centra en el desarrollo del chatbot DVIL integrado en la plataforma de Data Visualisation in Linguistics (DVIL), un recurso que ofrece herramientas de visualización y análisis para mejorar el análisis de toxicidad en el corpus NewsCom-TOX. El chatbot facilita la interacción en lenguaje natural con la interfaz, al incorporar técnicas de comprensión del lenguaje natural y algoritmos de aprendizaje automático para permitir el diálogo entre los usuarios y la interfaz mediante consultas, para llevar a cabo un análisis exhaustivo del corpus. La participación del lingüista ha sido fundamental para mejorar las capacidades del chatbot mediante la mejora de la comprensión del lenguaje natural y la mejora de las respuestas proporcionadas por el chatbot. Este trabajo presenta el estado actual del chatbot DVIL y resalta su potencial para facilitar el análisis dentro de la interfaz DVIL, al tiempo que identifica los esfuerzos en curso para futuras mejoras.

Palabras clave: chatbot, interfaz de visualización, comprensión del lenguaje natural

RESUM

Aquest treball se centra en el desenvolupament del xatbot DVIL integrat en la plataforma de Data Visualisation in Linguistics (DVIL), un recurs que ofereix eines de visualització i anàlisi per millorar l'anàlisi de la toxicitat en el corpus NewsCom-TOX. El xatbot facilita la interacció en llenguatge natural amb la interfície. Incorpora tècniques de comprensió del llenguatge natural i algoritmes d'aprenentatge automàtic per permetre el diàleg entre els usuaris i la interfície mitjançant consultes, i per dur a terme una anàlisi exhaustiva del corpus. La participació del lingüista ha estat fonamental per afinar les capacitats del xatbot a través de la millora de la comprensió del llenguatge natural i de les respostes proporcionades pel xatbot. Aquest treball presenta l'estat actual del xatbot DVIL i destaca el seu potencial per facilitar l'anàlisi dins de la interfície DVIL, tot identificant els esforços en curs per a futures millores.

Paraulas clau: xatbot, interfície de visualització, comprensió del llenguatge natural

TABLE OF CONTENTS

1. INTRODUCTION

This paper serves as a written record and discussion of the development of the DVIL chatbot, outlining the practical nature of this final project. The work carried out to develop the DVIL chatbot comes under the project FairTransNLP-LANGUAGE: Analysing toxicity and stereotypes in language for unbiased, fair and transparent systems (PID2021-124361OB-C33)[1]. This project aims to model toxicity language and stereotypes in order to mitigate the undesirable effects of biased data and build high-quality unbiased and fair datasets[2]. To fulfil these objectives, the NewsCom-TOX corpus was developed (Taulé et al., 2021).

The NewsCom-TOX corpus consists of 4,357 comments posted in response to different articles extracted from Spanish online newspapers (ABC, elDiario.es, El Mundo, NIUS, among others) and discussion forums (such as Menéame) from August 2017 to July 2020. These articles were manually selected taking into account their controversial subject matter, their potential toxicity, and the number of comments posted (minimum 50 comments) (Taulé et al., 2021). All comments from the selected article were extracted for annotation. Given the subjectivity of the annotation of toxicity, the methodology applied aimed to provide annotators with contextual information referring to the comment's level in the thread. Annotators were provided with a spreadsheet with numerous columns. The first included the comment they had to annotate, as well as the other comments in its thread. They began annotating the features argumentation, constructiveness, stance, target, stereotype, sarcasm, mockery, insult, improper language, aggressiveness, and intolerance. Binary values were assigned (1 or 0) indicating that the presence/absence of the feature. Then annotators labelled toxicity, beginning with binary values: 0 or 1 corresponding to not toxic or toxic. Finally, a level of toxicity was determined. Comments annotated as toxic (1) were annotated with a level of toxicity: toxicity_level_1=mildly toxic', 'toxicity_level_2=toxic' or 'toxicity_level_3=very toxic'.

As previously mentioned, the methodology provided annotators with contextual information with the aim of reducing subjectivity. The context in which a comment is interpreted by annotators plays role in determining the features present in the comment as annotators could refer to comments in the same thread. Additionally, the annotation of features played a role in determining the level of toxicity, certain features or combinations of features may be associated with higher toxicity levels.

[1] The FairTranNLP-Language project is funded by MCIN/AEI/10.13039/501 100011033/FEDER,UE
[2] http://clic.ub.edu/en/projects

Given the significance of the context and the annotation of features in the annotation of toxic language, the analysis and interpretation of results is complex. The DVIL platform (Kavaz et al., 2021; Kavaz et al., 2022) provides the annotators of this corpus the opportunity to analyse their data, for instance, the distribution of annotated messages, the relationships between features and the detection of inconsistencies in the annotation through visualisations. This resource provides an interface for annotators to visualise their data with visualisation tools such as layouts, glyphs and filters for features as well as statistics which aid the interpretation of results.

The DVIL chatbot is a component implemented into the DVIL interface, with the aim of facilitating interaction with the interface. Capable of interpreting user queries related to actions carried out the interface, contextual and statistic-based queries as well as help-related queries, the chatbot also carry out the user's request by performing an action and displaying natural language responses. Responses may be expressed through different types of interactions, which will be discussed in section 2.1.2.

In this paper I will introduce the concepts of chatbots (Section 2), the DVIL chatbot (Section 3), the Rasa[3] platform (Section 4) and the necessary concepts addressed to carry out improvements, while addressing the improvements themselves (Section 5). Then, I will introduce an observational study carried out to evaluate the improvements (Section 6) before concluding with a next set of improvements to be made as part of future work (Section 7).

2. CHATBOTS

In this section I will describe what a chatbot is, the applications of chatbots and elaborate on the categorisation of chatbot characteristics.

2.*1. What is a chatbot?*

A chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were

[3] https://rasa.com

communicating with a real person (What Is a Chatbot | Oracle, n.d.). A chatbot uses Natural Language Processing (NLP) mechanisms which allow it to interpret and participate in dialogues with humans by extracting the useful, structured and comprehensible data and specific information from unstructured and complex human language (*NLP Chatbot: What Is Natural Language Processing and How It Works?*, n.d.).

The implementation of chatbots is common in several industries including customer service; sales and marketing; retail and ecommerce; banking finance and fintech; healthcare; media and entertainment and education among many more (*20 Fascinating Chatbot Applications From Six Key Industries*, n.d.). Given the wide variety of industries, possible functions and interactions, chatbots can be categorised by type, taking into account several factors. (*Tipos de Chatbot: Ventajas y Características - Centribal*, n.d.). Using the categorised developed by Eduard Vives Isern (2023), Table 1 provides a generalised scheme based on their categorisation, while introducing additional information.

Intelligence	Interaction type	Specialisation	Distribution platform
Rule-based	Buttons	General chatbot	Open Source
Artificial Intelligence (AI)	Textual	Specialised chatbot	Commercial
Hybrid	Multimedia		
	Speech Interaction		

Table 1. The categorisation of the types of chatbot according to four categories

Due to restrictions on the length of this paper, I will only elaborate on the characteristics which categorise the DVIL chatbot.

2.1.1. Artificial intelligence (AI) chatbots

AI Chatbots use NLP technologies to understand the intent behind the question posed by the user (*Best Types of Chatbots for 2023|Rule-Based Chatbots vs AI Chatbots*, n.d.). NLP technologies in this context can be understood to the incorporation of Machine Learning and

Deep Learning, into Natural Language Understanding (NLU) models. AI Chatbots can be considered the most intelligent chatbot type, as NLP technologies allow for more complex understanding on behalf of the chatbot and a more natural interaction between humans and chatbots as the contexts in which the dialogue takes place are learned and improved on (*Tipos de Chatbot: Ventajas y Características - Centribal*, n.d.). Linguists can play an important role in the development of this type of chatbot by defining the contexts in which a query takes place, allowing for more flexible and more flows of dialogue between the user and the chatbot. AI chatbots can also be referred to as virtual assistants or virtual agents (*What Is a Chatbot? | IBM*, n.d.). Some examples of virtual assistants include Apple's Siri[4] or Amazon's Alexa[5].

2.1.2. Interaction

The DVIL chatbot interacts using different interaction types in different contexts. Upon continuation, I will explain the interaction types used by the DVIL chatbot.

2.1.2.1. Text-based interaction

Text-based communication allows the user and the chatbot communicate via written text (Tipos de Chatbot: Ventajas y Características - Centribal, n.d.). A website chat widget, a small pop-up window that stays on your website (*What's in a Chat Widget? - FROGED*, n.d.) facilitates communication between the user and the chatbot provides a field for text to be introduced or read. Communicative intents tried by the user may be in the form of written text, and chatbot responses likewise. Chatbot responses and dialogue flow may be facilitated by the incorporation of questions or *buttons,* guiding the dialogue through a series of answers until the desired intent is carried out.

Figures 1 and 2 show examples of how a user can create an account on the DVIL interface as the chatbot asks questions to fill in the information needed to carry out the action.

[4] https://www.apple.com/siri/
[5] https://developer.amazon.com/en/alexa

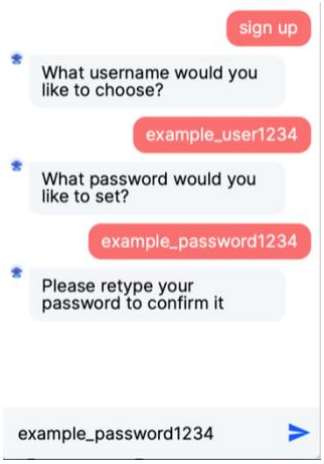


Figure 1. An example of a text-based interaction taking place in the web widget where the chatbot asks questions to collect the information necessary to carry out the user's request to *sign up*.

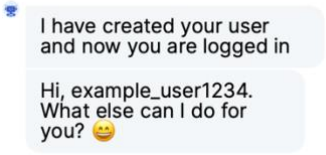


Figure 2. The text response given to convey to the user that the action *sign up* has been completed.

In this example the DVIL chatbot widget contains a field where the user may enter their queries using written text, and the response will also appear in the same window. All dialogue turns are presented in the chatbot widget itself, with the user's message displayed in a pink text bubble aligned to the right, and the chatbot's response in a white text bubble aligned to the left. This structure and colour system visually aids interaction as the user can clearly see the turns of the interaction.

2.1.2.2. Multimedia interaction

Multimedia interaction involves the incorporation of photos, videos, GIFs, etc. (Tipos de Chatbot: Ventajas y Características - Centribal, n.d.). The DVIL chatbot uses a variety of multimedia elements to communicate effectively. Examples of the implementation of these elements will be expanded on in Section 4.1.4.

2.1.2.3. Speech interaction

The incorporation of NLP applications such as Speech-to-Text (STT) technologies and the use of microphones allow for interaction via spoken messages (*Tipos de Chatbot: Ventajas y Características - Centribal*, n.d.). STT technologies convert this spoken message into written text, which the chatbot is then able to interpret as to be able to carry out the desired action.

The DVIL chatbot has a microphone button, allowing users to activate the microphone on their device as to communicate their query using their voice. This feature is currently under development.

2.3.3. Specialised chatbot

Specialist chatbots cover a specific process in-depth, which translates into a narrow spectrum and longer interactions (*Generalist or Specialist: Which Chatbot Is Right for You?*, n.d.). A chatbot can be considered a specialised when the conversations that it is capable of holding are designed around particular processes.

The DVIL chatbot can be considered specialised as its knowledge is designed to complement the functions DVIL interface. Its specialisation includes interpreting intents related to carrying out actions specific to the interface such as selecting layouts, glyphs, and filters, as well as understanding questions and offering help related to the features and functions unique to the interface. This chatbot also can decipher contextual and statistic-based questions users may ask related unique to the structure of the visualisation, annotation methodology and hypotheses for data annotated in the NewsCom-TOX. In turn, the DVIL chatbot can report unique information for different data files. An example of one of these unique questions is seen below in Figure 3.

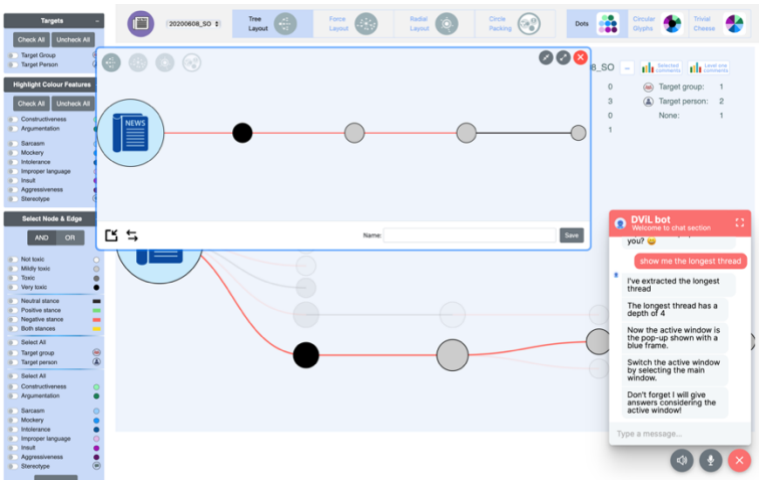


Figure 3. A user request to see the longest thread, a chatbot exclusive function of the DVIL interface

2.3.4. Open-Source

An Open-Source platform provides a chatbot with messaging applications that simulate a conversation between humans. Open-source means the original code for the software is distributed freely and can easily be modified. Open-source software leads to higher levels of transparency, efficiency, and control through shared contributions (13 Best Open Source Chatbot Platforms to Use in 2022 | Botpress Blog, n.d.).

The DVIL chatbot employs the platform Rasa to execute the code necessary for interaction on the DVIL interface. The workings of this platform will be expanded upon in Section 4.

3. DVIL CHATBOT

Summarising the categories explained in the previous section, the DVIL chatbot is an AI chatbot which executes on an Open-Source platform. Its knowledge and functions are specialised towards the DVIL interface with the goal of facilitating the visualisation of data and the statistical analysis of the NewsCom-TOX corpus. The DVIL chatbot itself allows users to analyse their corpus by requesting the chatbot to carry out functions that they would otherwise have to carry out manually; the opportunity to carry out actions that are not accessible on the interface itself and to the possibility to ask for help and explanations. The interaction possibilities of the chatbot include text-based interaction, multimedia interaction and speech interaction though a website chat widget.

My objectives for this project involved improving the chatbot from a linguistic point of view in several ways: the first being to improve training *examples* for *intents* and *synonyms* as well as defining new *intents* and *synonyms* and their training *examples*. Additionally, I improved *responses* given by the chatbot, so they were more suitable or provide more useful and concise information, and developed new *responses* that corresponded to newly defined *intents*. The sets of improvements carried out aimed to promote dialogue between the user and the chatbot. By using the resources of the Open-Source platform Rasa, I was able to carry out the necessary improvements to the DVIL chatbot.

4. RASA PLATFORM

In this section I will expand on what the Rasa platform is, its mechanisms and components and how the data file system works to manage dialogues between the user and chatbot.

As previously mentioned, Rasa is an open-source conversational AI platform that allows you to understand and hold conversations and connect to messaging channels and third party systems through a set of APIs. It supplies the building blocks for creating virtual (digital) assistants or chatbots (*Introduction to Rasa Open Source & Rasa Pro*, n.d.) .

There are two mechanisms inside of Rasa that drive the Conversational AI: Natural Language Understanding (NLU) and Dialogue policies (*Introduction to Rasa – Rasa Learning Center*, n.d.).

The NLU mechanism can be understood as the part of the system that interprets raw text and generates machine-readable information (*Introduction to Rasa – Rasa Learning Center*, n.d.). This mechanism allows you to develop the NLU training data for your machine learning model to learn from, by introducing keys including *intent*, *synonym*, *entity* and their corresponding training *examples*, as well as the keys *rule* and *story* and *responses*, the machine learns to identify and assign a user input to a specific set of *rules* which the chatbot then adheres to and carries out the desired request. These concepts will be developed on in section 4.1.

Dialogue Policies refers to the rules that define and predict the actions to take during interaction between user and chatbot. These rules manage and make natural dialogues possible, considering the context of the turn being executed.

Rasa uses YAML and stores the necessary information using YAML files (.yml). YAML is a human-readable data serialization language that is often used for writing configuration files. This file format is used for data rather than documents (*What Is YAML?*, n.d.).

Due to the restrictions of the extension of this paper, I will only expand the development of the NLU training data that are involved in the linguistic improvement the chatbot.

4.1. NLU Training data

The NLU mechanism of Rasa uses YAML as a unified and extendable way to manage all training data, including NLU data, stories and rules (*Training Data Format*, n.d.). The training data can be understood as the examples that are used to train the Machine Learning model and is what makes the chatbot capable of giving responses to questions or carrying out actions requested by the user. In the case of the DVIL chatbot, training data is split over 4 main files. In the following subsections I will introduce the files and introduce the *keys* involved in developing the training data for a successful dialogue to be able to take place between user and chatbot.

4.1.1. nlu.yml file

This file allows us to define *intents* and *synonyms* with their respective *examples*. The unique definitions of these *keys* can be understood as the modification of the configuration settings of the *keys,* which is what allows us to create the NLU training data models for *intents*.
The *intent* key defines a unique attempt, and the key *examples* allows us to provide a set of training examples of the user's attempt to convey their message to the chatbot. In terms of natural language, an intent may be expressed as a question, with varying degrees of politeness, an instruction or command or a word/group of words.

As we see in Figure 4, the configuration settings of the *intent* have been defined as define_commentdepth, which refers to a user's the attempt at asking what comment depth means. This *key* introduces the *examples* which can be understood as the possible ways in which a user may carry out this request. This *intent* is configured with 13 *examples* of how to ask about comment depth.

```
- intent: define_commentdepth
  examples: |
    - Can you tell me what is meant by comment depth?
    - I was wondering if you could explain comment depth to me?
    - Could you explain to me what the depth of the comment refers to?
    - I would like you to explain what comment depth is
    - Would you be able to give me a tutorial for comment depth?
    - I want you to give me an explanation about comment depth
    - I was hoping you could help me understand comment depth
    - I'd like you to define comment depth
    - Would it be possible to have a tutorial for comment depth?
    - Could you possibly go into more detail about what comment depth is for?
    - Give me help on comment depth
    - Guide for comment depth
    - May you inform me on comment depth
```

Figure 4. Intent for a user's request to define comment depth.

At this point I would like to emphasise that this *intent* is specific to the DVIL interface and the structure of the NewsCom_TOX corpus. The purpose of this *intent* lies in the fact that users must be able to ask for and receive information about all information provided about the corpus structure on the interface, and how it is presented in the visualisation. In Figure 5, I have provided an image to show how the user can see *threads* of comments, comments being displayed as nodes and threads as coloured edges between them, and *comment depth* in the thread, being the position taken by the node in the thread itself. I would also like to affirm that this is not considered a *response*, it is simply an illustration of how the interface displays this information. *Responses* to queries will be dealt with in Section 4.1.4.

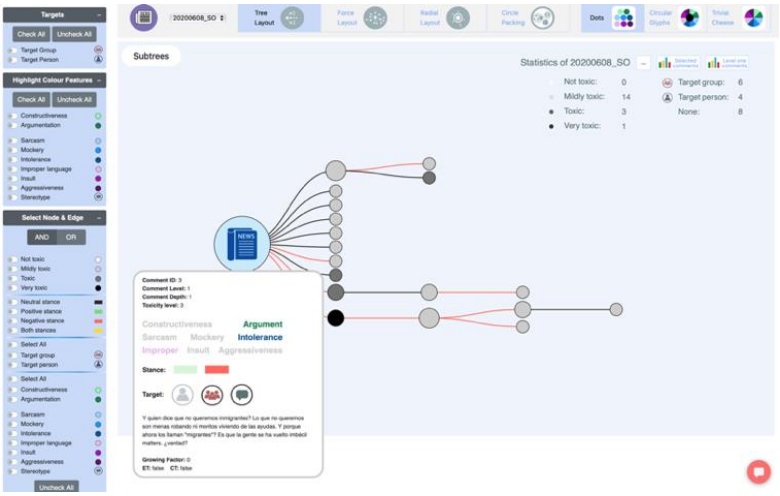


Figure 5. How *threads* and *comment depth* are displayed on the DVIL interface

The development of *intents* will be expanded on in sections 5.2 and 5.3, with some examples provided in *Appendix I* and *II* of *intents* and the phase in which they were developed.

The Rasa platform also gives us the possibility to use the *entity* key in conjunction with the *intent* key. The *entity* key allows us to configure the introduction of a piece of structured information which allows us to extract keywords in a user's input. Figure 6 shows an example of an *intent* that has been configured as a user's request to change the layout to tree layout. This *intent* has 13 training examples which introduce either two or one instances of the *entity* key.

Using the first training example as my example, the first entity that is extracted is *layout* and the second *first visualisation.*

```
- intent: change_layout_tree
  examples: |
    - I want to change the [layout]{"entity": "object_requested", "value": "layout"} to [tree layout]{"entity": "layout_requested", "value": "tree"}
    - Can you swap the [layout]{"entity": "object_requested", "value": "layout"} to [1st visualization]{"entity": "layout_requested", "value": "tree"} layout
    - I was hoping you could bring up [tree layout]{"entity": "layout_requested", "value": "tree"} style
    - Could I possibly choose to [tree visualization]{"entity": "layout_requested", "value": "tree"} type
    - Would you be able to use to [visualisation 1]{"entity": "layout_requested", "value": "tree"} option
    - Could you possibly show me [first visualisation]{"entity": "layout_requested", "value": "tree"} format
    - Would it be possible for you to display features with [1st layout]{"entity": "layout_requested", "value": "tree"} display
    - Show me [layout one]{"entity": "layout_requested", "value": "tree"}
    - Would you visualise the current display as [tree layout]{"entity": "layout_requested", "value": "tree"}
    - I was wondering if I could select [first layout]{"entity": "layout_requested", "value": "tree"} display
    - Let me choose [tree]{"entity": "layout_requested", "value": "tree"} option
    - Let's visualise the selected features in [visualisation one]{"entity": "layout_requested", "value": "tree"} format
    - Change layout to [tree]{"entity": "layout_requested", "value": "tree"} design
```

Figure 6. Intent for the user requesting to change to *tree layout* and its examples with *entity* keys.

In this context, the use of the *entity* key is beneficial for the development of the training data as we can extract entities from the user's request. Additionally, entities can be used in conjunction with the *synonym* key.

The *synonym* key allows us to configure *examples* for the same *entity* defined in the *intent*. With reference the *entities* defined in Figure 6, in the following Figures 7 and 8 I will exemplify the use of the *synonym* key and its use with the *entity* key. In both examples, the key *synonym* introduces *examples* of synonymous instances that should also be interpreted as an entity with the same value as the defined by the synonym key. Figure 7 introduces *examples* synonymous with the entity *layout.*

```
- synonym: layout
  examples: |
    - visualisation layout
    - visualization layout
    - display layout
    - design
    - visualisation design
    - display design
    - display type
    - visualisation type
    - display layout
    - visualisation
```

Figure 7. 10 synonymous *examples* for the *entity layout.*

In the same way, Figure 8 introduces *examples* synonymous with the entity *tree.*

```
- synonym: tree
  examples: |
    - tree layout
    - first visualization
    - 1st visualization
    - 1 visualization
    - visualization 1
    - visualization one
    - 1st visualisation
    - 1 visualisation
    - visualisation 1
    - visualisation one
    - first layout
```

Figure 8. 11 synonymous *examples* for the entity *tree.*

Using these *examples*, the *synonym* key maps the *entities* to their value. Referring to Figure 6, this *intent* defines instances of a user request to change the layout to tree layout. *Synonyms* will be interpreted in the same way and the extracted entities *layout* and *first visualisation,* even if these exact examples are not defined as *examples* for the *intent*. In Figure 9, I present an example of how the NLU mechanism is capable of extracting *entities* from a user's request to change the layout to tree layout, mapping the *entities* to their values using *synonym* keys shown in Figures 7 and 8.

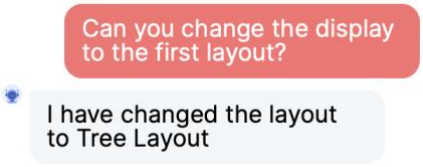


Figure 9. An example of the *synonym* key mapping the *entities* to their value, allowing for synonymous *examples* to be recognised as the same *entity*.

4.1.2. rules.yml

The rules.yml file is designated for defining rules. *Rules* are a type of training data used to train your assistant's dialogue management model. *Rules* describe short pieces of conversations that should always follow the same path (*Rules*, n.d.). In the following example, Figure 10 shows the configuration of the *rule* key. The key *rule* starts the rule and defines a unique rule scenario. The *steps* configure the order of events that must occur. The *intent* defines the query posed user and the *action* defines action that the chatbot must carry out to complete the *rule*.

```
- rule: user asks for tutorial
  steps:
    - intent: ask_tutorial
    - action: utter_tutorial_mode
```

Figure 10. The implementation of the *rule* key in Rasa

4.1.3. stories.yml

The *stories.yml* file is for configuring *story* examples. A *story* is a representation of a conversation between a user and an AI assistant, converted into a specific format where user inputs are expressed as *intents* (and *entities* when necessary), while the assistant's responses and actions are expressed as action names (*Stories*, n.d.). Figure 11 shows an example of series of *steps* necessary to deal with a scenario in which the user requests for help. The key *story* defines a specific scenario and introduces the *steps* that would occur in the *story*, with *intents* and *actions* defined as necessary to complete this dialogue.

```
- story: Help Scenario
  steps:
    - intent: greet
    - action: utter_offers_help
    - intent: reminder_help
    - action: utter_reminder_help
    - intent: reminder_highlevel_questions
    - action: utter_reminder_highlevel_questions_one
    - action: utter_reminder_highlevel_questions_two
    - intent: thanks
    - action: utter_you_are_welcome
    - intent: help_main_graph_barchart
    - action: utter_help_main_bar_one
    - intent: statistics_help
    - action: utter_help_main_bar_three
    - intent: help_layouts
    - action: utter_help_layouts_one
    - action: utter_help_layouts_two
    - intent: help_subtree_barchart
    - action: utter_help_subtree_bar
```

Figure 11. The implementation of the *story* key to define a Help Scenario

Both *rules* and *stories* are used to develop the chatbot's dialogue management model. However, the story key has a specific advantage that they allow us to generalise unseen conversation paths, resulting in the generation of a model which permits more natural interaction between the user and chatbot (Stories, n.d.).

4.1.4. domain.yml

The domain.yml file has two purposes. First, it is where the contents of the previous files are combined. The *keys* of the other NLU data files are stated and organised as follows: intents,

actions, forms, entities. Given that all the NLU data developed is then defined in this file, its function is to allow conversation to take place between the user and the chatbot.

The domain.yml file also allows us to configure the *responses* key, permitting us to define the *responses* given by the chatbot. A *response* can be understood as the response invoked from the chatbot after receiving the input message according to the rules defined in the *rules.yml* or *stories.yml* files. The *responses* key defines what actions or *responses* the chatbot will carry out as a result.

The *responses* key allows us to write a *text* which will be the prompt for the chatbot. We can also provide response variations, or we can more richer responses by adding buttons or images (*Responses*, n.d.). Table 2 provides an example of the execution of an intent as the NLU data is compiled in the domain.yml file so that successful interaction can take place.

nlu.yml	rules.yml	domain.yml

- intent: filters_group
  examples: |
  - Can I select [target group]{"entity": "filter_name"}
  - Choose [target group]{"entity": "filter_name" }
  - Check [target group]{"entity": "filter_name" }
  - May you tick [target group]{"entity": "filter_name"}
  - Show me [target group]{"entity": "filter_name"}
  - I want to see [target group]{"entity": "filter_name"}
  - I want you to check [target group]{"entity": "filter_name"}
  - Activate [target group]{"entity": "filter_name"}
  - Enable [target group]{"entity": "filter_name"}
  - Highlight [target group]{"entity": "filter_name"}
  - Turn on [target group]{"entity": "filter_name"}
  - Could you tick off [target group]{"entity": "filter_name"} filtering
	- rule: asking user which filter target group
	  steps:
	  - intent: filters_group
	  - action: utter_filters_target_group
		- filter_test:
		  use_entities:
		- filters_group
		
		utter_filters_target_group:
		  - text: "How do you want to display it?"
		  buttons:
		  - title: "Targets"
		  payload: '/icon_feature_group'
		  - title: "Select Node & Edge"
		  payload: '/highlight_check{"filter_name": "target group"}'
		
		utter_icon_feature_group:
		  - text: "intent_show_targets;target-group;I have checked Target Group"
		
		if (!breakLoop) {
		  if (filters.length === 1) {
		  e.text = "I checked the filter"

- Switch on [target group]{"**entity**": "filter_name"} filter - I was hoping you could check [target group]{"**entity**": "filter_name"} - I was wondering if you could enable [target group]{"**entity**": "filter_name"} filtering		

Table 2. Example of domain.yml compiling information from other files.

In this example, the user requests to select the *target group* filter. In the nlu.yml file, the training *examples* provided under the intent *filters_group* extract the *target group* entity. The *rule.yml* file states that upon the input of the intent *filters_group,* the action *utter_filters_target_group* must be carried out. The domain.yml file first reiterates the intent *filters_group* (and that this intent allows for *entities* to be extracted) and the action in the form of a *response* is developed, allowing the *rule* to be completed and the desired interaction to take place.

As stated in Section 3, the DVIL chatbot uses different interaction types. Table 3 provides examples of *responses* that the DVIL employs, and how this *response* is displayed in the widget and on the interface.

Response type	Presentation of the response on the DVIL interface
Text only	
With variations	

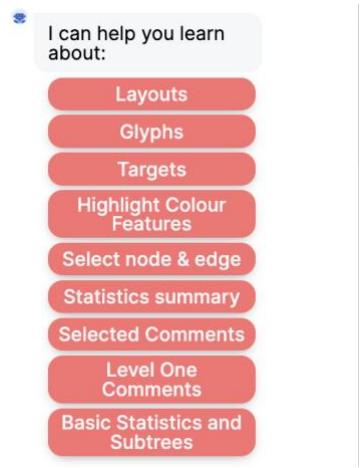
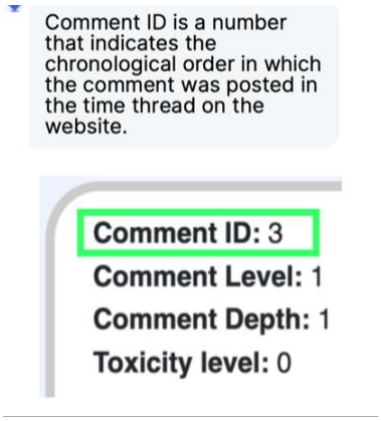
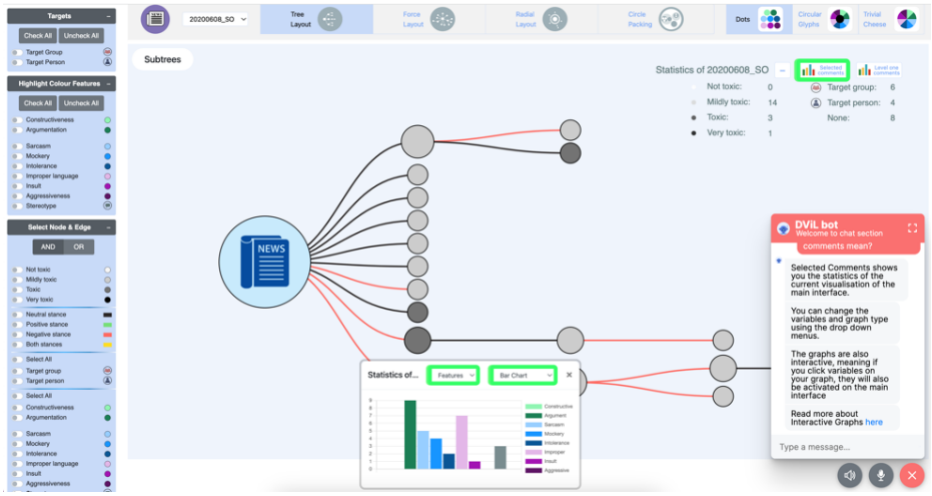
With buttons	
With images	
With visual enhancements and links	

Table 3. Examples of different types of responses

Appendix III provides an additional column to Table 3, which includes the corresponding configuration settings (as modified in the domain.yml file), resulting in the development of these unique responses.

5. IMPROVEMENTS TO THE DVIL CHATBOT

To carry out improvements to the DVIL chatbot I staged my progress into three phases: Phase one: Exploration and defining improvements, Phase two: Improving training examples and responses, Phase three: Developing new intents.

5.1. Phase one: Exploration and defining improvements

Phase one involved learning about the interface levels of the tool (*Initial interface, Uploading interface* and *Analysis interface)* and their functions, the interaction possibilities with the chatbot and the development of a first set of improvements related to the interface itself and chatbot improvement.

To summarise, I first suggested modifications to all interface levels, in particular changes to titles, field and button names and distributions of filters. Due to the restriction on the length of this paper, I will only expand on the *Analysis interface,* its functions and the modifications that were made as this where most interaction between the user and the chatbot occurs due to the variety of functions available and the modifications of this interface display were most significant. Appendix IV provides images all the interface levels.

5.1.1. Analysis interface

This interface is where the user carries out their analysis, offering a range of options that visualise their data. These options include layouts, glyphs, filter options (*Targets, Colour Features, Node and Edge*), statistics and subtrees. In Figure 12, I provide an image of the analysis interface.

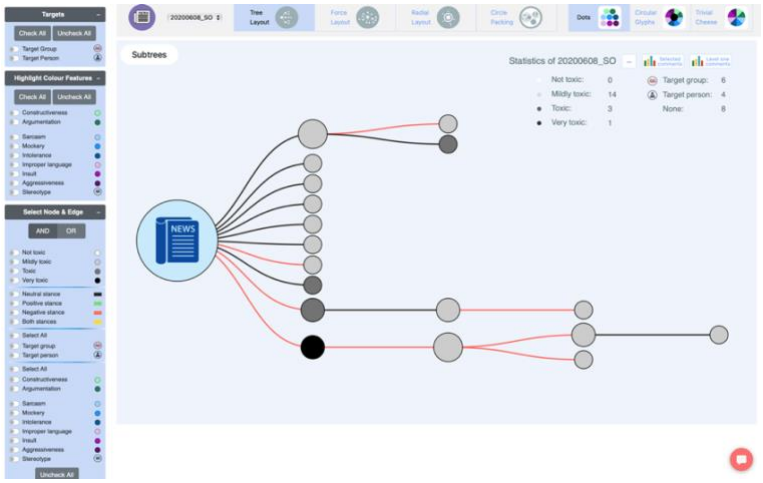


Figure 12. Analysis interface

It is also important to point out that although all actions available on the interface may be carried out by interacting with the chatbot, there are some functions of the DVIL tool that are exclusively accessible through interaction with the DVIL chatbot such as selecting threads and subtrees, basic statistic questions and help.

5.1.1.1. Modifications to the Analysis interface

Modifications to this interface included changing menu names to *Highlight Colour Features* and *Select Node and Edge*; changing *Summary* to *Statistics*, removing *Stereotypes* from the *Statistics* summary and *Target* menu and added to the *Highlight Colour Features* and *Select Node and Edge* menu, the addition of *Both stances* under the *Select Node and Edge* menu. These changes aimed to provide the user the appropriate language on the interface to allow them to access the functions of the interface through the chatbot.

5.1.1.2. Improvements to the DVIL chatbot on the Analysis interface

During the exploration process I highlighted a selection of improvements that would be most beneficial to implement to create a more functional and useful chatbot for the DVIL tool. These improvements were divided into two categories: the improvement of intents/ responses and the development of new intents/ responses.
As for the improvement of *intents* and *responses*, I first noticed that almost all basic functions that the interface offered were difficult to execute through interaction with the chatbot. This issue would be solved through the improvement of examples related to these intents. I will go into more detail about how I developed these training examples in section 5.2. Phase two: Improving training examples and responses.

Moreover, I identified many issues related to the responses that the chatbot gave. I categorised these responses into *inappropriate responses,* corresponding to responses provide incorrect information for the query posed; *impossible responses* that affirmed actions had been carried out when they had not or these actions were impossible to carry out, and *general incoherencies* referred to mistakes in grammar or spelling or incoherencies between British English and American English. Additionally, in some cases, when the intent was not recognised by the chatbot, *no response* was given.

The correction of these responses would be dealt with in one of two ways. The first solution involved improving corresponding *intent* training examples (as the chatbot would be able to interpret user queries better, providing a correct response). Some of the problematic responses would also need to be solved by creating new *intent/response* pairs.

As for the development of new *intents/responses* (Phase three*),* I began by highlighting *missing intents* which corresponded to intents that would complement intents that already existed but were missing. For example, if there is the option to select filters from one menu, we must also have this option for all filters. Additionally, due to the complexity of the tool, I decided it was necessary to implement help-related intents and responses for users to be able to take advantage of all the tool possibilities and with the aim of encouraging users to interact with the chatbot.

To briefly summarise my findings and suggestions of the first phase, it was very clear that the interface itself was very complex, yet this visualisation tool could be extremely useful for annotators to analyse their data using visualisation methods. However, for the chatbot to complement the tool and be useful for the user, more training examples must be added for the *intents* and new *intents* which complement the original intents must be developed. Access to these *intents* must be aided with language provided on the interface and help must be easily accessible and useful for the user to take advantage of the resource.

5.2. Phase two: Improving training examples and responses

This phase involved carrying out improvements to *intents* and *responses* using the Rasa platform. I first began modifying *general incoherencies* such as British/American English, grammatical and spelling mistakes found in the chatbot *responses* and examples under the *synonym* keys.

Next, I began the most significant task of this phase which involved developing the training *examples* of these *intents* and *synonyms* to facilitate successful interaction with the user and chatbot. To be able to do this, I aimed to identify lexico-syntactic patterns and develop lists of synonyms using linguistic resources including WordNet®[6] and WordReference[7] in order to define a variety of training *examples*.

[6] Princeton University "About WordNet." WordNet. Princeton University. 2010
[7] https://www.wordreference.com

5.2.1. Linguistic resources

To search words and see basic examples of context to develop lists of synonyms for verbs such as *check*, *highlight*, *choose*, *change,* I used WordNet® and WordReference.

WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

WordReference provides a range of bilingual and monolingual dictionaries including English synonyms and usage. By consulting these resources, I identified synonymous words and developed lists to paraphrase *examples* that would be used to develop the NLU training data.

It is important to note that some synonyms identified would be used as training *examples* for *entities* with a corresponding *synonym* key, and others would form part of synonym lists that I would use to paraphrase the lexico-syntactic patterns identified in order to develop *examples* for each *intent*.

For example, to improve the training *examples* for the intent *colour_feature_sarcasm*, which is configured to interpret a user's request to select the sarcasm colour feature, I developed four synonym lists, corresponding to the underlying lexico-syntactic structures defining the request: *I want to*, *check*, *visualise* and *colour feature.* These lists are displayed in the following Figures 13, 14, 15 and 16.

```
#- synonym: I want to
#  examples: |
 #   - I want to
 #   - I would like to
 #   - I'd like to
#    - Can you
#    - Can I
#    - May you
 #   - May I
#    - Could you
#    - Could I
#    - I want you to
#    - I would like you to
#    - I'd like you to
#    - I was hoping you could
#    - I was wondering if you could
#    - Would I able to
#    - Would you be able to
#    - Would it be possible for you to
#    - Could you possibly
#    - How about you
```

Figure 13. Synonym list for *I want to*

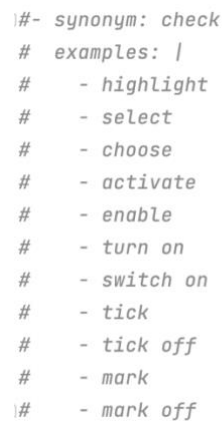


Figure 14. Synonym list for *check*

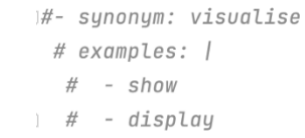


Figure 15. Synonym list for *visualise*

Figure 16. Synonym list for *colour feature*

By substituting synonyms from the lists into the corresponding parts of the lexico-syntactic patterns identified, I was able to implement more training *examples* in the following way as shown in Figure 17.

```
- intent: colour_feature_sarcasm
  examples: |
    - Highlight sarcasm colour feature
    - Can you highlight sarcasm glyph
    - Check sarcastic colour filter
    - Switch on sarcasm colour feature
    - Show me sarcasm colour feature
    - May I tick sarcasm colour feature filtering
    - Check sarcastic colour
    - I want to see sarcasm colour feature
    - I would like to see sarcasm colour feature
    - Turn on sarcastic colour filtering
    - Could I switch on sarcasm colour feature
    - Select sarcastic colour feature filter
    - Choose sarcasm colour
    - Activate sarcasm colour feature
    - Enable sarcasm colour feature
    - Tick off sarcasm colour feature
    - I was hoping you could check sarcasm colour feature
    - I was wondering if you could select sarcasm colour filter
    - Would you be able to mark off sarcasm colour glyph
```

Figure 17. The *intent* and training *examples* behind the request to select the sarcasm colour feature

I would also like to reaffirm that given that Rasa uses Machine Learning to develop its NLU model, it is not necessary to use every synonym from the lists for the chatbot to recognise *intents* (Figure 18).

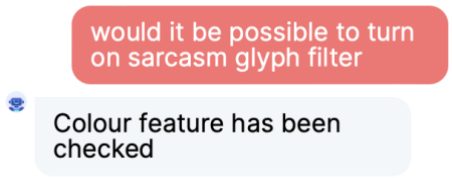


Figure 18. An example of a user request to select the sarcasm colour feature using examples from the synonym lists that are not defined in the training examples of the intent *colour_feature_sarcasm*.

Given that the request uses a lexico-syntactic structure as defined in the training *examples* of the *intent*, the *intent* is recognised as an attempt to select the sarcasm colour feature. By implementing like structures and synonyms, I was able to improve 24 intents. Examples of these *intents* are provided in *Appendix I* and *II*.

As to improve *responses*, I paraphrased responses and provided some responses with *variations* (as exemplified in Table 3) using synonyms from the lists I developed.

5.3. Phase three: Developing new intents

I developed *intents* related to interface functions and help-based intents and responses to guide the user though analysis. The motivation behind these new intents was due to the variety and complexity of functions the DVIL interface has to offer, for the interface to be functional and accessible the user should always be able to access help. The development of these help options can be split into four main categories: Back-up dialogues, Defining the interface contents, Tutorial and Repurposing help-related intents. A total of 94 new intents were defined in this phase.

5.3.1. New functions

I developed new intents that would complement the functions that are already available on the DVIL interface. This involved the possibility of selecting all options and filters and deselecting all filters. Some examples can be seen in *Appendix I* and *II*.

5.3.2. Back-up dialogues

Back-up dialogues involved asking the user an additional question if they have not provided enough information for the intent to be carried out effectively. The implementation of these dialogues is based around selecting filters without specifying a menu or indicating a menu without naming the filter. These intents are useful as given that the same filter names appear in various menus, and with long menu names, users may choose not to add this information to their request.

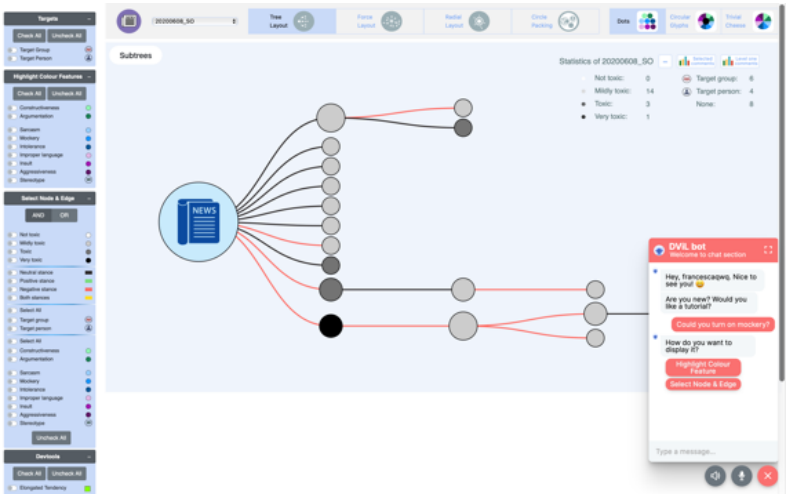


Figure 19. A back-up dialogue for when a user omits the menu from their request to select the *mockery filter*

Additionally, an automatic back-up response was implemented as affirming to the user that their request had not been understood, as to assure a response was always given, as opposed to *no response* (*Appendix II, Example 23*).

5.3.3. Definitions and explanations of the interface contents

The purpose defining the interface contents was so that the user can ask questions about anything they see on the interface. This involved the identification of the contextual information, where this is displayed and how it is accessed and possible functions and how they work. Some of these definitions are purely linguistic, for example definitions of targets, toxicity and features, and others add information about the use of the function and their relationship with other functions on the interface. All definition type information provided by the chatbot was complemented with different visual aids such as outlining or images. In Figure 20, I provide an example of an interaction between the user and the chatbot when the user has asked what glyphs are for.

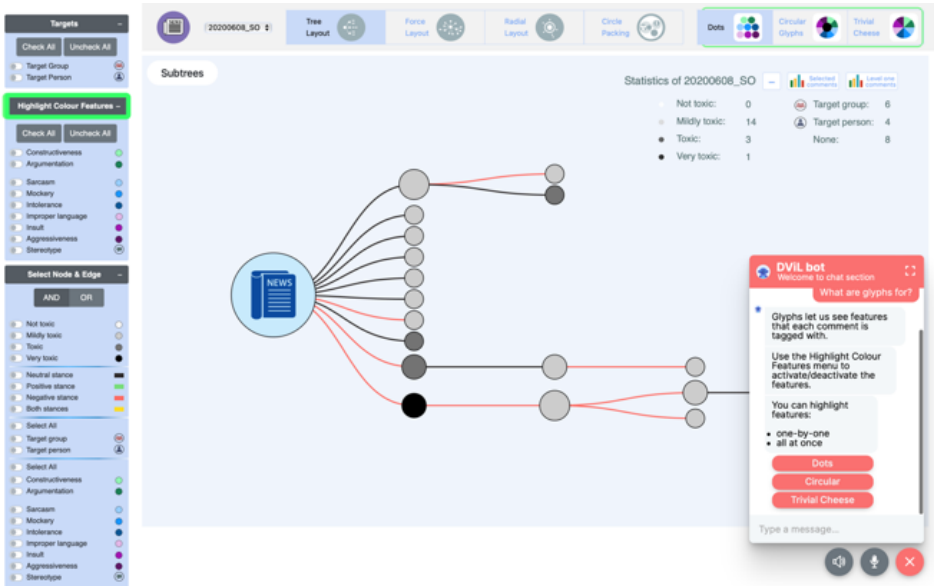


Figure 20. The DVIL chatbot using text and visual enhancements to respond to the user request to learn more about glyphs.

5.3.4. Tutorial

The development of the tutorial was a key part to adding help into the chatbot. I aimed to provide the opportunity to learn about the tool. This information is accessible in two ways: the chatbot offers this option automatically when the user first enters the main interface, or the user

can request a *tutorial.* Due to the large amounts of information presented in the tutorial, the response is structured by function and has buttons which the user must click to continue the tutorial. Visual enhancements as seen in Figure 20 are also implemented.

In Figures 21 and 22 you will see a small section of the of the tutorial and the visual aids designed to guide the user through the tools of the interface.

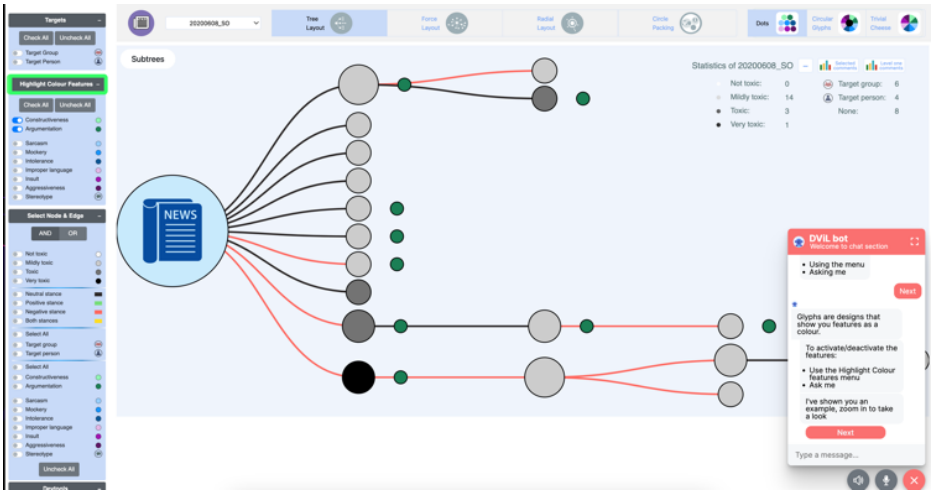


Figure 21. The third step in the tutorial explaining *glyphs* with written text and visual enhancements.

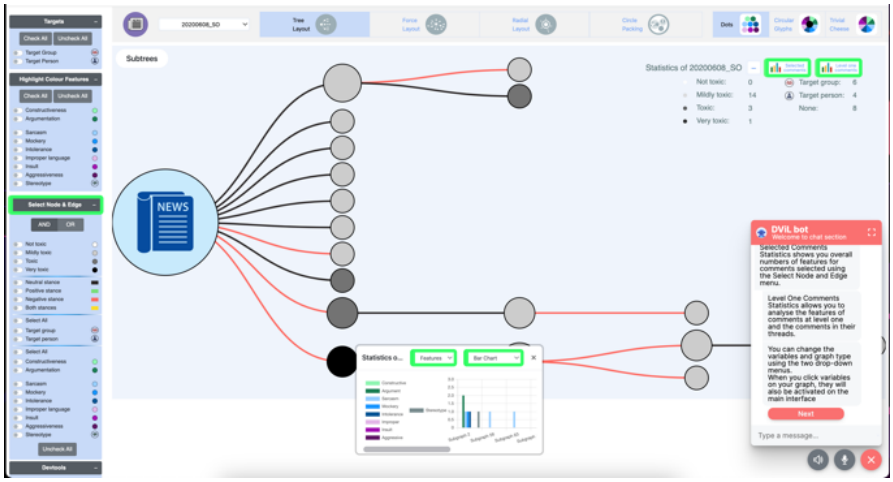


Figure 22. The seventh step in the tutorial explaining *statistics* with written text and visual enhancements.

5.3.5 Repurposing help-related intents

The repurposing of help-related intents allowed me to create two sub-tutorial intents that would allow users ask to recap how the tool works and the functions that are exclusive to the chatbot. The modification to these intents involved developing a more appropriate training set, and the

modifications of the responses now corresponded with the modifications to the chatbot interface and the new functions. Moreover, these intents included buttons. These buttons allowed more intents to be repurposed as multiple responses are available to the user, allowing them to access large amounts of information, structured through guided dialogue and comprehensible chunks. In Figures 23, 24 and 25, I show examples of a *repurposed intent* that allows users to recall the functions of the tool. The example shows how the button structure allows the users to access information quickly and how it leads to a dialogues completion and an action to be carried out.

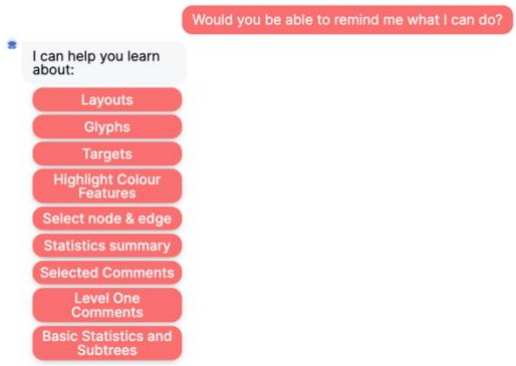


Figure 23. An example of the *repurposed intent* for a user to recall the functions of the tool

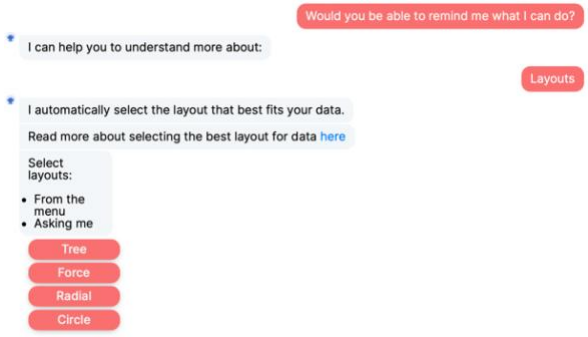


Figure 24. The continued dialogue provided by the chatbot after the user has selected the button *layouts*.

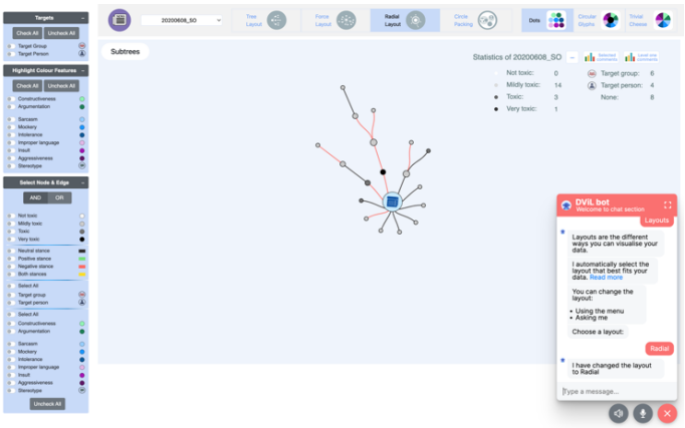


Figure 25. The completed dialogue and response after the user has selected the *Radial* button.

In *Appendix I. Example 23,* I have provided images of the other repurposed intent for users to recall the questions that are specific to the chatbot.

6. EXPLORATORY CHATBOT STUDY

To evaluate the improvements made to the DVIL chatbot, an exploratory study was carried out[8]. Data was collected corresponding to user *intents* and the usability of the chatbot at this stage, which could be used to improve the current version of the chatbot. Moreover, as the DVIL project also involves the development of the tool itself, an additional motivation for the study included obtaining feedback regarding the user's opinions on the most adequate visualisation *layouts* for the dataset.

6.1. Procedure

We invited five participants: 3 female (aged 18-30) and 2 male (aged 18-40). No participant had English as a first language. One female and one male were familiarised with the annotation of the NewsCom-TOX corpus. All participants received an introductory presentation explaining the annotation of the NewsCom-TOX corpus, the DVIL interface, its tools and the chatbot's functions. Each participant was then invited individually to carry out tasks to be

[8] I would like to thank PhD candidate Ecem Kavaz, Dr. Inmaculada Rodríguez and Dr. Anna Puig from the Department of Mathematics and Computer Science who helped with the preparation and the execution of the exploratory study.

solved through interacting with the chatbot, answer questions regarding *layouts* and finally evaluate their experience with the chatbot.

The participants were presented with a questionnaire which introduced the purpose of the study and consent form and the tasks. The consent form informed the participants that the screen of the laptop they would be working on was being recording, as well as their voice. As this study was observational, the other investigator and I took on different roles: moderator and observer. The moderator would help the participant get familiar with the interface, guiding them through the tools of the interface. At this point the moderator also advised participants to not interact with the chatbot. We also helped participants carry out a mock question which involved having to select filters and look at *statistics*. The participant was then allowed to move onto the task questions alone. The role of the moderator from now on would be to intervene when necessary. Moreover, it is important to note that the moderator encouraged participants to speak aloud to understand doubts and their thought processes. If a participant needed help, the moderator first encouraged them to try and ask the chatbot for help, then would intervene to remind them of things they had been previously explained if they forgot, and as a final resort, they would step in by telling them what to do, but not how to ask the chatbot.

The observer (myself) took on a silent role during the tasks, noting number of *successful intents*, number of *failed attempts*, and transcribes the intent, they comment on observations such as if they asked for help, what help was given, their reactions to responses given by the chatbot (pleased, frustrated, confused...) and any other observations as well as noting the final state of the tasks *(complete/incomplete)*.

After completing the two main tasks, the participants rated their experience with the chatbot and had the opportunity to leave more in-depth comments.

Appendix V contains a copy of the questionnaire that was filled in by each participant.

6.2. Observation of intents

In this section I will comment on the styles in which users posed the query and discuss the successful and failed intents. Moreover, I will give categorise the failed queries, present

participant evaluation scores and comments as to develop a series of next set of improvements necessary for the DVIL chatbot to be fully functional.

When interacting with the DVIL chatbot, participants adopted different types of requests:

- The question itself: Some people asked the question from the questionnaire directly. A comment from users was that they would like to be able to ask the question exactly and the chatbot answer with a number/ which features/what level of toxicity or relationships between features. For example, "how many comments have the labels argumentation and constructiveness at the same time?"
- Short intents: Some users used imperative type request. For example, "choose only comments with stereotype" and "show me non toxic comments". Some users also tended to stick to one way of asking when they saw it worked for example, always using the same verb "check" as in "check constructiveness".
- Longer intents: Some longer intents corresponded to questions-style queries or imperatives and defining relative clauses. Some examples include "Can you show me the comments with argumentations?" or "show me the comments that have the label argumentation".

6.3. Error analysis

To assess the improvements implemented, I carried out an error analysis, classifying user attempts as either successful or failed intents.

A successful intent is defined as if the chatbot correctly interpreted the user response and carried out the desired action and/or correct response, or if the intent was not recognised, the chatbot stated that this intent had not been recognised.

Failed attempts refer the *intents* in which the requested to the action was not carried out. I categorised failures by reason for failure: *true failures, impossible* and *chatbot-guided*. A *true failure* can be understood as when the chatbot did not interpret the user response correctly and did not out carry out the desired action and/or offered an incorrect response due to spelling mistakes or poorly constructed petitions (as the participants were not first language English speakers) or that training *examples* did not contemplate an *intent* of this type. Errors classed as *impossible* refer to either *intents* that were not recognised due to this not being a possibility of

the interface (or the chatbot) or actions that are possible on the interface but are not possible to carry out with the chatbot at this point in development such as requests selecting more than one filter or anaphoric references. Finally, errors classed as *chatbot-guided* refer to intents that did not respond to the user's query but offer guidance on how to obtain the information they are looking for such as when the user asked for the number of comments, the chatbot responded with help with statistics or opened the statistics window, showing the user how to find the answer.

In Table 4, I provide a breakdown of the number of successful and failed intents, with their corresponding percentages regarding the total number of intents.

User	No. of intents	No. of successful intents	No. of failed intents		
			True Failure	Impossible	Chatbot-guided
1	25	18 (72%)	1 (4%)	5 (20%)	1 (4%)
2	28	13 (46%)	5 (18%)	9 (32%)	1 (4%)
3	18	13 (72%)	1 (6%)	4 (22%)	0 (0%)
4	27	22 (81%)	1 (4%)	4 (15%)	0 (0%)
5	15	9 (60%)	1 (7%)	3 (20%)	2 (13%)

Table 4. Results summarising the number of successful and failed intents for each user.

Table 5 presents examples of the successful and failed intents observed in the study.

Successful intents		show me the statistics of selected comments
		show me the comments that have the label argumentation
		Can you show me the comments with argumentations?
		Can you give me a definition of constructiveness?
		add the comments that have the label constructiveness
		which of these subtrees is the most toxic?
Failed intents	True Failure	deselect all tags
		Turn of features
		can you also select the option OR?
	Impossible	how many are there?

		show me all comments with argumentation and constructiveness
		Can you explain the intolerance categories
	Chatbot-guided	how many comments are selected right now?
		how toxic are these comments?

Table 5. Examples of successful and failed attempts.

7. CONCLUSIONS AND FUTURE WORK

As part of this final project, my contribution to the improvement of to the DVIL chatbot have involved modifying language on the DVIL interface, improving *intents*, *synonyms* and *responses* as well as the addition of new *intents* related to the functions available on the interface, back-up dialogues, definitions of the interface content, explanations of tools, the implementation of a tutorial and providing repurposed help. I then carried out an exploratory study to evaluate the improvements implemented and to identify areas for improvement to decide my future work.

The results of the exploratory study revealed that with the improvements to the original *intents* and with the complementation of new *intents* related to the actions on the interface, analysis is facilitated, with the number of successful intents outweighing the failed intents in the cases of every participant. Referring to the *failed intents* observed in the exploratory study, some *intents* were classified as *true failures* since the training *examples* I developed did not include *intents* employing those structures or used vocabulary which was not defined. Additionally, participants may have spelled incorrectly or used poorly constructed queries. To correct these failures, I would revise these *intents* to accommodate for more synonymous lexico-syntactic structures.

As for *impossible intents* and *chatbot guided intents*, I would begin by developing *intents* covering all possible functions of the interface. These *intents* would include closing statistics pop-up windows and selecting more than one filter at a time. Moreover, regarding some of the comments left by participants in evaluation section of the questionnaire, it may be possible to improve some chatbot responses to accommodate for intents related to number of comments such as once a filter is selected, the chatbot responds affirming the filter has been selected and the number of comments that are now visualised. It is important to note that given that interface

has statistics tools, having the chatbot directly answer statistic-based queries are not intents of priority, but it will be possible to improve chatbot-guided intents so that future users can find the information more easily.

As for *intents* involving two options (for example, selecting two filters), these types of intents could be made possible in the next phases of development. However, petitions involving the selection across multiple menus is complicated. For example, requesting a layout type and a glyph type at the same time was an intent that I tried to develop but failed, resulting in responses affirming that two-actions are not possible at the same time. More time would have to be spent to develop a possible solution to allow for multiple actions to be carried out. These improvements could be most significant when a user wants to the number of comments tagged with multiple features, or features associated with other features, it could be interesting to have *intents* that allow for multiple filters to be selected and opening statistics.

I would also like to develop help-related intents. These include back-up dialogues such as when the user omits information from the query. Currently there are back-up dialogues for selecting filters and deselecting filters. However, the deselecting filters back-up has a significant contextual flaw. If the participant has the same filter selected in both menus and asks to remove the filter without specifying the menu, the chatbot asks the user to confirm where they want the filter removing from. However, this dialogue is also prompted if the user only has the filter selected in one menu, but omits the menu name, thus the chatbot is not taking into account the previous context being that when the filter is referred to, there is only one possible answer. As for definitions, explanations, tutorial and repurposed help, some responses could be modified in order to clarify some doubts that users had such as *statistics* and *subtrees*. Additionally, at this point the tutorial is relatively static. More variety may be added in order to make it more interactive. However, given that in the study the tutorial was not accessed by any user, further studies would need to take place to understand where and how users may benefit most from interactive help.

My work in this project did not extend to the improvement of chatbot exclusive queries relating to threads, subtrees and basic-statistics questions. In the study, users had issues understanding how to access statistics for subtrees. As part of a future project, I would like to develop more help around the subtrees feature and develop *intents* so that all the functions in the window are available through the chatbot. Some complications may arise around these new intents for

example if a user was to request statistics, it would have to be understood that the user is asking for the statistics of the subtree, and not the statistics of the main interface. One solution may be back-up dialogues or improved responses to avoid ambiguity as to what is being displayed. In addition, the identification of other substructures of interested to be displayed, starting with requests stated by participants from the study including shortest thread. Finally, more complex intents involving anaphora and contextually dependent dialogue flows would be important to develop in order to recreate a more human-like interaction experience.

As a final step taken to improve the DVIL chatbot, the observational study carried out will be extended to more participants in preparation for the publication of an article related to the findings of this study. In this article I will have the opportunity to discuss the results that I have not developed on in this study such as the user evaluations of the experience, information regarding their reactions to responses and emotions expressed and specific comments.

8. REFERENCES

Taulé, M., Ariza, A., Nofre, M., Amigó, E., & Rosso, P. (2021). Overview of DETOXIS at IberLEF 2021: DEtection of TOXicity in comments In Spanish. *Procesamiento Del Lenguaje Natural*, *67*(0), 209–221. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6390

Kavaz, E., Puig, A., Rodriguez, I., Taule, M., & Nofre, M. (2021). Data Visualisation for Supporting Linguists in the Analysis of Toxic Messages. *Computer Science Research Notes*, *3101*, 59–70. https://doi.org/10.24132/CSRN.2021.3101.7

Kavaz, E., Puig, A., Rodríguez, I., Chacón, R., De-La-Paz, D., Torralba, A., Nofre, M., & Taule, M. (2022). Visualisation of hierarchical multivariate data: Categorisation and case study on hate speech. *Https://Doi.Org/10.1177/14738716221120509*, *22*(1), 31–51. https://doi.org/10.1177/14738716221120509

What is a Chatbot | Oracle. (n.d.). Retrieved June 7, 2023, from https://www.oracle.com/chatbots/what-is-a-chatbot/

NLP Chatbot: What is Natural Language Processing and How It Works? (n.d.). Retrieved June 7, 2023, from https://www.xenioo.com/nlp-chatbot-what-is-and-how-works/

20 Fascinating Chatbot Applications From Six Key Industries. (n.d.). Retrieved June 7, 2023, from https://www.freshworks.com/freshdesk/customer-service/chatbot-applications-blog/

Tipos de chatbot: ventajas y características - Centribal. (n.d.). Retrieved June 8, 2023, from https://centribal.com/es/tipos-de-chatbot-ventajas-y-caracteristicas/

Eduard Vives Isern. (2023). *ChatbotDViL: Diseño y implementación de una interfaz en lenguaje natural para visualización de datos*.

Best Types of Chatbots for 2023|Rule-Based Chatbots vs AI Chatbots. (n.d.). Retrieved June 8, 2023, from https://mindtitan.com/resources/guides/chatbot/types-of-chatbots/#how-ai-chatbot-is-different

What is a chatbot? | IBM. (n.d.). Retrieved June 8, 2023, from https://www.ibm.com/topics/chatbots

What's in a Chat Widget? - FROGED. (n.d.). Retrieved June 8, 2023, from https://froged.com/blog/whats-in-a-widget

Generalist or Specialist: Which Chatbot is Right for You? (n.d.). Retrieved June 8, 2023, from https://www.spixii.com/blog/generalist-or-specialist-which-chatbot-is-right-for-you

13 Best Open Source Chatbot Platforms to Use in 2022 | Botpress Blog. (n.d.). Retrieved June 8, 2023, from https://botpress.com/blog/open-source-chatbots

Introduction to Rasa Open Source & Rasa Pro. (n.d.). Retrieved June 8, 2023, from https://rasa.com/docs/rasa/

Introduction to Rasa – Rasa Learning Center. (n.d.). Retrieved June 8, 2023, from https://learning.rasa.com/conversational-ai-with-rasa/introduction-to-rasa/

What is YAML? (n.d.). Retrieved June 8, 2023, from https://www.redhat.com/en/topics/automation/what-is-yaml

Training Data Format. (n.d.). Retrieved June 8, 2023, from https://rasa.com/docs/rasa/training-data-format/

Responses. (n.d.). Retrieved June 8, 2023, from https://rasa.com/docs/rasa/responses/

Rules. (n.d.). Retrieved June 8, 2023, from https://rasa.com/docs/rasa/rules/

Stories. (n.d.). Retrieved June 8, 2023, from https://rasa.com/docs/rasa/stories/

9. APPENDICES

Appendix I. Examples of *intents* their *examples* as developed using the Rasa platform with their corresponding phase of development

Intent	Phase

Example 1. Basic formality

`- intent: greet`
`  examples: |`
`  - hey`
`  - hello`
`  - hi`
`  - hello there`
…
	2

Example 2. Request to change to the tree layout/ Action possible on interface

`- intent: change_layout_tree`
`  examples: |`
`  - I want to change the [layout]{"entity": "object_requested", "value": "layout"} to [tree layout]{"entity": "layout_requested", "value": "tree"}`
`  - Can you swap the [layout]{"entity": "object_requested", "value": "layout"} to [1st visualization]{"entity": "layout_requested", "value": "tree"} layout`
`  - I was hoping you could bring up [tree layout]{"entity": "layout_requested", "value": "tree"} style`
`  - Could I possibly choose to [tree visualization]{"entity": "layout_requested", "value": "tree"} type`
`  - Would you be able to use to [visualisation 1]{"entity": "layout_requested", "value": "tree"} option`
`  - Could you possibly show me [first visualisation]{"entity": "layout_requested", "value": "tree"} format`
…
	2

Example 3. Request to change the glyph type to circular/ Action possible on interface

`- intent: change_glyph_circular`
`  examples: |`
`  - Would it be possible for you to display features with to [circular]{"entity": "glyph_requested", "value": "circular"} style glyph`
`  - Show me [circular]{"entity": "glyph_requested", "value": "circular"} style`
`  - Would you visualise the colour features as [circular]{"entity": "glyph_requested", "value": "circular"} glyphs`
`  - I was wondering if I could select [circular]{"entity": "glyph_requested", "value": "circular"} display`
`  - Let me choose [circular]{"entity": "glyph_requested", "value": "circular"} option`
`  - Let's visualise the selected features in [circular]{"entity": "glyph_requested", "value": "circular"} glyphs`
…
	2

Example 4: Request to select a named filter from the Select Node and Edge menu/ Action possible on the interface

<pre>- intent: highlight_check
  examples: |
    - Highlight [non toxic]{"entity": "filter_name", "value": "non toxic"} comments
    - May you tick [positive]{"entity": "filter_name"} nodes
        - Check [negative]{"entity": "filter_name"} edges
    - Show me [negative stance]{"entity": "filter_name"} edges filter
    - Highlight [neutral]{"entity": "filter_name"} comments
    - Can I highlight [neutral stance]{"entity": "filter_name"} nodes
    - Check [neutral]{"entity": "filter_name"} edges
    - Show me [neutral stance]{"entity": "filter_name"} edges filter
    - Tick off [both stances]{"entity": "filter_name"} nodes filtering
    - I want to see [target person]{"entity": "filter_name"} nodes and edges filtering
    - Check [stereotype]{"entity": "filter_name"} edges
    - Show me [stereotype]{"entity": "filter_name"} edges filter
    - Highlight comments with [all targets]{"entity": "filter_name"}
    - Can I highlight [both targets]{"entity": "filter_name"} nodes and edges
    - Check [all targets]{"entity": "filter_name"} edges
    - Show me nodes and edges with [both targets]{"entity": "filter_name"}
    - I want to see nodes with [constructiveness]{"entity": "filter_name"}
    - Highlight [argumentation]{"entity": "filter_name"} node and edge
    - I want to see [sarcastic]{"entity": "filter_name"} nodes
    - Select the node and edge filter [intolerance]{"entity": "filter_name"}
        - Turn on [intolerance]{"entity": "filter_name"} nodes
        - Show me [improper language]{"entity": "filter_name"} nodes
    - Show [insult]{"entity": "filter_name"} comments
    - Select [all features]{"entity": "filter_name"} node and edge
    - Show me nodes with [all levels of toxicity]{"entity": "filter_name"}
    - Activate the select [all levels of toxicity]{"entity": "filter_name"} filter
     - Enable the filter select [all stances]{"entity": "filter_name"}
…</pre>
	2

Example 5. Request for help for 'Selected comments'/Improvement of a pre-existing help related intent

<pre>- intent: help_main_graph_barchart
  examples: |
    - What are selected comments?
    - How can I analyse selected comments data?
    - I want to know how many comments have the features I have selected
    - Can I know how many of comments are tagged?
    - How can I see the number of comments?</pre>
	2

- How can i know the number of comments with a certain feature - Can I know what features my comments are tagged with? …	

Example 7. Request to see a subregion of the visualisation: The most toxic subtree

- intent: most_toxic_subtree examples: | - Show me the most toxic subtree - I want to see the most toxic subtree - Which subtree is the most toxic? - Can you show me the most toxic subtree? - I would like to see the most toxic subtree …	2

Example 7. Request to change filter option to OR in the Select node and edge menu/ Action possible on interface

- intent: switch_or examples: | - I want to change to or - I would like to switch to OR - Can you to adjust to OR - Could I want to change to and to or - I was hoping you could switch and for or …	3

Example 8. Request to deselect all filters in the Select node and edge menu/ Action possible on interface

- intent: highlight_uncheck_all examples: | - Could you possibly disable edges filters - I would like you to turn off all nodes filtering - Take off all edges filtering - Would it be possible for you to get rid of nodes filters - Clear the edges filters …	3

Example 9. Request to select the target 'target person' from the Targets menu/ Action possible on interface

- intent: icon_feature_person examples: | - Choose target person icon - Tick off target person icon feature - I was hoping you could check target person icon feature - I was wondering if you could select target person icon filter …	3

Example 10. Request to deselect the target 'target group' from the Targets menu / Action possible on interface

- intent: uncheck_target_group examples: | - Could you hide target group icon features - Would you be able to untick target group icon filters - Could you possibly clear target group icon filtering - Would it be possible for you to unmark target group icon filters - Unselect target group icon filtering	3

```
    - Get rid of target group icon features
    - Would I be able to take target group icon off
…
```

Example 11. Request to select all the targets from the Targets menu / Action possible on interface

```
- intent: check_all_target_icons
  examples: |
    - Highlight all target icons
    - Can I highlight all target icon features
    - Check all target icon features
    - Switch on all target icon filters
    - Show me all target icon filtering
…
```

3

Example 12. Request to deselect all the targets from the Targets menu / Action possible on interface

```
- intent: uncheck_all_target_icons
  examples: |
    - Can I disable all target icons
    - Uncheck all target icons features
    - May you turn off all target icons filters
    - Would it be possible for you to unmark both target icon
filters
    - Get rid of both target icon features
    - Would I be able to take both target icons off
…
```

3

Example 13. Request to select the feature 'sarcasm' from the 'Highlight colour features' menu/ Action possible on interface

```
- intent: colour_feature_sarcasm
  examples: |
    - Highlight sarcasm colour feature
    - Can you highlight sarcasm glyph
    - Check sarcastic colour filter
    - Would you be able to mark off sarcasm colour glyph
…
```

3

Example 14. Request to deselect the feature 'aggressiveness' from the 'Highlight colour features' menu/ Action possible on interface

```
- intent: colour_feature_aggressiveness_uncheck
  examples: |
    - I was wondering if you could deactivate aggressiveness
colour glyph filters
    - Would you be able to untick aggressive colour filtering
    - Clear off aggressive colour feature filter
    - May you remove aggressiveness colour filtering
…
```

3

Example 15. Request to select all features from the 'Highlight colour features' menu/ Action possible on interface

```
- intent: check_all_colour_features
  examples: |
    - Select all colour features
    - Choose all colours
    - Activate all colour features
```

3

- Enable all colour features filters - I was hoping you could check all colour features …	

Example 65. Request to deselect all features from the 'Highlight colour features' menu/ Action possible on interface

- intent: uncheck_all_colour_features examples: | - Can I disable all colour features - Uncheck all colours - May you turn off all colour features filters - I want to reset all colour filters - Deselect all colour filtering …	3

Example 17. Request to open the 'Level One Comments' statistics option/ Action possible on interface

- intent: statistics_features_subtrees examples: | - I was hoping you might bring up the statistics of comment level one - Would you be able to open the level one comments statistics - Could you possibly help me compare level one comments - Analyse level one comments - I'd like you to show me a chart for comment level one - Could you possibly compare the colour features of comments in first position …	3

Appendix II. More examples of *intents* their *examples* and their respective *responses* as developed using the Rasa platform with their corresponding phase of development

Intent	Response	Phase

Example 18. Request to select the filter 'mockery' without specifying the menu and the back-up response/ Back-up dialogue

- intent: filters_mockery examples: | - Can I select [mockery]{"entity": "filter_name"} filtering - May I choose [mockery]{"entity": "filter_name" } filter - Check [mockery]{"entity": "filter_name" } …	utter_filters_mockery: - text: "How do you want to display it?" buttons: - title: "Highlight Colour Feature" payload: '/colour_feature_mockery' - title: "Select Node & Edge" payload: '/highlight_check{"filter_name": "mockery"}'	3

Example 19. Request to deselect the filter 'intolerance' without specifying the menu and the back-up response/ Back-up dialogue

- intent: filters_intolerance_hide examples: | - Clear off [intolerance]{"entity": "filter_name"} filters - May you remove [intolerance]{"entity": "filter_name"} [intolerance]{"entity": "filter_name"} filters - Get rid of [intolerance]{"entity": "filter_name"} features …	utter_filters_intolerance_uncheck: - text: "Where do you want to uncheck it from?" buttons: - title: "Highlight Colour Feature" payload: '/colour_feature_intolerance_uncheck' - title: "Select Node & Edge" payload: '/highlight_uncheck{"filter_name": "intolerance"}'	3

Example 20. Request to select a filter in the node/edge menu without specifying the filter name and the back up response/ Back-up dialogue

- intent: show_node_and_edge examples: | - Can I select node and edge - Choose node and edge - May you tick node and edge - Show me node and edge - I want to see node and edge …	utter_which_node_and_edge: - text: "intent_node_edge_signal;highlight_accordion_button;Please choose a feature you want to highlight"	3

Example 21. Request to define 'Comment Level' and the response/ Definitions

- intent: define_commentlevel
  examples: |
  - Can you tell me what is meant by comment level?
  - I was wondering if you could explain comment level to me?
  - Could you explain to me what the level of the comment refers to?
  - I would like you to explain what comment level is
  - Would you be able to give me a tutorial for comment level?
  - Learn more comment level
…
	utter_comment_level_definition:
	  - text: "intent_signal_level_one_button;chart_sub_button;Comment levels refers to the relationship of the news article with its' comments. \ \n
	
	  Comment level 1 indicates that the comment was posted directly under the news article. \ \n
	
	  Comment level 2 indicates that the comment originates from another comment. \ \n
	
	  To access this information, you can hover over each node. \ \n
	
	  To display statistics for Level One Comments, click the Level One Comments button or ask me. \ \n
	
	  Read more about Comment Level [here](https://dspace5.zcu.cz/handle/11025/45010)"
	  image: "https://ecemkavaz.github.io/imagesRasa/CommentLevel4.png"
		3

Example 22. Request for difference between 'Select Node and Edge' menu and 'Highlight Colour Feature menu' and the response/ Explanation

- intent: help_highlight_select_menus
  examples: |
  - What is the difference between highlight colour features and select node and edge menus?
  - Why there are two menus Highlight colour features and Select node and edge
  - What is the difference between colour feature and node and edge filters?
  - What will be different if I highlight colour features or select nodes and edges?
…
	utter_help_highlight_select_menus:
	  - text: "Highlight colour features shows tagged features different colours in Glyphs \ \n
	
	  Select Node and Edge will show comments with features you have selected \ \n
	
	  After selecting features in the Select Node and Edge menu you can look at Statistics"
		3

Example 23. Request for a reminder about the subtree questions and the response/ Repurposed help

- intent: reminder_high_level_questions
  examples: |
	utter_reminder_high_level_questions_one:
	  - text: "You can ask me basic statistic questions or to visualise
		3

- Could you remind me of the subtree choices - What are the subtree options again? - What are the basic statistic questions you can answer? - Would you possibly remind me of the basic statistics options - What are the questions I can only ask you? - How can I see subtrees? - How can I find out the subtrees choices again? - Is it possible to analyse a smaller part of the visualisation? - Can I analyse a small section of my data? …	subtrees" utter_reminder_high_level_questions_two: - text: "Here are the options:" buttons: - title: "The most tagged features" payload: '/most_tagged_features' - title: "The largest thread" payload: '/largest_thread' - title: "The longest thread" payload: '/longest_thread' - title: "The most toxic thread" payload: '/most_toxic_thread' - title: "The most toxic subtree" payload: '/most_toxic_subtree' - title: "The widest level" payload: '/widest_level' - title: "More help" payload: '/reminder_help'	

Example 24. Rule and dialogue policies associated with the back-up response when an intent is not recognised

Rule and dialogue policies	Response	Phase
- rule: Ask the user to rephrase whenever they send a message with low NLU confidence steps: - intent: nlu_fallback - action: utter_please_rephrase	utter_please_rephrase: - text: "I'm sorry, I didn't quite understand that. Could you rephrase that please?"	3
language: en pipeline: - name: WhitespaceTokenizer - name: RegexFeaturizer - name: LexicalSyntacticFeaturizer - name: CountVectorsFeaturizer - name: CountVectorsFeaturizer analyzer: char_wb min_ngram: 1 max_ngram: 4		

```
    - name: DIETClassifier
      epochs: 100
    - name:
EntitySynonymMapper
    - name:
ResponseSelector
      epochs: 100
    - name:
FallbackClassifier
      threshold: 0.4
      ambiguity_threshold:
0.1

policies:
    - name:
MemoizationPolicy
    - name: TEDPolicy
      max_history: 5
      epochs: 100
    - name: RulePolicy
    - name: FallbackPolicy
      nlu_threshold: 0.5
```

Example 25a. Tutorial of the DVIL interface: automatic question and request for a tutorial intent (Phase 3)

```
if tracker.get_slot("sessionid") and tracker.get_slot("sessionid") !=
"None":
    if nickname_value:
        messages = ["Hey, " + nickname_value + ". Nice to see you! 😄",
                    "Hi, " + nickname_value + ". Welcome back!  😄",
                    "Hello, " + nickname_value + ". I'm here if you need
me 😄"] if not greet_again_value else \
            ["Hey, " + nickname_value + ". How can I help you? 😄",
             "Hi, " + nickname_value + ". What can I do for you? 😄"]
        dispatcher.utter_message(text=random.choice(messages))

        # if first_login_value or greet_again_value:
        dispatcher.utter_message(text="Are you new? Would you like a
tutorial?",
                                 buttons=[
                                         {"title": "Yes", "payload":
"/ask_tutorial"},
                                         {"title": "No", "payload":
"/tutorial_do_not_want"}
                                     ])
```

```
- intent: ask_tutorial
  examples: |
    - I want a tutorial
    - I need a tutorial
    - I was wondering if you could explain everything again
    - Give me a tutorial on everything
    - I was hoping you could tell me everything you can do again
…
```

Example 25b. Respective responses to tutorial (Phase 3).

Yes using automatic question button option or response to the intent	No using automatic button option

```
utter_tutorial_mode:
  - text: "Welcome to the tutorial, please don't interrupt! \ \n

   This tool helps you:

             - Visualise features \ \n
             - Carry out statistical analysis \ \n
             - Answer basic statistics questions and visualise subtrees \ \n
             - Ask for help"
    buttons:
      - title: "Next"
        payload: '/tutorial_layouts'

utter_tutorial_mode_layouts:
  - text: "intent_layouts_signal;layouts-radio-buttons;I automatically select the layout that best fits your data. \ \n

      Change this:

    - Using the menu \ \n
    - Asking me"

    buttons:
      - title: "Next"
        payload: '/tutorial_glyphs'

utter_tutorial_mode_glyphs:
  - text: "intent_glyphs_blinking_tutorial;Glyphs are designs that show you features as a colour. \ \n

 To activate/deactivate the features:

  - Use the Highlight Colour features menu \ \n
  - Ask me \ \n

  I've shown you an example, zoom in to take a look"
    buttons:
      - title: "Next"
        payload: '/tutorial_targets'
```

```
utter_tutorial_do_not_want:
  - text: "Okay, remember I can: \ \n

                            - Change visualisation options / Select filters \ \n
                            - Show you subtrees and answer basic statistics questions \ \n
                            - Offer you help or a tutorial  \ \n

  P.S. Think about what I have told you I can do, and ask questions related to these things. That way, I can help you carry out your analysis step-by-step 😄 "
```

```
utter_tutorial_mode_targets:
  - text:
"intent_targets_signal;targets_acco
rdion_button;All comments are
tagged with the targets that
they're directed at:

                        - Target
group \ \n
                        - Target
person \ \n
                        - None \ \n


Activate/Deactivate targets: \ \n

                        - Using the
Targets menu \ \n
                        - Using the
Select Node and Edge menu \ \n
                        - Asking me
\ \n

                        You can also
look at the breakdown of targets
with Statistics. "
    buttons:
      - title: "Next"
        payload:
'/tutorial_node_and_edge'

utter_tutorial_mode_select_node_edg
e:
  - text:
"intent_node_edge_signal_tutorial;S
elect Node and Edge will select
comments with features you've
selected \ \n

  To activate/deactivate the
features \ \n
  - Use the Select node and edge
menu \ \n
  - Ask me \ \n

  I've shown you an example, zoom
in to take a look"


    buttons:
      - title: "Next"
        payload:
'/tutorial_statistics_introduction'


utter_tutorial_mode_statistics_intr
oduction:
  - text:
"intent_statistics_signal_tutorial_
intro;After you have chosen the
```

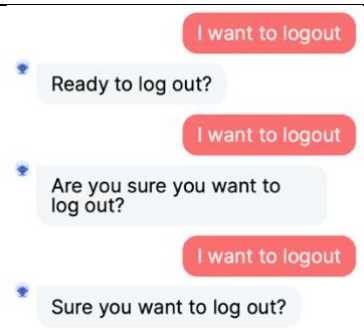
```
features you want to analyse, you
can look at Statistics. \ \n

  Statistics summary gives you a
breakdown of toxicity and targets
in the current visualisation. When
you select filters, my answer
changes"
    buttons:
      - title: "Next"
        payload:
'/tutorial_statistics'

utter_tutorial_mode_statistics:

  - text:
"intent_tutorial_statistics_charts_
signal;statistics_all_features_subt
rees;Selected Comments Statistics
shows you overall numbers of
features for comments selected
using the Select Node and Edge
menu. \ \n

            Level One Comments
Statistics allows you to analyse
the features of comments at level
one and the comments in their
threads. \ \n

             You can change the
variables and graph type using the
two drop-down menus. \ \n
             When you click
variables on your graph, they will
also be activated on the main
interface"

    buttons:
      - title: "Next"
        payload:
'/tutorial_most_tagged_features'

utter_tutorial_mode_most_tagged_fea
tures:
  - text: "You can ask me to answer
basic statistics questions like: \
\n
               - What is the most
tagged feature? \ \n

               My answer will also
change depending on your current
visualisation"
    buttons:
      - title: "Next"
        payload:
'/tutorial_high_level'

utter_tutorial_mode_high_level:
  - text: "To see subtrees, ask me
```

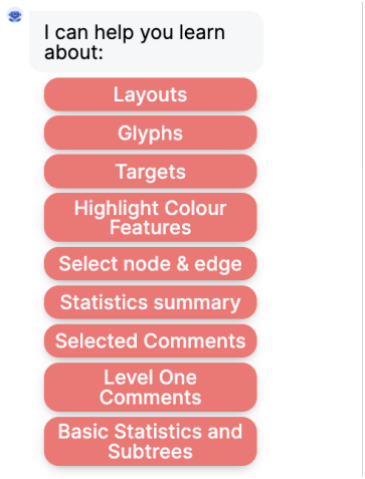
```
to show you:

               - the largest thread
\ \n
               - the longest thread
\ \n
               - the most toxic
thread \ \n
               - the most toxic
subtree \ \n
               - the widest level "
    buttons:
      - title: "Next"
        payload:
'/tutorial_subtrees'

utter_tutorial_mode_subtrees:
  - text: "intent_subtrees_signal;I
can also highlight features in
subtrees  \ \n

               To save them, use
the button in the pop-up window \
\n "

    buttons:
      - title: "Next"
        payload:
'/tutorial_help_choices'

utter_tutorial_help_choices:
  - text: "The help I can offer is:
\ \n

      - Tutorial \ \n
      - Definitions of features \
\n
      - Explanations of interface
options \ \n
      - 'Fast help'"

    buttons:
    - title: "Next"
      payload:
'/tutorial_change_file'

utter_tutorial_mode_change_file:
  - text:
"intent_change_file_signal;You can
change files using the drop-down
menu "

    buttons:
      - title: "Next"
        payload:
'/tutorial_chatbot'

utter_tutorial_mode_chatbot:
  - text: "And the final part of
this interface is me! \ \n
```

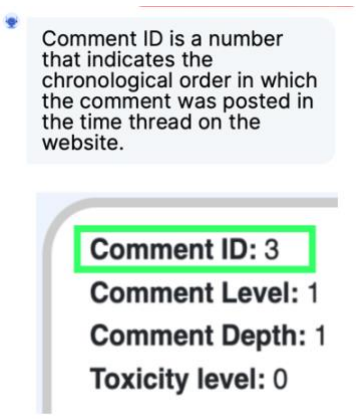
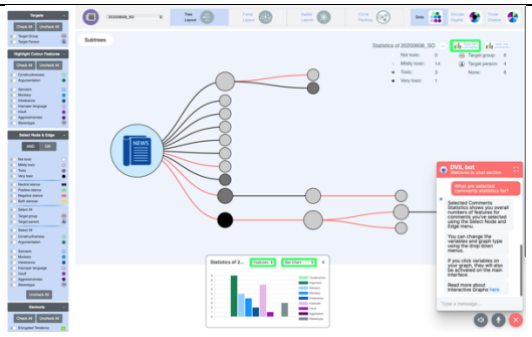
```
                I'm here to: \ \n
                - Change
visualisation options / Select
filters \ \n
                - Show you subtrees
and answer basic statistics
questions \ \n
                - Offer you help  \
\n

                 Use this window or
your microphone to talk to me 😄 \
\n

                 P.S. Think about
what I have told you I can do, and
ask questions related to these
things. That way, I can help you
carry out your analysis step-by-
step 😄 "


utter_tutorial_do_not_want:
  - text: "Okay, remember I can: \
\n

                           - Change
visualisation options / Select
filters \ \n
                           - Show
you subtrees and answer basic
statistics questions \ \n
                           - Offer
you help or a tutorial  \ \n

  P.S. Think about what I have told
you I can do, and ask questions
related to these things. That way,
I can help you carry out your
analysis step-by-step 😄 "
```

Appendix III. Different types of reponses given by the chatbot and their development using the Rasa platform

Response type	Presentation of the response on the DVIL interface	Configuration settings used to develop the responses in the domain.yml file
Text only		`utter_colour_feature_sarcasm:` `  - text: "intent_show_colour_features;sarcasm;I've checked the sarcasm colour feature"`

With variations		`utter_logout_confirmation:`
		`- text: "Sure you want to log out?"`
		`- text: "Are you sure you want to log out?"`
		`- text: "Are you sure you're ready to log out?"`
		`- text: "Ready to log out?"`
		`- text: "Are you ready to log out?"`

With buttons		`utter_reminder_help:`
		`  - text: "I can help you learn about: "`
		`  buttons:`
		`  - title: "Layouts"`
		`  payload: '/help_layouts'`
		`  - title: "Glyphs"`
		`  payload: '/help_glyphs'`
		`  - title: "Targets"`
		`  payload: '/targets_help'`
		`  - title: "Highlight Colour Features"`
		`  payload: '/help_colour_features'`
		`  - title: "Select node & edge"`
		`  payload: "help_select_node_edge"`
		`  - title: "Statistics summary"`
		`  payload: '/help_statistics_summary'`
		`  - title: "Selected Comments"`
		`  payload: '/help_main_graph_barchart'`
		`  - title: "Level One Comments"`
		`  payload: '/help_subtree_barchart'`
		`  - title: "Basic`

		Statistics and Subtrees" payload: '/reminder_high_level_questions'
With images		utter_comment_id_definition: - text: "Comment ID is a number that indicates the chronological order in which the comment was posted in the time thread on the website. \ \n You can see this information by hovering over nodes \ \n" image: "https://ecemkavaz.github.io/imagesRasa/CommentID5.png"
With visual enhancements and links		utter_help_main_bar_one: - text: "intent_signal_additional_graphs;chart_main_button;statistics_all_features_tree;Selected Comments Statistics shows you overall numbers of features for comments you've selected using the Select Node and Edge menu. \ \n You can change the variables and graph type using the drop down menus. \ \n If you click variables on your graph, they will also be activated on the main interface \ \n Read more about Interactive Graphs [here](https://dspace5.zcu.cz/handle/11025/45010)" utter_help_bar_chart_pop_up: - text: "If you want more information about other types of statistics:" buttons: - title: "Click here" payload: '/statistics_help'

Appendix IV. Interfaces of the DVIL resource. Changes to these interfaces involved changing spellings and keywords.

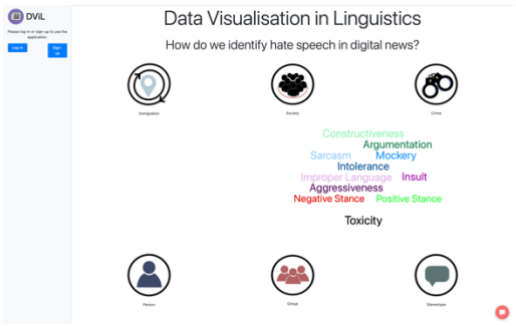


Figure 1. Initial interface.

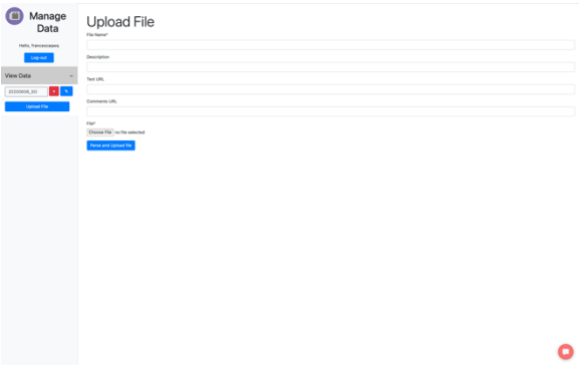


Figure 2. Uploading interface

Appendix V. Chatbot Exploratory Study questionnaire

Exploratory Chatbot Study

This study is designed to observe the interactions between the user and chatbot on the DVIL interface.

The **NewsCom-TOX** corpus was developed to detect toxic messages posted in the conversation threads in comment section of digital news articles. The Data Visualisation In Linguistics (DVIL) project aims to incorporate visualisation tools to help linguists analyse their corpora annotated in the NewsCom-TOX corpus.
This questionnaire consists of 3 tasks. To complete the first 2 tasks you must interact with the DVIL chatbot and answer the questions. The final task invites you to evaluate your experience with the DVIL interface and the DVIL chatbot.

We will use the collected data to evaluate visualisations and chatbot interaction and include results in studies.

The expected duration of this questionnaire is 30 minutes.

* Indicates required question

CONSENTIMIENTO INFORMADO DEL PARTICIPANTE

Título del proyecto de investigación: Data Visualisation for Linguistics

Investigadoras principales: Ecem Kavaz & Francesca Wright
El participante tiene que leer y contestar las frases siguientes con atención:

- He leído toda la información que le ha sido facilitada sobre este proyecto.
- He tenido la oportunidad de preguntar y comentar cuestiones sobre el proyecto.
- He recibido la suficiente información sobre este proyecto.
- He recibido respuestas satisfactorias a todas las preguntas.
- He comprendido que durante mi participación en este estudio, se realizará la grabación de la pantalla del portátil, se registrará mi voz y se realizarán observaciones relacionadas con mis reacciones al llevar a cabo las tareas.
- He comprendido que es libre de abandonar este proyecto sin que esta decisión pueda ocasionarme ningún perjuicio, en cualquier momento y sin dar ninguna razón.
- He comprendido que los resultados obtenidos a través de mi participación en este estudio serán anónimos.

En el caso que más adelante usted quiera hacer alguna pregunta o comentario sobre este proyecto, o bien si quiere revocar su participación en el mismo, por favor contacte con: Ecem Kavaz & Francesca Wright

Facultad de Matemáticas e Informática y Facultad de Filología y Comunicación

E-mail de contacto: ekavazka27@alumnes.ub.edu & frwrighw11@alumnes.ub.edu

1. Estás de acuerdo en participar? Sí/No *

Information and Demographics

2. Gender *

Mark only one oval.

- ◯ Male
- ◯ Female
- ◯ Other
- ◯ Prefer not to say

3. Age *

Mark only one oval.

- ◯ [18-30]
- ◯ [31-40]
- ◯ [41-50]
- ◯ [50+]

4. First Language

Mark only one oval.

- ◯ Catalan
- ◯ Catalan/Spanish
- ◯ Spanish
- ◯ English
- ◯ Other

5. Do you have any experience in corpus annotation? *

Mark only one oval.

- ◯ Yes
- ◯ No

6. Do you have any experience in data visualisation? *

Mark only one oval.

- ◯ Yes
- ◯ No

Sign up
Login in to the DVIL interface using the username and password we have given you.

7. Choose the username used to access the DVIL interface *

Mark only one oval.

- ◯ user1
- ◯ user2
- ◯ user3
- ◯ user4
- ◯ user5
- ◯ user6
- ◯ user7
- ◯ user8
- ◯ user9
- ◯ user10

Training Time with WIMP
Now you have some time to train.
Choose training_dataset from the drop down-menu. Take 2 minutes to play with the platform to becom
familiar with the tool. **Please do not interact with the chatbot.**

Let's practice selecting some filters.
In the positive stance, how many comments are directed at a target group?
In the negative stance, how many comments are directed at a target person?

Now the training is over. Clear any filters and reset to **Tree Layout** and **Dots.**

Instructions

You will work through 2 tasks. Each task is introduced with a hypothesis and the questions to follow aim to draw conclusions about the hypothesis stated.

Instructions:

Use the chatbot to carry out all questions in tasks 1 and 2 unless specified.

Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

Task 1a

Step 1: Select the file task_1_dataset **using the drop-down menu**

Step 2: Look at all the layout options and decide which is best for your analysis.

8. The chatbot automatically selected Radial layout for this data. Did you stay in this layout or change it after looking at all of them? Please select the final layout used.

   Mark only one oval.

   - ◯ Tree Layout
   - ◯ Force Layout
   - ◯ Radial Layout
   - ◯ Circle Packing

9. Why did you choose this layout?

Read the following hypothesis:
“The presence of argumentation in general implies constructiveness.

*Therefore, we are interested in checking **the relationship between** the categories **Argumentation** and **Constructiveness.**”*

Translation:

"La presencia de argumentación en general implica constructividad.

*Por tanto, nos interesa comprobar **la relación entre** las categorías **Argumentación** y **Constructividad**."*

Now open the chatbot

10. **Ask the chatbot** any questions in order to solve the question: In how many comments do we see **Argumentation** and **Constructiveness** together? Write a number.

    Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

Ask the chatbot any questions to clear all filters. Close any pop-up windows with the mouse.

Read the following hypothesis
*“Constructive comments tend to be less toxic than non-constructive ones, although constructive comme can possibly be found with a low degree of toxicity **(Toxicity level = 1)**.”*

Translation:

*"Los comentarios constructivos tienden a ser menos tóxicos que los no constructivos, aunque posiblemente se pueden encontrar comentarios constructivos con un grado de toxicidad bajo **(Toxicity level=1)**."*

11. **Ask the chatbot** any questions in order to solve the following question: What is the most common level of toxicity for comments tagged with the feature ***Constructiveness****?*

    Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

    Mark only one oval.

    - ◯ Not toxic (Level 0 Toxicity)
    - ◯ Mildly toxic (Level 1 Toxicity)
    - ◯ Toxic (Level 2 Toxicity)
    - ◯ Very Toxic (Level 3 Toxicity)

12. **Ask the chatbot** any questions to answer the following: Are there other **features** that appear for **this level of toxicity**? Select all that are relevant

    Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

    Check all that apply.

    - ☐ Constructiveness
    - ☐ Argumentation
    - ☐ Sarcasm
    - ☐ Mockery
    - ☐ Intolerance
    - ☐ Improper Language
    - ☐ Insult
    - ☐ Aggressiveness
    - ☐ Stereotype

Ask the chatbot any questions to clear all filters. Close any pop-up windows with the mouse.

Task 2a

Step 1: Select file task2_dataset from drop-down menu

Step 2: Look at all the layout options and decide which is best for your analysis.

13. The chatbot automatically selected Force layout for this data. Did you stay in this layout or change it after looking at all of them? Please select the final layout used.

Mark only one oval.

- Tree Layout
- Force Layout
- Radial Layout
- Circle Packing

14. Why did you choose this layout?

Read the following hypothesis

"The presence of negative stereotypes (stereotyped prejudices) can lead to negative judgments or attitudes, of intolerance that can end in discrimination, marginalisation and exclusion. That is, you can determine that the content of the comment is more toxic.Therefore, we are interested in: a) check ***the relationship between*** *the* ***Stereotype*** *and* ***Toxicity level*** *categories b) check* ***the relationship between*** *the* ***Stereotype*** *and* ***Intolerance*** *categories"*

Translation:

"La presencia de estereotipos negativos (prejuicios estereotipados) puede llevar a emitir juicios o actitud negativas, de intolerancia que pueden terminar en discriminación, marginación y exclusión. Es decir, pue determinar que el contenido del comentario sea más tóxico."

15. **Ask the chatbot** any questions in order to solve the following question: What is the most common **level of toxicity** for the feature **Stereotype**?

    Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

    Mark only one oval.

    - ◯ Not toxic
    - ◯ Mildly toxic
    - ◯ Toxic
    - ◯ Very Toxic

Ask the chatbot any questions to clear all filters. Close any pop-up windows with the mouse.

Task 2b

Step 1: To carry out this task you need to view **the most toxic subtree**. **Ask the chatbot.**

Step 2: Look at all the layout options in the pop-up window and decide which is best for your analysis.

16. The chatbot automatically selected Tree layout in the subtree pop-up window. Did you stay in this layout or change it after looking at all of them? Please select the final layout used.

    Mark only one oval.

    - ◯ Tree Layout
    - ◯ Force Layout
    - ◯ Radial Layout
    - ◯ Circle Packing

17. Why did you choose this layout?

18. **Now you will look at the subtree.**
Ask the chatbot any questions in order to solve the following question: What is the most common **level of toxicity** for the feature **Intolerance** in **the most toxic subtree**?

Remember, you are also free to ask the chatbot for **a tutorial, definitions, explanations** and **fast help** and to **change layouts** and **glyph types** at any point in order to help you carry out the analysis.

Hint: You must make sure there is a blue line around the pop-window if you want to interac with the subtree.

Mark only one oval.

- ◯ Not toxic
- ◯ Mildly toxic
- ◯ Toxic
- ◯ Very Toxic

Data collection

Step 1: Tell the chatbot "Analysis done" to download your data

Evaluation of the DVIL chatbot

Here we ask you to give us your opinion on the DVIL chatbot

19. I was immediately made aware of what information the chatbot can give me *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

20. The chatbot gives me the appropriate amount of information *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

21. The chatbot's responses were easy to understand *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

22. I feel like the chatbot's responses were accurate *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

23. I find that the chatbot understands what I want and helps me achieve my goal. *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

24. The interaction with the chatbot felt like an ongoing conversation *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

25. The chatbot was able to keep track of context *

Mark only one oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly Agree

26. Was it easy for you to answer the questions in both tasks? Please explain. *

27. Did the chatbot help you to understand visualisations? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very useful

28. Please write any comments about the previous question (Did the chatbot help you to understand visualisations?) Please write your answer in English or Spanish.

29. In Task 2, we asked you to use the chatbot to look at **the most toxic subtree**. There are als some other options you can ask the chatbot to show you:

- the most tagged feature
- the largest thread
- the longest thread
- the most toxic thread
- the widest level

What else would you like to be able to ask? Write your questions below.
Please write your answer in English or Spanish

30. Please write any other comments that you want to tell us *(Feedback, Problems, Ideas...).* Please write your answer in English or Spanish.