Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1788–1797
Marseille, 20-25 June 2022

EZCAT: an Easy Conversation Annotation Tool

Gaël Guibon[a,b], Luce Lefeuvre[b], Matthieu Labeau[a], Chloé Clavel[a]
[a] LTCI, Télécom-Paris, Institut Polytechnique de Paris, [b] Direction Technologies, Innovation & Projets Groupe, SNCF
19 place marguerite perey, 91120 Palaiseau; 1/3 Av. François Mitterrand, 93210 Saint-Denis
{gael.guibon, matthieu.labeau, chloe.clavel}@telecom-paris.fr
{luce.lefeuvre, ext.gael.guibon}@sncf.fr

Abstract

Users generate content constantly, leading to new data requiring annotation. Among this data, textual conversations are created every day and come with some specificities: they are mostly private through instant messaging applications, requiring the conversational context to be labeled. These specificities led to several annotation tools dedicated to conversation, and mostly dedicated to dialogue tasks, requiring complex annotation schemata, not always customizable and not taking into account conversation-level labels. In this paper, we present EZCAT, an easy-to-use interface to annotate conversations in a two-level configurable schema, leveraging message-level labels and conversation-level labels at once. Our interface is characterized by the voluntary absence of a server and accounts management, enhancing its availability to anyone, and the control over data, which is crucial to confidential conversations. We also present our first usage of EZCAT along with our annotation schema we used to annotate confidential customer service conversations. EZCAT is freely available at `https://gguibon.github.io/ezcat`.

Keywords: conversations, annotation tool, text messages

1. Introduction

In the recent years, text have been one of the main user generated content modality, whether it is a comment, a blog, or chats. Among these, textual conversations led to many studies on dialogue (Zhu et al., 2019; Ma et al., 2020; Colombo et al., 2020; Deriu et al., 2021; Razumovskaia et al., 2021), which could be multi-modal (Nie et al., 2019; Chapuis et al., 2020) or specialized on a modality such as text. These studies have been possible thanks to the rise of available public corpora dedicated to dialogue (Jurafsky et al., 1997; Shriberg et al., 1998; Stolcke et al., 2000; Novikova et al., 2017; Li et al., 2017). However, this improvement only concerns a part of the User Generated Content (UGC), and private dyadic conversations are yet to be found. These conversations usually come from messaging applications such as WhatsApp, Telegram, or WeChat, to name but a few. Due to privacy concerns, sharing them is not easy; but many dialogue systems would benefit from performance reports obtained on real data. This requires annotated data sets, and therefore tools for conversation annotation. We propose a quick, easy-to-use, and freely accessible annotation tool dedicated to spontaneous conversations.

This tool aims at reducing annotation costs and to enabling short annotation tasks with simple custom annotation schemata for anyone. It is designed to quickly annotate private conversation data in order, for instance, to have another point of view of the performance of a model on a real-data test set. We follow the global tendency in Natural Language Processing (NLP) which consists in developing applications dedicated to specific tasks by opposition with more generic annotation tools, for instance GATE (Cunningham et al., 2002) or Glozz (Widlöcher and Mathet, 2012). This tendency led to applications such as a web-based language learning tool using syntactic and morphological parsing (Nagata, 2002), NLP services for desktop clients (Witte and Gitzinger, 2008), or semantic interactive annotation tools (Klie et al., 2018). Moreover, in the scope of annotation tools dedicated to dialogues, they often incorporate more specific concepts of dialogues such as adjacency pairs which, although they are interesting to study, for instance for question-answering, are not always mandatory to start annotating small corpora for other purposes. Plus, they may confuse users. Any additional concept to the interface can become an obstacle keeping the non-linguist neophyte from starting a simple annotation of his personal data. Because we want to ease the access to annotation and render quick tests possible for anyone, we stand out from the trend of drifting towards more complex applications, and do not underestimate the value of small applications that are completely and freely accessible.

In this paper, we present EZCAT, an annotation tool dedicated to annotate textual conversations which is freely accessible, requires no installation, and comes with configurable annotation schemata. EZCAT is made with the assumption that the dialogue structure follows the conversation structure. The advantages of EZCAT are the following:

1. Data confidentiality. No data is stored anywhere else than the user's desktop, even though it is a web application. JSON files are to be loaded or downloaded for both data and configuration but are not uploaded to any server. Indeed, no connection is required, the application does not access any external API and can be run from static files.

The configuration can be stored in the browser local storage to ease resuming the annotation process.

2. Easy usage and access. EZCAT does not require any account creation and thus is freely accessible to anyone. Moreover, no installation is needed and only a browser is used to run the application, at the very least. Optional installers for desktop or mobile devices are provided.

We used this application to annotate a corpus of private conversations. Thus, in addition to the tool, we present the annotation schema we used along with the first annotators' feedback. We put the annotation schema as one of the default available configurations for EZCAT.

2. Related Work

It is appropriate to compare EZCAT with other existing related tools. Table 1 displays the main differences between EZCAT and other annotation tools for dialogues or conversations.

Conversation Annotation Tools. DialogueView (Heeman et al., 2002; Yang and Heeman, 2005) mainly considers the audio signal and integrates 3 views in the user interface: the word view (audio time aligned), the utterance view, and the block view (for discourse blocks). It is written in Tcl/Tk and represents annotations as XML files, as it works locally. DialogueView is not suitable for annotating text conversations with two-level annotations such as EZCAT.

A web-based application for annotating sentiment in dialogue (Langlet et al., 2017) focuses on verbal expression of sentiments, verbal content and conversation structure, as it only allows access to manually transcribed conversations. This kind of application is suitable for human-agent conversations but does not consider the two-level annotation and requires a server access or installation using PHP. EZCAT makes up for these limitations while focusing on textual data.

CAMS (Duran and Battle, 2018) is another dialogue annotation tool but made for a specific annotation campaign, as it was initially the case for EZCAT. However, EZCAT goes beyond its original campaign by allowing configurations. Moreover, CAMS comes with downsides as it requires a Python Flask server and is not easily modifiable (some programming is needed).

LIDA (Collins et al., 2019) is a dialogue annotation tool which integrates recommendations from dialogue systems and enable inter-annotator agreement inside the application. It is made using a Flask REST API and thus, requires a server to run. While the application is interactive and allows message-level annotation, it does not allow conversation-level annotation. An evolution of LIDA, named MATILDA (Cucurnia et al., 2021), has been recently released. It adds a multi-language, multi-annotator annotation support but still requires a server, and is not hosted anywhere, hence an installation or a docker usage is required.

Non-specific tools. GATE (Cunningham et al., 2002) is a Java-based tool which allows easy enhancement and tries to allow a large set of different annotations. It comes with the downside of a difficult and time-costly setup to start the annotation process. Among the recent annotation tools, INCEpTion (Klie et al., 2018) is dedicated to data involving spans and relations between spans, which makes it useful for semantic annotations by considering semantic resources such as knowledge bases. It is an evolution of WebAnno (Eckart de Castilho et al., 2016) without the syntactic representations. Its main advantage is the label recommendation system integrated into the application to learn from the annotation habits of the user. However, it does not consider conversation-level annotations. Arborator-Grew (Guibon et al., 2020) is an interactive annotation tool dedicated to treebanks curation through grammar rules. It comes with the downside of either setting up an account or installing the different modules locally using multiple languages. Along with Arborator-Grew, a freely accessible version has been made to better suit quick and easy CONLL curation without the server requests. This version is named Arborator-Draft[1]. Our will to share EZCAT as a freely accessible application without any installation nor account stems from the usage feedback and observation with regard to Arborator-Draft versus the Arborator-Grew full version.

These tools do not allow to assign a label to the whole conversation considering the previous message annotation phase, which makes the two-level annotation unavailable to annotators. It is always required to either go through a lengthy installation process or send data to a server which may be impossible due to confidentiality issues. Also, none of them consider the need to annotate conversations from common instant messaging applications. To fill the gap created by those missing features, we share EZCAT, a freely accessible application, which allows message-level and conversation-level annotations, and does not require any installation or server access.

3. Software Architecture

EZCAT is fully made using Javascript and Vue.js[2], a progressive Javascript framework for single page applications. We chose this technology in order to render the application usable for any device. Also, it easily allows multiple transformations and extensions by modifying the code. Here are some examples:

- Easy integration of external API
- Compile EZCAT into native desktop and mobile applications using Electron[3]

[1] https://arborator.github.io/draft/live.html

[2] https://vuejs.org/

[3] https://www.electronjs.org/

Annotation Tool	Recommenders	Inter-Annotator Resolution	Conversation-level Label	Server /Login	Required Installation	Language
EZCAT	No	No	Yes	No	No	Javascript
MATILDA	Yes	Yes	No	Yes	Yes	Python
LIDA	Yes	Yes	No	Yes	Yes	Python
INCEpTion	Yes	Yes/No	No	Yes	Yes	Java
GATE	No	Yes/No	Yes (difficult)	Yes	Yes	Java
TWIST	No	No	No	No	Yes	-
BRAT	Yes	No	No	No	Yes	Python
DOCCANO	No	No	No	-	Yes	Python
DialogueView	No	No	No	-	Yes	TcK/TK

Table 1: Overview of the differences between EZCAT and other applications.

- Easy internationalization by adding languages using `VueI18n`[4]

EZCAT represents the "view" part of a traditional MVC (model-view-controller) software architecture, made independent by externalizing the data handling to the user. Thus, data are stored on the user hard drive disk only. To handle the pseudo-model part, we use a state management pattern[5] in which we store the current annotation schema and the temporary data. All these information are only stored locally into the RAM used by the browser. On the downside, this implies a limited computational power, depending on the browser capacity and its Javascript engine. This is why we would have to resort to an external API or a server if for instance, we wanted to add smart label recommendation, as this is a web-based client application (such as Microsoft Teams[6] or Discord[7]). Another limitation is the user history and tracking, which cannot be integrated due to the absence of user profiles. This hinders the application's usage tracking, but yields the benefit of real accessibility of the application. Finally, the design we chose is not suitable for large scale annotation campaigns, especially with a large number of annotators, but can still serve for lower scale campaigns, which cover most of annotations of confidential data (see following Section 5).

Due to the tool being deliberately simple, the software architecture is straightforward. In spite of this, we use a component-based approach, which will allow more advanced modifications in the future.

4. Main Features

EZCAT possesses several features that distinguish it from other related applications. These features are either inherent to the software architecture and design choices, or stem from the targeted goal of the application.

Instant Messaging Conversation Imports. In EZCAT, users can import their own private conversations from some of the most popular instant messaging applications. For instance, users can export their conversations from WhatsApp[8] and load them into the application without any concern about security issues as EZCAT does not send any data. Then, users can start annotating messages and conversations by selecting a default annotation schema or configuring another one. At the moment, EZCAT allows WhatsApp and Telegram[9] imports, as visible in Figure 1.

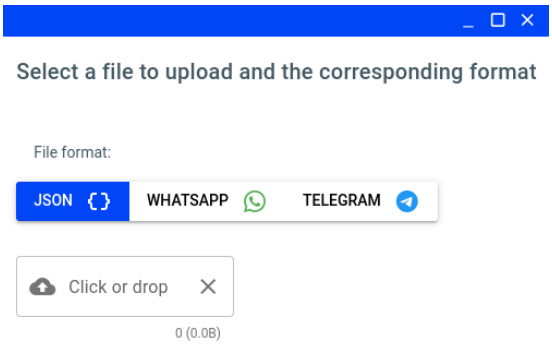


Figure 1: Example of the component for instant messaging import. Imports can come from our custom JSON format, the text file from exported WhatAapp conversations, or the HTML file from exported Telegram ones.

A two-level annotation process with a portable and configurable annotation schema. Annotations schemata are represented into a JSON file and can embed the description of each label, to be shown as tooltips. This means it consists in a dictionary with two main types of labels: message-level labels and conversation-level labels. It is possible to add as many labels as desired with different types of representations, raw values, selection from a list of values, booleans, or range radio button (for Likert scale (Likert, 1932) for instance). In EZCAT, a conversation annotation is constituted ofa couple of steps visible in Figure 4, which

[4] `https://kazupon.github.io/vue-i18n/`
[5] `https://vuex.vuejs.org/`
[6] `https://www.microsoft.com/en-us/microsoft-teams/group-chat-software`
[7] `https://discord.com/`
[8] `https://www.whatsapp.com/`
[9] `https://telegram.org/`

can be considered as mandatory in order to advance to the following one. For instance, we would usually prefer to enforce that all the messages are labeled first, before being able to label the whole conversation. This is why in the configuration, each element can be set as mandatory or not. We also allow the user to configure if all elements should be annotated in order for the conversation to be saved and considered as labeled. The configuration file for the annotation schema can be exported and/or set as a default one to load from the browser's local storage. Figure 2 shows the interface for modifying the annotation schema.

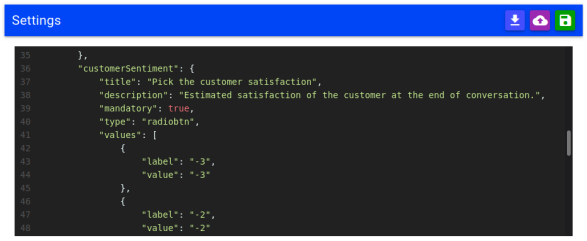


Figure 2: Annotation schema configuration page. The annotation schema is a JSON object with some customizable user interface indications such as the type ("radiobtn" for radio buttons selection in this example).

No user account. By opposition with other conversation annotation tools, we made EZCAT user-free, which comes with pros and cons, mainly advantaging the accessibility over the interactivity (see Section 3). If many annotators need to interact and if the data privacy policy allows it, more complex tools such as MATILDA (Cucurnia et al., 2021) can be used. However, the deliberate choice to make EZCAT user-free makes it suitable to quickly test annotation schemata, annotate private data from a couple of annotators independently, or simply curate already annotated corpora and browse them.

Resuming work. Annotation time, and therefore the annotation efficiency, is one of the main purposes of EZCAT, as such it allows to go directly to the last annotated conversation in order to continue the annotation process for fast resuming. Also, the conversation list possesses two modes: one that displays every conversations as a infinite scroll list, and another one that focuses only on the non annotated conversations. Moreover, at any time, on top of the conversation list, a progress bar indicates the annotation progress for the whole set of conversations, as visible in Figure 3.

Available anywhere. EZCAT is available anywhere, either from the hosted URL `https://gguibon.github.io/ezcat` or by running the compiled HTML file in a browser from the current release. Moreover, optional desktop installers, along with mobile ones (Android and iPhone), are available without any difference at use time. This is one of the benefits from the chosen architecture (Section 3).

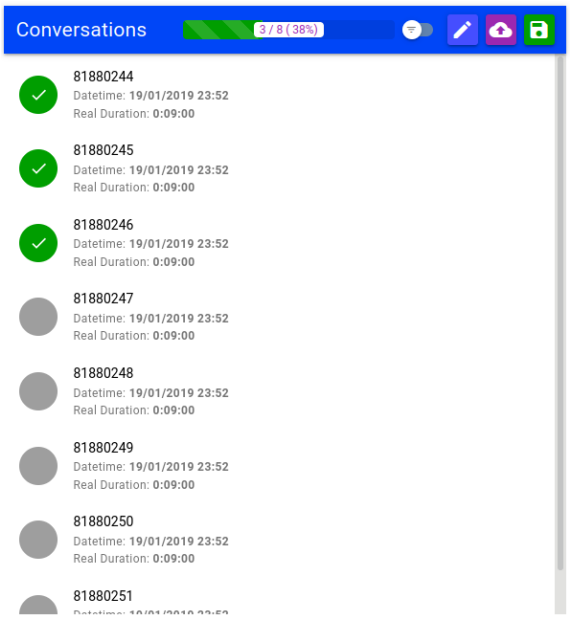


Figure 3: Example of the corpus view with a list of all the conversations, the difference between tagged and not tagged conversations, and the toggle button to only gather conversations to annotate.

5. Annotation Schema

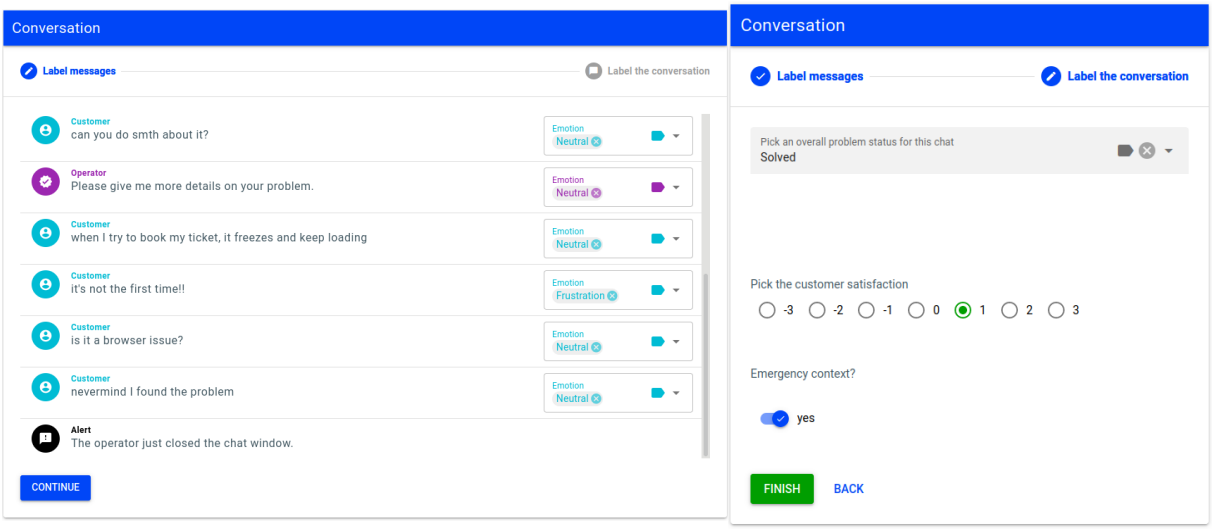
We first encountered the need to annotate confidential data made of dyadic textual conversations from customer service, this led to the creation of EZCAT. Indeed, we desired an application without any server, that focuses on the essentials part of this specific annotation task in order not to disturb the two annotators. By using EZCAT to annotate we did not have to worry over security issues or specific accounts. Even though we cannot share the result of the annotation campaign due to confidentiality limitations, we share the annotation schema we made for this task. This is one of the default annotation schemata available in EZCAT.

5.1. Message-level Labels

As shown in Figure 5, we consider emotions as a message-level label, in a mono-label approach. We consider the emotion to be labeled as the prevalent one: the most important emotion the message conveys. However, emotion label assignment takes into account the previous messages, the conversation context. In our annotation schema we consider 10 labels that differ from the standard emotions used while annotating message's emotion (Novielli et al., 2018) and are more precised than polarities (Chowdhury et al., 2016). We consider 9 emotion labels (3 positive ones, 4 negative ones, and one ambivalent one) with the additional neutral label, for a total of 10 emotions.

Neutral. The neutral label is used when no specific emotion is conveyed by the message. This is, by far, the most frequent label in a conversation data set.

No emotion. We consider this label as the total absence



(a) First step: message-level annotation

(b) Second step: conversation-level annotation

Figure 4: The two-step process for annotating conversations in EZCAT.

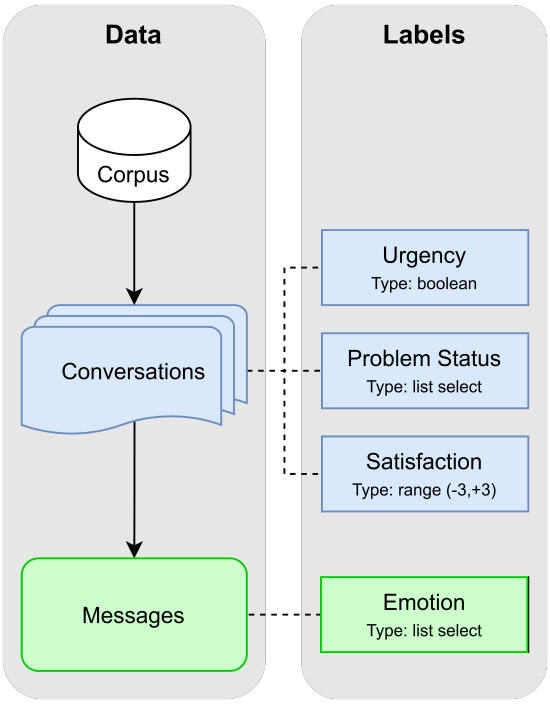


Figure 5: Annotation schema used for our annotation campaign while developing EZCAT. The type fields refer to the configurable options in to modify the user interface accordingly.

of possible emotion. Indeed, in our customer service data set, some alerts are automatically prompted for specific actions such as “user x left the chat” or “operator sent a link”. We call these “alerts”, and they are labeled as “no_emotion”. To differentiate it with the ”Neutral” label, the latter means that the emotional content of the message, written by a human, has been considered as neutral by the annotator, while this label only concerns automatic behaviors.

For the positive leaning labels we have the following:

Amusement. This label represents the amusement in a boarder scope including light-hearted feelings. We chose this label instead of the commonly used ”Joy” label due to the specificity of our data context: customer services do not target joy as an emotion but rather a global amusement from the customer.

Satisfaction. The satisfaction is both an emotion and a feeling, and compared to (Chowdhury et al., 2016), we consider the satisfaction at the conversation level. In the context of customer service, this label represents the objective of the conversation, to better satisfy the customer (Danesi and Clavel, 2010). However, this label is complex and can represent the satisfaction towards different elements depending of the message’s content and the conversational context. For instance, a message such as ”Great, you are here!” after the first message from the operator would be labeled as satisfaction, even if in this case, the satisfaction comes from the announced start of the conversation. Depending on the context this same content could be as relief.

Relief. The customer can express relief from multiple events, such as a reduction fee or a refund being applied to his order. Unlike the previous ”satisfaction” label, this label is as is. It represents the most straightforward definition of relief, without any ambiguity with other labels.

We distinguish 5 negative emotions as follows:

Fear. We use the label fear to encapsulate multiple emotions at once: fear, anxiety, dread, worry, apprehension and stress. All these notions are represented in this ”fear” label. We chose to merge these notions in order to simplify the annotation process, and to not dive into their subtle differences.

Sadness. From customer service conversations, the customer can feel sadness from a specific situation. The sadness label follows the word definition, but only

if it is not related to disappointment. For instance, the customer can express their sadness from the operator's impossibility to solve their problem.

Disappointment. The customer express their disappointment towards something. The specific target object is not relevant to this label, as long as the disappointment is clearly expressed.

Frustration. Customer's frustration is an important factor in customer service conversations. With this label we want to identify all the messages with the frustration as the most important emotion expressed given the context. Hence, we do not make a distinction between mild and intense frustration, nor between implicit and explicit ways to express it.

Anger. The final negative leaning emotion label we consider is the anger. However, to better identify it from frustration, we only label a message with anger when the anger expression is explicit, very intense, and without any ambiguity towards the "frustration" label.

Along with these labels, we consider a final label which can be both mildly negative or positive depending on the context:

Surprise. The surprise emotion is usually difficult to distinguish from positive or negative labels such as frustration or amusement. As long as there is no ambiguity with the latter labels, we consider the message's main emotion to be the surprise. This label is totally dependent to the conversational context as a simple message such as "For real?!" can have multiple interpretations.

5.2. Conversation-level Labels

One of the specificity of our annotation schema (Figure 5) is the presence of conversation-level annotations. We made this annotation schema in order to annotate customer service conversation, this means previous message-level labels only serve as a first step to better identify the conversation-level labels. We consider 3 types of labels:

Problem Solving Status. We want to identify conversations by the solving of the problem faced by the customer upon arrival or latter on during the conversation. To do so, we make the distinction between 5 statuses, meaning 5 labels:

- **Solved.** The customer's problem is solved. The operator managed to give expected information or a way to solve the problem.
- **To be tested.** The operator gave a way to solve the problem but it is still to be tested whether or not it is sufficient to solve the customer's problem. By opposition with the "solved" label, the conversation does not indicate in anyway a positive result from the customer.
- **Out of scope.** The problem does not belong to the operator prerogatives and responsibility. In this case, operators usually try to redirect the customer or explicitly indicate they cannot help them on this matter.
- **No Solution.** The operator do not seem to find any solution for the problem at hand, nor do they find any external help or services to redirect the customer to. In the annotator's point of view, if any of the other labels do not work, this one is to be used.
- **Aborted.** This label can be seen as the equivalent to the "neutral" label for emotions. When the conversation do not hold any information in regards to the problem status, we use the "aborted" labels. For instance, when a conversation is too short and only contains one message "Hi, I have an issue with my ticket" and the customer does not answer further, we use this label. This label's real purpose is to help us identify conversations to be analyzed for the problem status.

Customer Satisfaction. We annotate the customer overall satisfaction by the end of the conversation. This label requires the annotator to keep in mind the previous message-level annotation context due to the satisfaction being closely related to the customer's emotions. We consider this label as a variant of the Likert scale (Likert, 1932) where we add the neutral (*i.e.* zero) value. The customer satisfaction thus ranges from -3 to +3 as shown in Table 2. This Table also shows a difference between unsatisfied label names and satisfied ones. To better distinguish them, the difference between dissatisfaction labels stems from the intensity of the dissatisfaction. However, the satisfaction labels consider the target of the satisfaction: "Midly Satisfied" refers to the customer being a bit satisfied but without any takes on specific problem parts. For instance, the customer can be satisfied from the interaction with the operator but not from the given solutions. "Partly Satisfied" refers to the customer being not fully satisfied from the solutions given to address their problem, but still being satisfied by some of them. "Fully Satisfied" label refers to the customer being satisfied by all the solutions given and their problem fully addressed. Beware for this label we consider a problem solved as a problem fully addressed, which means a customer could still be fully satisfied if he obtained the expected explanation even if this means his problem cannot be solved. This is for instance the case for an obviously impossible refund.

Customer Urgency. The final conversation-level label we consider is the customer urgency. It indicated whether or not the customer seems to be in an urgent situation, and as such is represented by a boolean value in the application (see the type field in Figure 5). An urgent situation can be implicit or implicit. For instance, the customer can express it as such: *"My plane is leaving in 10 minutes and I cannot find my reservation number."*

-3	Very Unsatisfied
-2	Unsatisfied
-1	Mildly Unsatisfied
0	None
+1	Mildly Satisfied
+2	Partly Satisfied
+3	Fully Satisfied

Table 2: Customer satisfaction labels for the current conversation.

6. EZCAT's First Usage

We used EZCAT to annotate confidential dyadic conversations from a customer service involving a customer seeking help and an operator employed to assist the customer (Guibon et al., 2021). To annotate those conversations we actually needed an easy-to-use application which would ensure the confidentiality by not storing data on external platforms. EZCAT has first been designed to serve this purpose, and we annotated our confidential corpus using the annotation schema (Section 5). This implies that we consider two annotation levels: message-level labels and conversation-level labels, following the exact structure as Figure 5. We started with a subset of 100 conversations to gain feedback and clarify the annotation schema interpretation across the 2 annotators.

Our confidential corpus is written in French and is made of 5,000 conversations from which we annotated a subset of 1,500 conversations, leading to a total of 20,754 messages. The average message length is 15.14 messages per conversation. We do not have a way to identify real speaker turns, and because all messages have a very short time difference in this corpus, we prefer not to infer speaker turns and consider the message as the unit of analysis. Moreover, due to this specificity, we voluntarily omitted the speaker turn annotation to ease the annotation process. This means the conversation context is a sequence of messages instead of a sequence of speaker turns which could have contained one or more messages artificially glued together.

Language	French
Max Msg/Conv	84
Avg Msg/Conv	13
Labels	11
Nb. Conv.	1,500

Table 3: Statistics for both datasets DailyDialog (DD) and Live Chat Customer Service (chat).

Figure 6 illustrates the distribution of emotion labels in the Live Chat Customer Service data set. We can see that the "neutral" label at the message-level is the most frequent by a large margin (81.5%), which makes it very unbalanced in terms of emotions. Excluding this label gives a slightly more balanced label set, as the satisfaction represents 44.9% of the other emotions, and the "frustration" 20.8%.

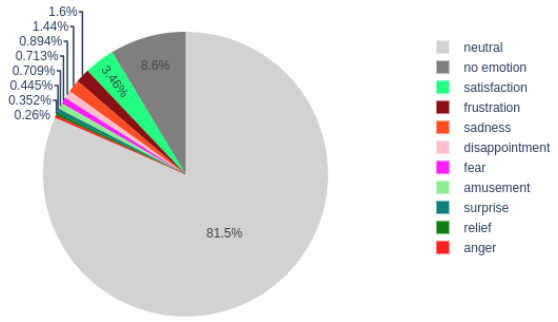


Figure 6: Emotion Distribution in Live Chat Customer Service

At the end of the annotation process, we computed Cohen's κ scores on the main 3 label types in order to display the inter-annotator agreement, and obtained substantial agreement at the message level and moderate agreement at the conversation level (Landis and Koch, 1977). According to Fleiss (Fleiss et al., 1981) κ values below 0.40 represent poor agreement, while values between 0.40 and 0.75 represent fair to good agreement. Values higher than 0.75 are excellent. Considering this interpretation, we obtained fair to good agreement scores.

Cohen's κ-scores for the 3 label types are as follows: 1) the emotions at the message level ($\kappa = 0.65$); 2) the visitor's satisfaction at the conversation level ($\kappa = 0.45$); and 3) the request's status at the conversation level ($\kappa = 0.46$). Emotions specific κ-scores are displayed in Table 4.

Emotion	κ-score
Amusement	0.1115
Anger	0.1608
Disappointment	0.1609
Frustration	0.1193
Neutral	0.3187
Fear	0.1111
Satisfaction	0.2068
Relief	0.1429
Surprise	0.1885
Sadness	0.2860
Global	0.6499
Global w/o Neutral and no_emotion	0.3885

Table 4: By-category agreement scores for emotions in Live Chat Customer Service

In regards to conversation-level annotations, the majority of the satisfaction labels are either positive or neutral with a majority of neutral (*i.e.* zero) values in Table 5. This creates an even more unbalanced distribution of the satisfaction labels than the emotion ones, as shown in Figure 7.

On the other hand, problem statuses are more balanced

satisfaction	1	0	-3	-1	3	2	-2
count	406	634	25	73	168	159	35

Table 5: Problem status distribution across the annotated data

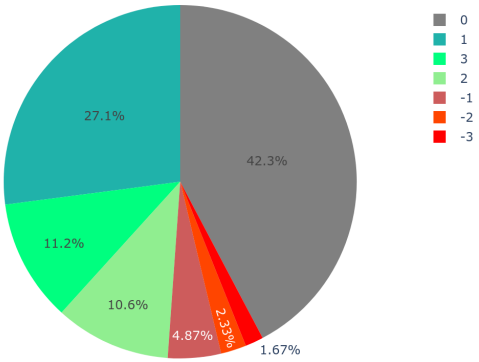


Figure 7: Satisfaction distribution on the annotated data

(Figure 8) even though a few conversations lead to "No solution" labels, with only 52 conversations, as visible in Table 6. This is explained by our annotation schema in which we indicate to the annotators to consider this label only if none of the other can be applied.

	type	count
0	Solved	538
1	Out of scope	198
2	To be tested	463
3	Aborted	249
4	No Solution	52

Table 6: Conversation counts per problem status

6.1. Annotators' Feedback on EZCAT

While annotating the data set, we received feedback from the usage of EZCAT. The first feedback concerns the annotation speed that is enhanced by the application being dedicated to the task: average estimated time is 50 conversation per hour. The ease of use noted by annotators led to more annotated conversations and thus, decreased the cost for the annotation campaign. The second feedback led us to improve the app, we made a few tweaks to help smooth the annotation process such as adding a possible constraint on enabling going from the first step (message-level annotation) to the second step (conversation-level annotation). Also, we changed the order of the emotion labels in the annotation schema configuration in order to center the most common label ("neutral") and, by doing so, preventing additional clicks or scroll downs by the annotator. This request is an example of direct feedback and requests to modify the app we received during the annotation campaign. Furthermore, we will integrate a feedback form in the application to further improve it.

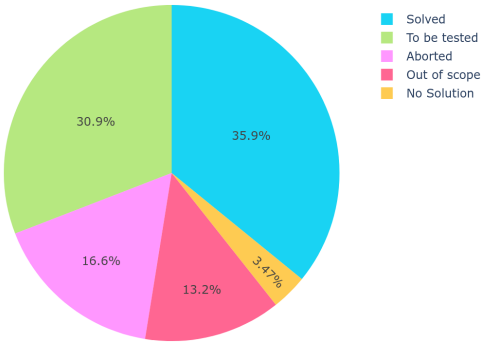


Figure 8: Problem status distribution across the annotated data

7. Future Improvements

We identify several improvements which would not alter the accessibility of EZCAT. First, we plan to enable the estimation of inter-annotator agreements within the application by loading directories of multiple files. In the annotation campaign we conducted (Section 6), we calculated the inter-annotator agreements using external Python scripts. EZCAT would benefit from including it for the dedicated JSON format with the corresponding schema configuration file. Second, we are working on integrating turns into the application as it can be relevant for specific dialogue data sets, as well as multiple labels per message to consider overlapping labels. For the moment, EZCAT's JSON format consider turns but the interface does not display them. We also plan on integrating imports of additional instant messages to expand it to other messaging applications which allow conversation exports such as iMessage. Finally, adding several languages to the interface, as the software architecture already allows it (Section 3), would further improve the accessibility of EZCAT.

8. Conclusion

In this paper, we presented EZCAT, an easy to use, freely accessible application dedicated to annotation of spontaneous text conversations. This application meets the need of easy access and fast annotation processes dedicated to private data, ranging from samples of confidential data or export of personal conversations from popular instant messaging applications. Along with the architecture, we detailed the first usage of the application for an annotation campaign dedicated to customer service conversations and the corresponding annotation schema that we designed for this task, considering two levels of annotation. EZCAT allowed us to faster annotate conversations, due to its simple usage and focus on one annotation process. While we already used the resulting annotated corpus (Guibon et al., 2021), we hope to help the community create their own annotated conversations using EZCAT to ease the starting process and suit their own annotation schema whether it is based on emotion, dialog act, or only considering conversation-level labels.

9. Acknowledgements

This project has received funding from SNCF (the French National Railway Company), the French National Research Agency's grant ANR-17-MAOI, and the DSAIDIS chair at Télécom-Paris. We give special thanks to Hélène Flamein and Aurore Lessieux for the help provided.

10. Bibliographical References

Chapuis, E., Colombo, P., Manica, M., Labeau, M., and Clavel, C. (2020). Hierarchical pre-training for sequence labelling in spoken dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2636–2648.

Chowdhury, S. A., Stepanov, E. A., Riccardi, G., et al. (2016). Predicting user satisfaction from turn-taking in spoken conversations. In *Interspeech*, pages 2910–2914.

Collins, E., Rozanov, N., and Zhang, B. (2019). Lida: Lightweight interactive dialogue annotator. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 121–126.

Colombo, P., Chapuis, E., Manica, M., Vignon, E., Varni, G., and Clavel, C. (2020). Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7594–7601.

Cucurnia, D., Rozanov, N., Sucameli, I., Ciuffoletti, A., and Simi, M. (2021). Matilda-multi-annotator multi-language interactivelight-weight dialogue annotator. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 32–39.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175.

Danesi, C. and Clavel, C. (2010). Impact of spontaneous speech features on business concept detection: a study of call-centre data. In *Proceedings of the 2010 international workshop on Searching spontaneous conversational speech*, pages 11–14.

Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., and Cieliebak, M. (2021). Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810.

Duran, N. and Battle, S. (2018). Conversation analysis structured dialogue for multi-domain dialogue management. In *The International Workshop on Dialogue, Explanation and Argumentation in Human-Agent Interaction (DEXAHAI), December. https://sites. google. com/view/dexahai-18/home*.

Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Fleiss, J. L., Levin, B., Paik, M. C., et al. (1981). The measurement of interrater agreement. *Statistical methods for rates and proportions*, 2(212-236):22–23.

Guibon, G., Courtin, M., Gerdes, K., and Guillaume, B. (2020). When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5293–5302, Marseille, France, May. European Language Resources Association.

Guibon, G., Labeau, M., Flamein, H., Lefeuvre, L., and Clavel, C. (2021). Few-shot emotion recognition in conversation with sequential prototypical networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6858–6870.

Heeman, P. A., Yang, F., and Strayer, S. E. (2002). Dialogueview-an annotation tool for dialogue. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, pages 50–59.

Jurafsky, D., Shriberg, E., and Biasca, D. (1997). Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO.

Klie, J.-C., Bugert, M., Boullosa, B., de Castilho, R. E., and Gurevych, I. (2018). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Landis, J. R. and Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Langlet, C., Duplessis, G. D., and Clavel, C. (2017). A web-based platform for annotating sentiment-related phenomena in human-agent conversations. In Jonas Beskow, et al., editors, *Intelligent Virtual Agents*, pages 239–242, Cham. Springer International Publishing.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Ma, Y., Nguyen, K. L., Xing, F. Z., and Cambria, E. (2020). A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Nagata, N. (2002). Banzai: An application of natural language processing to web-based language learning. *CALICO journal*, pages 583–599.

Nie, L., Wang, W., Hong, R., Wang, M., and Tian, Q. (2019). Multimodal dialog system: Generating responses via adaptive decoders. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1098–1106.

Novielli, N., Calefato, F., and Lanubile, F. (2018). A gold standard for emotion annotation in stack overflow. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pages 14–17. IEEE.

Novikova, J., Dušek, O., and Rieser, V. (2017). The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Razumovskaia, E., Glavaš, G., Majewska, O., Ponti, E. M., Korhonen, A., and Vulić, I. (2021). Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems. *arXiv preprint arXiv:2104.08570*.

Shriberg, E., Bates, R., Taylor, P., Stolcke, A., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., and Van Ess-Dykema, C. (1998). Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):439–487.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Meteer, M., and Van Ess-Dykema, C. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–371.

Widlöcher, A. and Mathet, Y. (2012). The glozz platform: A corpus annotation and mining tool. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 171–180.

Witte, R. and Gitzinger, T. (2008). Semantic assistants–user-centric natural language processing services for desktop clients. In *Asian Semantic Web Conference*, pages 360–374. Springer.

Yang, F. and Heeman, P. A. (2005). Dialogueview: an annotation tool for dialogue. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 20–21.

Zhu, C., Zeng, M., and Huang, X. (2019). Multi-task learning for natural language generation in task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1261–1266.