Analyzing Patterns of Conversational Breakdown in Human-Chatbot Customer Service Conversations

Analyzing Patterns of Conversational Breakdown in Human-Chatbot Customer Service Conversations

Anouck Braggaar[1(✉)], Florian Kunneman[2], and Emiel van Miltenburg[1]

[1] Tilburg University, PO Box 90153, 5000, LE Tilburg, The Netherlands
{a.r.y.braggaar,c.w.j.vanmiltenburg}@tilburguniversity.edu
[2] Utrecht University, Trans 10, 3512, JK Utrecht, The Netherlands
f.a.kunneman@uu.nl

Abstract. Many chatbots still struggle with correctly interpreting and responding to user enquiries. Therefore, it is important to figure out how and why chatbot-human conversations break down. In this study we analyzed features in user-utterances directly before a bot-initiated repair to determine their presence and prominence as possible predictors of conversational breakdowns. We used data from a real-life public transport customer service chatbot to demonstrate the errors that occur in actual deployed systems. The analysis shows that there are some features (such as commonness, outdated words, and unexpected words) that occur more often in utterances directly before a repair. Some features also correlate with each other and occur together, such as outdated words and subjectivity. By using feature analysis, many opportunities for improvement can be found either live (during the interaction) or afterwards.

Keywords: Chatbots · Customer service · Breakdowns · Features analysis

1 Introduction

Despite recent developments in the field of conversational AI, customers often encounter conversational breakdowns when chatting with a customer service chatbot. Many chatbots are not able to provide adequate answers to user requests [12]. According to the National Voice Monitor (2024), in the Netherlands only eight percent of user questions is correctly answered by the chatbot [49]. Although new possibilities involving generative AI should make it easier to prevent conversational breakdowns, companies are often hesitant to fully exploit this technology as they are afraid of the possible risks this will bring [49]. Consequently, many chatbots in customer service are often still intent-based (recognizing the user's intent by using natural language understanding [31]) or rule-based (following a pre-defined path and/or uses simple pattern-matching [31]).

Sometimes chatbots are deployed online at early stages of development for incremental improvement, however this approach can lead to frustration with the

A. Følstad et al. (Eds.): CONVERSATIONS 2024, LNCS 15545, pp. 3–22, 2025.
https://doi.org/10.1007/978-3-031-88045-2_1

user [15]. Conversational breakdowns in chatbot conversations are detrimental to user experience, often leading to user frustration [1] which could potentially lead to lower re-use intentions or an overall downgraded experience with chatbots. Therefore, these breakdowns should be avoided as much as possible. This paper aims to identify different predictors of conversational breakdowns in a real-world customer service chatbot.

Starting point. Our paper builds on earlier work by Reinkemeier and Gnewuch [40]. We also analyze conversations from a real-world task-oriented dialogue system. However, the domain and approach are different. Whereas Reinkemeier and Gnewuch [40] focus on user-chatbot interactions from an insurance company, we focus on user-chatbot interactions from a Dutch public transport company. Through an ongoing collaboration with this company, we are able to obtain large amounts of real-life data in the public transport domain. As Zumstein and Hundertmark [51] discuss, this is a very rich domain where chatbots typically offer many different services, from buying tickets to providing timetable information. Of course, offering a wide array of different services also means that there is more room for the chatbot to misunderstand the user's intentions, potentially leading to conversational breakdowns. We aim to identify different causes of those breakdowns.

Approach. Reinkemeier and Gnewuch [40] set out to create a classifier to predict the breakdown type based on four types of breakdowns ('*elaborated*', '*specific*', '*brief*', and '*cryptic*'). These breakdown types were detected by clustering user utterances with low intent recognition scores into four categories. Reinkemeier and Gnewuch [40] decided on four categories because this was the optimal number found in a pre-clustering based on agglomeration coefficients. After clustering they labeled the four categories based on the user utterances in the categories. The classifier was then able to compute the distance of a new user message to the cluster centers. While they show that this classifier can quite accurately predict these four categories [40], much information is lost by differentiating user utterances in four general categories. We initially also attempted to cluster the user-utterances, using the HDBSCAN clustering algorithm [35] with BERT embeddings to represent the different user utterances [46], but this method mostly lead to many noise points (points that could not be fitted in one of the clusters) and no meaningful clusters.[1] Thus, we took a step back: to better understand how breakdowns arise, we carried out a quantitative corpus analysis. This paper identifies several different properties of the user utterances that may lead to conversational breakdowns, and statistically compares the differences between messages that either do or do not result in repairs.

Research Question. We aim to zoom in on how user utterances prior to repair differ on the presence and prominence of particular features. This will facilitate a more fine-grained insight into factors that may influence non-understanding

[1] The algorithm mostly clustered on length. The code and results for the clustering can be found on GitHub.

and how to act upon this. With this analysis we will aim to answer the following question, **RQ**: To what extent are particular features useful as a signal of a conversational breakdown? By answering this question we will learn more about the capabilities of the system and will be able to see on which points it needs improvement. As we will see, simple features can be effective in finding different ways to improve existing task-oriented dialogue systems. The features can be used in real-time dialogue breakdown detection or used to assess where the chatbot could be improved. Our case-study highlights the abundance of low-hanging fruit in an already deployed commercial system that serves thousands of users everyday. It is likely that smaller providers of dialogue systems will benefit from these results as well.

2 Previous Work

2.1 Classification of Errors

Many papers try to find patterns in the kinds of breakdowns that occur in human-chatbot conversations. As a result, many overviews and taxonomies have been created that show what kind of breakdowns occur in (spoken) dialogue system interactions. Table 1 gives a comprehensive overview of some of the existing taxonomies. The table shows how the different categories relate to each other. Two main categories are often distinguished, misunderstanding (wrongly interpreting the user) and non-understanding (having no interpretation at all) [5,23]. Within this category multiple different categorizations have been made. For example, Higashinaka et al. [18,20] distinguish four categories of errors: (1) utterance-level (error in form of content of language [18]), (2) society-level (violating a social norm [18]), (3) response-level (system ignores how it should answer the user [18]), and (4) context-level (topic relevant and consistency [18]). Sanguinetti et al. [42] on the other hand make a distinction into four categories based on the maxims of Grice [16]: (1) quantity (the utterance should have sufficient information, not too much [16]), (2) manner (be clear about the utterance [16]), and (3) relation (utterance should be relevant to the conversation [16]). Within these categories, more specific error types can be distinguished such as an excess of information (within the quantity type). Lastly, Sanguinetti et al. [42] define a generic error class for all other errors and for non-cooperativity of the user (the user intentionally obstructing the system).

Other work creates an overview of errors based on a diverse background of research disciplines [39]. This involves errors in human-human communication and also human-computer communication. Paek [39] proposes that there are four levels that are necessary to reach understanding between conversation partners. If one or multiple of these levels fail, an error could occur. Different combinations of failing channels result in different kinds of errors [39], such as misunderstanding and non-understanding. There is also work focusing on the errors occurring specifically in spoken dialogue systems (e.g. [37]). Since different domains and modalities lead to different errors (e.g. speech recognition errors in spoken systems) and consequently different repairs (e.g. handovers to a human employee

Table 1. Categorization of chatbot and user errors. Based on and adapted from [5, 18, 20, 22, 23, 42].

Error	Source	Bot	User	Description
Misunderstandings	[5,23]	✓	✓	Obtaining an incorrect understanding of the user utterance.
Utterance-level errors	[18,20]			
Quantity	[42]	✓	✓	Based on the maxim of quantity [16].
Lack of information	[18,20,42]	✓	✓	The message does not contain enough information to proceed.
Excess of information	[42]	✓	✓	The message contains too much information.
Manner	[42]	✓	✓	Based on maxim of manner [16].
Ill formed /general quality	[22,42]		✓	The utterance is not in a way the system can understand.
Indirect question	[42]		✓	Questions that seem implicit and indirect.
Non understandable	[18,20,22,42]		✓	A completely non-understandable/uninterpretable utterance.
Grammatical/word usage /expression error	[18,20,22,42]	✓		The utterance contains grammatical mistakes, unnatural expressions or a specific word usage.
Semantic error	[18,20]		✓	There are semantic mistakes, the utterance is not valid.
Wrong information	[18,20,22]		✓	The information in the utterance is wrong.
Society-level errors	[18,20]			
Lack of sociality	[18,20,22]	✓	✓	Contains undesirable speech, no considerations for the other.
Lack of common sense	[18,20,22]	✓	✓	Different opinion that is not similar to the general consensus.
Non-understandings	[5,23]	✓	✓	Not understanding anything the user says.
Response-level errors	[18,20]			
Context-level errors				
Relation	[42]	✓	✓	Based on maxim of relation [16].
Indirect response	[42]	✓	✓	Indirectly answering the previous question.
Ignoring	[18,20,22,42]	✓	✓	The participant ignores the previous utterance. In some cases also the answer type that is requested is ignored.
Ignoring expectations	[18,20]	✓		The bot might correctly answer the question but the answer might be lacking in what could be expected for such a question.
Repetition	[18,20,22,42]	✓	✓	Repeating the previous utterances.
Straight wrong response	[42]	✓	✓	A utterance that is not relevant.
Topic change	[18,20,22,42]	✓		The topic is unexpectedly changed.
Unclear intention	[18,20,22]	✓	✓	It is not clear why something is mentioned at that specific time in the conversation (although the topic can be appropriate).
Answering with question	[42]		✓	Answering a posed question with a new question.
(Self-)contradiction	[18,20,22]	✓	✓	Contradicting own utterance or the utterance of the other.
Generic	[42]			
Non-cooperativity	[42]		✓	Not providing requested information.
Other	[42]	✓	✓	Any other error.

in customer service), we focus on only one modality in one specific domain. In this paper we will analyze which of a selected set of features could potentially lead to an error and consequently repair. Although some of the types of errors in Table 1 can not directly be deduced from formal properties of utterances (such as society level errors), the cues to others (such as grammatical mistakes) can directly be deduced from the user utterance. These features can thus be seen as direct or indirect indicators of the error types.

2.2 Automatic Detection of Errors

The automatic detection of breakdowns has become a popular focus of study. One can either detect breakdowns by detecting patterns in large amounts of data and consequently update the system, or use knowledge about breakdowns and detect errors in real-time. The goal of the current study is to detect patterns in a large amount of data.

Bernard and Balog [3] for example automatically detect three types of breakdowns. They use the system logs of conversations between the system and a user simulator to automatically detect if there is a system failure that results in a dialogue breakdown. The authors consider three breakdowns in their study. First, 'system error' which identifies system bugs. Secondly 'dialogue of the deaf' which consists for example out of conversational loops with a system repeating the same utterance. Lastly, they identify 'conversation flow discontinuation' which includes unexpected utterances [3]. In the case of the last error they compare the intents to a dialogue flow to see if recognized intents line up.

The shared task 'dialogue breakdown detection challenge' [21] comes with the task to detect breakdowns in chat-oriented dialogue systems. This task, which has now been organized multiple times, sets as goal for participants to detect utterances that lead to breakdowns in data sets in both English and Japanese [19]. Participants get a small set of utterances as context and have to detect if the next system utterance would lead to a breakdown [21]. They need to label utterances into three categories: (1) no breakdown, (2) possible breakdown, and (3) breakdown [19]. The data were labeled by human workers by breakdown types [19]. Therefore participants were able to treat this as type of classification task, resulting in different approaches such as designing LSTMs [17] or BERT [44]. For the English data, results were close to human performance [19].

As mentioned in the introduction (Sect. 1), the current paper will expand on the work of Reinkemeier and Gnewuch [40]. They cluster user utterances that result in breakdown based on four features: (1) semantic weight, (2) unknown words, (3) word count and (4) words per sentence. The clusters are analyzed and named according to the information (texts) in the clusters. Cluster one contains elaborate messages (very long messages), two contains specific ones (specific questions that a chatbot might not even be able to answer), three contains brief messages (very short ones) and the last cluster contains cryptic messages (which contain many unknown words). With these clusters they create a classifier that is able to detect to what category the user message belongs during a conversation [40].

In this paper we aim to extend upon the work of Reinkemeier and Gnewuch [40] by extending the set of features with other features that might be indicative of breakdowns and consequently examining if feature analysis can be predictive of conversational breakdowns.

3 Methodology

In this study we used a top-down approach with features that might be a predictor for a breakdown. These features were then compared to a control set of user utterances that do not occur together with a breakdown. Below, we will first discuss the total dataset followed by the selection process for our repair and control set. In Sect. 3.2 the used features are discussed both in terms of their implementation as well as the associated expectations on how they could be predictive of a breakdown.

3.1 Data

For this study we used real conversational data where a chatbot answers questions from actual customers of a Dutch public transport company. The chatbot has been in use for an extended period of time before the data was sampled, and serves a high volume of interactions.[2] It is professionally maintained, with continuous updates to improve the quality of the conversations. Given the high quality of the chatbot, we expect that any persistent causes of conversational breakdowns are also likely to occur in similar chatbots.

General Properties. The data were sampled by the company for a period of one year (between January 2021 up and until January 2022). In total the data contained 108,370 distinct session IDs. Within these sessions not only the conversations are recorded but also (technical) user actions such as only starting the chat and immediately leaving. Excluding sessions with only technical actions results in a total of 108,141 conversations. The chatbot conversations are primarily in Dutch although it sometimes happens that a costumer asks a question in English. In about 12% of the conversations there is at least one utterance marked as English by the intent-recognizer. The technical user actions such as joining are not very informative if we want to calculate conversation length. Therefore these actions (joining, re-joining, and leaving the chat) were removed, which resulted in conversations with a length of on average 17.85 turns. This does include dialogue with the human agent after handover. The data were anonymized by replacing personal information by either *<FILTERED>* or in the case of numbers by zeros.

[2] We are not at liberty to disclose the exact usage frequency.

Chatbot Trigger Data. Next to the chat data we also obtained the training triggers from the chatbot. The training triggers contain the sentences on which the intent recognition of the bot is trained, grouped by topic (which is the intent name). For example, 'Kwijt' ('Lost') triggers the lost and found intent, 'Vraag over factuur' ('Question about invoice') triggers the invoices intent and 'Can we speak in English' triggers the English intent. The trigger data were collected around May 2023, and are not completely synchronized with the trigger data at the time of the chats themselves. Dialogue designers continually refine the training intents and triggers. This means that when the chats and triggers are collected there can be a discrepancy between the collected triggers and the triggers actually used during these chats. Nonetheless we think that the triggers sketch a sufficient picture of both the domain words as well as the out-of-domain words, as the triggers capture key patterns in the conversation which presumably are stable over time.

Utterances Before Repair Test Data. For the analyses of utterances that occur before a system-initiated repair a subset of the total data set was made. Data were selected by searching for pre-determined system initiated repairs.[3] These repairs are thus used as a proxy to automatically detect points of possible breakdowns. As this system is intent-based and conveys pre-determined responses to the user, this was an easy way to find points in the conversations were a possible breakdown occurred. This also makes analyzing the chatbot responses themselves less interesting as they are always following similar patterns.

We chose two system-initiated repairs that potentially could hint to a breakdown. These repairs are standard expressions of the system and therefore a reliable way to determine where a repair is initiated. We chose two of the most general repairs, where no sign of understanding is given by the system. We focus only on non-understanding as the system cannot guide the user in the correct direction to continue the conversation with the bot (no intent is detected). Other repairs that showed any understanding of the system have been omitted. This includes repairs like 'I think you have a question about a subscription ...' and then proceeding to give options about subscriptions. The following two repairs are selected:

1. The system does not understand the user and asks to rephrase the utterance, thereby hinting to make the new utterance short and concise ('Unfortunately I don't fully understand what you mean. Could you rephrase the question in different words? Tip: I understand short and concise questions the best.')
2. The system knows it is not able to help the user and suggest redirecting to a human employee. ('I'm sorry, I believe I can not help you yet. Shall I connect you with my colleague?')

Both the utterances show that the bot does not recognize the appropriate intent. The second seems to mostly follow on the first repair (no understanding) if the

[3] All code for data selection and clustering can be found on GitHub.

bot still does not recognize an intent, even after rephrasing by the user (they are labeled in the data as default fallback and default consecutive). The second one is actually a repair that redirects to the employee and signals that the bot is not capable of solving the users request. Redirecting can occur in two situations: (1) the bot does not understand, and (2) the bot recognizes the question but knows it is not capable to help the user. We see that this one is used frequently and with numerous different user utterances preceding it, signalling that there is (still) no understanding (no recognized intent) by the bot. In 9,591 unique conversations at least one of these two repairs occurred. A total of 11,913 user utterances are found to appear before a system-initiated repair. On average, the repairs occurs at turn 13.34. Most often a repair occurs at turn three which means that it is initiated after the first user utterance.

Utterances Not Before a Repair Control Data. We also randomly selected utterances that occurred in conversations that did not include one of the selected repairs. The same number of 11,913 user utterances were selected for this data set. These data function as a control set with which the utterances occurring before a repair can be compared. The utterances we selected for this data set occurred in the first, second or third user turn, all before a bot utterance. We selected these utterances early in the conversation because in many cases when a repair is normally introduced this is often at turn three. At the same time, we only selected those of conversations that took up at least eight turns, so that conversations that only consist out of a user joining, a greeting and then leaving are dismissed. We suspected that some of the turns might actually be button options, therefore we manually identified 25 utterances that turned out to be buttons (or a general phrase used for testing the system), and excluded these from the data set. It is a challenging task to select turns that are (possibly) well understood by the system, because only a manual inspection will tell if it is really a well understood turn.

3.2 Features

The data were analyzed based on a selection of features, using the study of Reinkemeier and Gnewuch [40] as a starting point. They worked with semantic weight, total word count, words per sentence and percentage of unknown words. The features used in this study can be found in the list below and have been chosen because they have been used in earlier work (such as Reinkemeir and Gnewuch [40]) or because there are unwritten assumptions that features such as these may result in breakdowns in intent-based dialogue systems.

Counts. The number of words (split on spaces), sentences (tokenized with sentence tokenizer from NLTK [4]) and characters in the utterance (includes all characters, also punctuation marks). Table 1 shows that an excess of information can lead to a conversational breakdown. Previous work by Braggaar et al. [7] shows that this particular error occurs often in chatbot customer service conversations. Therefore, we suspect that the longer the user utterances are, the more difficult it is for the intent-recognizer to correctly classify the appropriate user intent.

Sentiment. Polarity and subjectivity are calculated and used as features by using the Pattern library in Python [10]. Previous work shows that user sentiment influences the experience users have with a chatbot resulting in lower user satisfaction when a humanized chatbot is used [9]. Different factors such as not understanding the user or chatbots failing to properly respond could all potentially lead to frustration [8]. We suspect that user sentiment is not only the result of a breakdown but could also be a predictor to a breakdown. Users expressing frustration may indicate that the chatbot was not able to correctly respond to their utterances so far and that the conversation may eventually lead to a breakdown.

References. Referencing to external entities or entities described earlier in the conversation might raise issues for the system. The list of used (Dutch) reference words can be found on GitHub. Many task-oriented data sets have based their interactions on a model that only takes into account the current user utterance [25]. In this case the intent is recognized based on only one user utterance and the recognizer does not take into account the context. Jakobovits et al. [25] argue that systems should possess 'conversationality' that takes into account more than just the current user turn by for example being able to appropriately handle references. The dialogue system in this study also works mostly intent-based, so we assume not much context is taken into account when answering a single user question. Therefore, using (many) reference words might lead to a possible breakdown.

Commonness. A combination of commonness and unknown words is made to predict wrong interpretations. Words with high scores for commonness that are not in the training data of the bot could indicate a wrong interpretation. Commonness is calculated based on the occurrences of entities on Wikipedia.[4] Wikipedia is used to obtain a general picture of commonly used words in the Dutch language to show which content words are used by the customer that do not belong to this specific customer service domain.

We worked with commonness as it is defined and used in paper such as Meij et al. [36] and Kunneman and Van den Bosch [29].[5] Words get assigned a score based on the probability that the word is used as anchor text in Wikipedia [29,36]. When a word is common (highly likely in the Wikipedia data) but not in the bot training data, this is reflected by a count for commonness.

Common words that are frequently used but not present in the training data might indicate a possible reason for a conversational breakdown. Recent work into domain adaptation of intent recognition has shown that there are some limitations to applying an intent recognizer to a different domain due to

---

[4] A Wikipedia-dump was used from May 2023, this includes all Dutch Wikipedia data. To decide how common entities should be, we decided on a cutoff value of 0.25 for unigrams, 0.5 for bigrams and 0.75 for 3–5 grams

[5] We adapted and used code from https://github.com/fkunneman/DTE/blob/master/dte/classes/commonness.py.

being out of scope [2]. Similar limitations could occur with out-of-domain vocabulary. Research has also shown that users often have high expectations of the chatbot's capabilities, ask very complex question or switch between multiple topics [30]. This could mean that customer vocabulary and chatbot vocabulary do not match, which leads to breakdowns.

Spelling/grammar. LanguageTool [38] is used to count different errors in the utterances. A selection was made to include errors that possibly have a large impact on how the chatbot understands the user, including for example typos and repetitions. Not all categories on the LanguageTool webpage and the LanguageTool source code have consistent examples or explanations. Therefore, the selection consisted out of items that have clear explanations and are useful in the context of chatbot errors (see Table 1). We included the following categories:[6]

- *Loan words.* Words or sentence structures adopted from other languages such as English.
- *Offensive.* Offensive or inappropriate language
- *Proper nouns.* Names of people.
- *Grammar.* Wrong inflections of words, missing letters etc.
- *Repetition.* Repetition of words.
- *Readability.* Very long words.
- *Typos.* Typing mistakes.
- *Outdated.* Outdated (old-fashioned) words, not used very often anymore.
- *Dyslexia.* Mistakes relating to dyslexia such as writing 'mir' instead of 'meer' ('more').
- *Form errors.* For example 'als/dan' (confusing 'if' and 'then').
- *Confused words.* Mixing up words or phrases.
- *Unexpected.* Unexpected sequences of words.
- *Full sentence.* Complete sentence with error such as the wrong inflection of a verb.
- *Other.* Other rules that do not fit a category, such as a sentence unexpectedly ending.

We suspect that the intent-recognizer has not been trained specifically to handle some of these features. For example with outdated words, older users may use a different vocabulary than younger uses which might not have been taken into account by the conversational designers (previous work shows that age is an important factor related to vocabulary size [26]). Spelling and grammatical mistakes might impact the correct recognition of intents and therefore lead to a breakdown. Martinovsky and Traum [34] for example show that multiple features such as ellipsis could lead to user frustration and possibly breakdowns in human-machine interaction. In addition, Jakobovits et al. [25] argue that systems should be able to normalize utterances, for example reducing the impact of typos for intent recognition. However, instead of normalizing the texts, others have argued to adapt the tools themselves to be able to handle such texts [11].

[6] More extended explanations of categories can be found on the LanguageTool website and in the LanguageTool Dutch grammar source code.

3.3 Analysis

To begin with, a comparison is made with utterances that do not occur before a repair (control data) by examining the frequency ratio. To see if the means of the features between test and control data significantly differ, a t-test was conducted with SciPy [45] with Holm-Bonferroni [24] correction in statsmodels [43] to correct for testing multiple features. Secondly, the correlations between the features in the test set were calculated to examine how often features occur together and the usefulness of certain combinations of features.

4 Results

We examined the presence and prominence of a selected set of features in a dataset with user utterances occurring before a repair (the test data). The results are compared to a data set that contains user utterances occurring in conversations without any of the used repairs (the control data). Both data sets contained 11,913 user utterances (see Sect. 3.1 for more information).

4.1 Uninformative Features

Table 2 shows the means and standard deviations of the features of the test data and the frequency ratios and difference in means between test and control utterances. Some features do not occur in both datasets at all (full sentence, grammar, offensive and proper nouns). While others rarely occur, namely those having a mean close to zero (such as loanwords). The readability feature only occurs in the test data and not in the control set (which results in a division by 0). The feature readability is only detected twice in the data, both in utterances in which the user adds a complete web page. These texts are copy-pasted without the proper use of spaces (for example between different headers on the web page).

4.2 Differences Between Conversations with and Without Repairs

We conducted multiple t-tests to examine if the means of the two data sets significantly ($\alpha <= 0.05$) differ per feature. The results are provided in Table 2, where asterisks indicate whether the difference between the means is significant. The frequency ratio provides another indication of the size of the difference.

The frequency ratio of the features is often above one, which means that the feature is more prominent in the pre-repair test data than in the control set. Some of the features have a remarkably high ratio (such as commonness and dyslexia), suggesting that these features are far more prominent in the pre-repair data. The only feature that has a ratio below one (indicating that the feature is less prominent in the pre-repair data) is reference words. This is striking as one would expect more reference words in utterances that cause a breakdown. The frequency ratio of the counts is very close to one, meaning that it occurs almost as often in the control data as in the test data (there are some zero counts for words/characters/sentences as users sometimes only pressed enter).

Table 2. Means, standard deviations, effect size (Cohen's d), difference between means (test - control), frequency ratios between repair and control data (* if p-value <= 0.05).

	Feature	Mean	SD	Effect size	Δ Mean	Frequency ratio
Counts	Word count	16.745	38.818	0.258	7.554*	0.997
	Character count	104.891	249.083	0.254	47.447*	0.997
	Sentence count	1.755	2.226	0.282	0.480*	0.997
Sentiment	Polarity	−0.005	0.165	0.016	0.002	1.362
	Subjectivity	0.150	0.276	0.175	0.045*	1.195
References	Reference words	1.161	2.116	0.109	0.195*	0.759
Commonness	Commonness	0.031	0.218	0.134	0.023*	3.071
Spelling/grammar	Dyslexia	0.002	0.046	0.036	0.001*	4.000
	Outdated	0.031	0.254	0.118	0.023*	2.717
	Repetition	0.007	0.091	0.058	0.004*	2.515
	Loanwords	0.001	0.035	0.018	0.001	2.333
	Other	0.003	0.065	0.040	0.002*	2.312
	Form errors	0.002	0.041	0.021	0.001	2.125
	Unexpected	0.022	0.170	0.091	0.013*	2.120
	Confused words	0.008	0.100	0.050	0.004*	1.957
	Typos	1.153	4.692	0.169	0.631*	1.737
	Full sentence	0.000	0.000	-	0.000	-
	Grammar	0.000	0.000	-	0.000	-
	Offensive	0.000	0.000	-	0.000	-
	Proper nouns	0.000	0.000	-	0.000	-
	Readability	0.000	0.026	0.018	0.000	Division by 0

Most of the means differ significantly, except for form errors and loanwords, these also very rarely occur in the test data. But, the effect sizes of all features are (very) low, suggesting limited differences between the groups. For words, the test data returns a mean of 16.7450 while for the control data the mean is 9.2250, which is quite a bit lower. We suspect that people sometimes overload the chatbot by giving an excess of information (see Table 1).

4.3 Correlations Between Features

Figure 1 shows the correlations between different features for the test data. We have removed the features that did not occur in the data at all (full sentence, grammar, offensive, proper nouns). Examining the correlations can shed light on how often features occur together and if it is useful to take into account these combinations for further improvement of the system.

Most correlations are quite low. However, the correlations with the items that count words, sentences and characters are in many cases quite high. This makes sense as one, for example, could expect more typos when sentences get extended. The correlation between subjectivity and reference words is also quite high, but this follows logically from how these features are measured. Pattern follows a rule-based approach to measure polarity and subjectivity [14], which possibly overlaps with the reference words list the authors compiled.

There are some interesting combinations to be found in this figure. Most of these combinations comprise either the subjectivity or reference words features. An example of an utterance with a high score for subjectivity and two counts of outdated words is someone explaining that he wants a refund. Two outdated words stand out immediately, namely 'aldaar' ('there') and 'hetgeen' ('which'), both words referring to either something in a previous sentence or in the same sentence. It can be questioned if the bot would stumble on these utterances as they do not convey any meaning on their own. An utterance that includes high counts for reference words, commonness and unexpected words is from a user complaining about a bike accident with a tourist. The user shares detailed information about the incident and ends with 'I will wait for your response. (...) Regards, (...)'. From the way the utterance is formulated it is not clear if the user requires a response. An inspection of the data shows that the user merely wants the complaint to reach the complaint division of the company. Overall, the relatively low correlations show that these errors do not occur together very frequently, indicating that it might be worth focusing on single predictors when creating repairs.

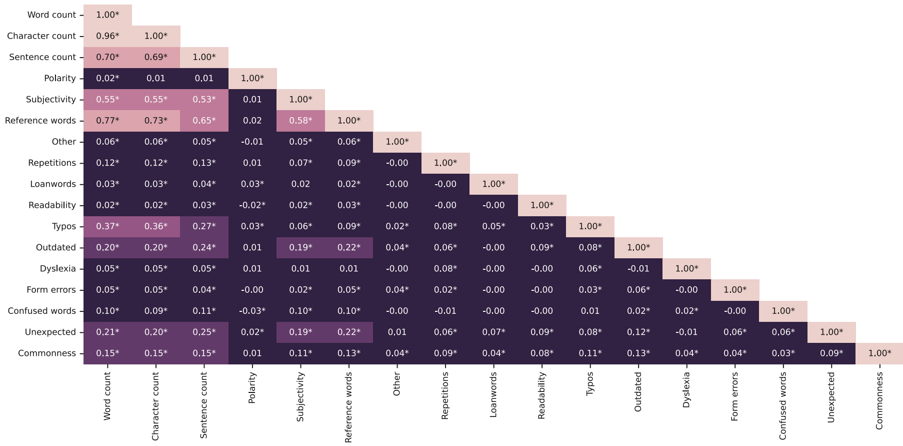

Fig. 1. Feature correlation of the utterances before a repair. * if p-value <= 0.05.

5 Discussion

This research aimed at exploring user utterances that lead to a repair and consequently could lead to a conversational breakdown. By examining the characteristics of the utterances by creating simple features, we set out to investigate if these can actually be used to either prevent breakdowns or repair them using this information.

We decided on multiple features that could be predictive of a breakdown. Interestingly, the results (Sect. 4) show that there are actually differences in the occurrence of features between utterances before a repair (test data) and utterances without repair (control data). Many features actually occur much more in utterances before a repair, suggesting that these features might be indicative for a possible breakdown. Similarly, the mean word count is almost twice as high for utterances directly before a repair. This suggests that users present too much information to the system (see Table 1, excess of information is an identified error [42]). Consequently, counting the number of words is a quick win and a more fine-grained analysis might not be necessary. To some extent, the results of the feature analysis could be traced back to the results of Reinkemeier and Gnewuch [40] in which too much information belongs to their 'elaborate' category. In comparison, an advantage of our more fine-grained selection of features is that certain inferences can be made from their combinations. Individual features provide a clearer picture of the difficulties in understanding. Combinations, such as an elaborate message with unknown common words, can be even more informative. Such insights can be used to give a more informative response to the user about the cause for non-understanding. Yet it remains to be tested whether it is actually necessary to provide more information or if 'be brief' is enough.

5.1 Limitations

The first limitation of this study is that deployed chatbots are not static objects. Since the chatbot is continuously updated to improve the service, it is a moving target. For example, we have seen that the training (trigger) data and the chatbot conversations are not obtained at the same time and do not completely line up. We see for example that at the time of the conversations there was a Covid-19 intent, but this was no longer included in the training. These issues make it harder to study the exact causes of conversational breakdowns. Still it is important to work with ecologically valid data. Thus we believe that the development of best practices for dealing with changes over time is an important research problem.

There are also many different ways to implement these features and the existing methods are not specifically designed for this kind of data. The choices made in implementation thus influence the results of the feature detection. For example, a tool like LanguageTool does not operate flawlessly, especially in this case where the data sometimes do not consist out of 'regular' full sentences. Other tools could be considered in the future such as T-Scan [32] to see how these specific implementations impact the results.

Data preprocessing also influenced some of the features. Due to the nature of the data, we were given a previously anonymized data set. Unfortunately, not in all cases anonymization was done correctly. In some cases personal information was still in the data, in other cases other words (like nouns or articles) are removed. This potentially impacts the derived features.

Creating an Anonymized Data Set. One of the key issues we are dealing with in this paper is the nature of the data. We would like to share the data for reproducibility purposes, but as the data are private and obtained through a company, we cannot make all of the data public. Data anonymization is often seen as a complex task [33], with many challenges for unstructured data [47]. Previous approaches such as Deperson have been created to anonymize sensitive customer data [6]. However, these approaches are not 100% flawless and can not ensure complete privacy. This poses challenges for sharing the data while maintaining complete privacy for both the users and the company. Consequently, we decided to make the features accessible instead of the complete data set. In a way, the feature vectors are a proxy to the complete sentences. Because this no longer contains private information, these data can actually be made available (see GitHub). We also created a broader dataset out of the data by including the feature vector of the sentence before a repair, the type of repair and the sentence directly after a repair. This makes at least part of this study reproducible without revealing private information.

5.2 Future Work

The method used in this paper can be applied to other datasets, datasets in other languages or other 'proxies' from the same data set. So, for example other events that happen in the data can be selected, such as a chatbot conveying misunderstanding and returning options to the user (indicated by pre-programmed responses). Of course, the results might be different, but we suspect that some of the patterns will hold for example for other chatbots or other languages (such as the length of the utterances or typos).

This study focused completely on analysis of an existing data set. However, another factor that could be taken into account in the future is how users actually perceive these conversational breakdowns and if this influences their overall satisfaction and experience with the chatbot. In the current study we analyzed user utterance for potential cues for a breakdown, but we do not know how users actually perceive this breakdown and if they have any ideas why this breakdown happens. It would be interesting to know the perceptions of users that interact with a system that breaks down.

Even with the newer models (large language model (LLM) based bots) it could happen that the chatbot misunderstands the user. Wester et al. [48] show that there are still limitations to the capabilities of these large language model based chatbots, both in terms of technical limitations or policy restrictions. It remains relevant to study the causes of breakdowns and explain what went wrong, but the indicators of a possible breakdown are harder to distinguish in LLM conversations. In the future, research should focus on when and how LLM-bots break down, be it through automatic or through manual analysis.

5.3 Practical Implications

Can we actually prevent any mistake from happening, or should we learn to deal with them in an appropriate way? The analysis of the features showed that some of them are more indicative than others. This information can be used to customize repairs. When a word is for example common but not in domain (commonness) the chatbot could directly show that this word is potentially the problem. Customers appreciate repairs that actually acknowledge the error [1, 41]. Similarly, providing options and showing uncertainty has been shown to reduce the number of inadequate responses [13]. Work on LLMs shows that it matters how these models express uncertainty in terms of perceived trust, reliance and performance [27]. Similarly, Wester et al. [48] show that it matters how a LLM-based chatbot conveys its capabilities when denying a user help with their request.

Using feature analysis in a live-chat can therefore be seen as a viable solution to help guide the conversation. Some of the errors on the other hand (excess of information by for example copy pasting a complete webpage) might be more on the user side and could be prevented by educating the user on the capabilities of the bots. Yeh et al. [50] for example study how users can be guided into using chatbots based on guidance-types and guidance-timings, showing that each combination has its pros and cons.

Some of these errors in the analysis give us insights in what users expect the chatbot to do. There is inevitably a distance between the bot and the user in terms of what the user expects and what the system is actually capable of doing. Therefore, in designing these systems, there should be a focus on reducing this distance [28]. The features can thus be used by developers to optimize the system but also for customizing repairs during the conversation.

6 Conclusion

This study has shown the variety of user utterances that lead to a system-initiated repair. We set out to determine to what extent the selected features are useful as a signal of a conversational breakdown. Feature analysis shows that there are some features (such as commonness, outdated words, and unexpected words) that occur more often in utterances directly before a repair. Some features also correlate with each other and occur together, such as commonness and subjectivity. The method shows that simple features can indicate the source of a conversational breakdown in the user utterance. This method can be used by chatbot developers to detect the source of common errors that occur in chatbot conversations either live during the interaction or afterwards.

Acknowledgments. This study was funded by the Smooth Operators project (NWO; KIVI.2019.009). The authors would like to thank Christine Liebrecht for providing feedback.

Disclosure of Interests. There are no competing interests.

References

1. Ashktorab, Z., Jain, M., Liao, Q.V., Weisz, J.D.: Resilient chatbots: repair strategy preferences for conversational breakdowns. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2019)
2. Atuhurra, J., Kamigaito, H., Watanabe, T., Nichols, E.: Domain adaptation in intent classification systems: a review. arXiv preprint arXiv:2404.14415 (2024)
3. Bernard, N., Balog, K.: Identifying breakdowns in conversational recommender systems using user simulation. In: Proceedings of the 6th ACM Conference on Conversational User Interfaces, pp. 1–10 (2024)
4. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc., Sebastopol (2009)
5. Bohus, D., Rudnicky, A.I.: Sorry, I didn't catch that! An investigation of non-understanding errors and recovery strategies. In: Recent Trends in Discourse and Dialogue, pp. 123–154 (2008)
6. Borovits, N., Bardelloni, G., Tamburri, D.A., Van Den Heuvel, W.J.: Anonymization-as-a-service: the service center transcripts industrial case. In: International Conference on Service-Oriented Computing, pp. 261–275. Springer, Cham (2023)
7. Braggaar, A., Verhagen, J., Martijn, G., Liebrecht, C.: Conversational repair strategies to cope with errors and breakdowns in customer service chatbot conversations. In: Følstad, A., et al. (eds.) Chatbot Research and Design, pp. 23–41. Springer, Cham (2024)
8. Chhabra, S., Kaushal, V., Girija, S.: Determining the causes of user frustration in the case of conversational chatbots. Behav. Inf. Technol. 1–19 (2024)
9. Crolic, C., Thomaz, F., Hadi, R., Stephen, A.T.: Blame the bot: anthropomorphism and anger in customer-chatbot interactions. J. Mark. **86**(1), 132–148 (2022)
10. De Smedt, T., Daelemans, W.: Pattern for Python. J. Mach. Learn. Res. **13**(1), 2063–2067 (2012)
11. Eisenstein, J.: What to do about bad language on the internet. In: Vanderwende, L., Daumé III, H., Kirchhoff, K. (eds.) Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 359–369. Association for Computational Linguistics, Atlanta, Georgia (2013). https://aclanthology.org/N13-1037
12. Følstad, A., et al.: Future directions for chatbot research: an interdisciplinary research agenda. Computing **103**(12), 2915–2942 (2021)
13. Følstad, A., Taylor, C.: Conversational repair in chatbots for customer service: The effect of expressing uncertainty and suggesting alternatives. In: International Workshop on Chatbot Research and Design, pp. 201–214. Springer, Cham (2019)
14. Gatti, L., van Stegeren, J.: Improving Pattern.nl sentiment analysis. In: 30th Meeting on Computational Linguistics in The Netherlands, CLIN 2020 (2020)
15. van der Goot, M.J., Hafkamp, L., Dankfort, Z.: Customer service chatbots: a qualitative interview study into the communication journey of customers. In: International Workshop on Chatbot Research and Design, pp. 190–204. Springer, Cham (2020)
16. Grice, H.P.: Logic and conversation. In: Speech Acts, pp. 41–58. Brill (1975)
17. Hendriksen, M., Leeuwenberg, A., Moens, M.F.: LSTM for dialogue breakdown detection: exploration of different model types and word embeddings. In: Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems, pp. 443–453. Springer, Cham (2021)

18. Higashinaka, R., Araki, M., Tsukahara, H., Mizukami, M.: Integrated taxonomy of errors in chat-oriented dialogue systems. In: Li, H., et al. (eds.) Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 89–98. Association for Computational Linguistics, Singapore and Online (2021). https://doi.org/10.18653/v1/2021.sigdial-1.10, https://aclanthology.org/2021.sigdial-1.10
19. Higashinaka, R., et al.: Overview of the dialogue breakdown detection challenge 4. In: Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems, pp. 403–417. Springer, Cham (2021)
20. Higashinaka, R., Funakoshi, K., Araki, M., Tsukahara, H., Kobayashi, Y., Mizukami, M.: Towards taxonomy of errors in chat-oriented dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 87–95. Association for Computational Linguistics, Prague, Czech Republic (2015). https://doi.org/10.18653/v1/W15-4611, https://aclanthology.org/W15-4611
21. Higashinaka, R., Funakoshi, K., Kobayashi, Y., Inaba, M.: The dialogue breakdown detection challenge: task description, datasets, and evaluation metrics. In: Calzolari, N., et al. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 3146–3150. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016). https://aclanthology.org/L16-1502
22. Higashinaka, R., Mizukami, M., Funakoshi, K., Araki, M., Tsukahara, H., Kobayashi, Y.: Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2243–2248. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/D15-1268, https://aclanthology.org/D15-1268
23. Hirst, G., McRoy, S., Heeman, P., Edmonds, P., Horton, D.: Repairing conversational misunderstandings and non-understandings. Speech Commun. **15**(3), 213–229 (1994). https://doi.org/10.1016/0167-6393(94)90073-6, https://www.sciencedirect.com/science/article/pii/0167639394900736, special issue on Spoken dialogue
24. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. 65–70 (1979)
25. Jakobovits, A.S., Piccinno, F., Altun, Y.: What did you say? Task-oriented dialog datasets are not conversational!? arXiv preprint arXiv:2203.03431 (2022)
26. Keuleers, E., Stevens, M., Mandera, P., Brysbaert, M.: Word knowledge in the crowd: measuring vocabulary size and word prevalence in a massive online experiment. Q. J. Exp. Psychol. **68**(8), 1665–1692 (2015). https://doi.org/10.1080/17470218.2015.1022560, pMID: 25715025
27. Kim, S.S., Liao, Q.V., Vorvoreanu, M., Ballard, S., Vaughan, J.W.: "I'm not sure, but...": examining the impact of large language models' uncertainty expression on user reliance and trust. In: The 2024 ACM Conference on Fairness, Accountability, and Transparency, pp. 822–835 (2024)
28. Kocaballi, A.B., Coiera, E., Berkovsky, S.: Revisiting habitability in conversational systems. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–8 (2020)
29. Kunneman, F., Van den Bosch, A.: Open-domain extraction of future events from twitter. Nat. Lang. Eng. **22**(5), 655–686 (2016). https://doi.org/10.1017/S1351324916000036

30. Kvale, K., Sell, O.A., Hodnebrog, S., Følstad, A.: Improving conversations: lessons learnt from manual analysis of chatbot dialogues. In: Følstad, A., et al. (eds.) Chatbot Research and Design, pp. 187–200. Springer, Cham (2020)
31. Luo, B., Lau, R.Y., Li, C., Si, Y.W.: A critical review of state-of-the-art chatbot designs and applications. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **12**(1), e1434 (2022)
32. Maat, H.P., et al.: T-scan: a new tool for analyzing Dutch text. Comput. Linguist. Netherlands J. **4**, 53–74 (2014)
33. Mackey, E.: A best practice approach to anonymization. In: Handbook of Research Ethics and Scientific Integrity, pp. 323–343 (2020)
34. Martinovsky, B., Traum, D.: The error is the clue: breakdown in human-machine interaction. In: Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems, pp. 11–17 (2003)
35. McInnes, L., Healy, J., Astels, S.: HDBScan: hierarchical density based clustering. J. Open Source Softw. **2**(11) (2017). https://doi.org/10.21105/joss.00205
36. Meij, E., Weerkamp, W., De Rijke, M.: Adding semantics to microblog posts. In: Proceedings of the fifth ACM International Conference on Web Search and Data Mining, pp. 563–572 (2012)
37. Möller, S., Engelbrecht, K.P., Oulasvirta, A.: Analysis of communication failures for spoken dialogue systems. In: Eighth Annual Conference of the International Speech Communication Association (2007)
38. Naber, D., et al.: A rule-based style and grammar checker (2003)
39. Paek, T.: Toward a taxonomy of communication errors. In: ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems (2003)
40. Reinkemeier, F., Gnewuch, U.: Designing effective conversational repair strategies for chatbots. In: ECIS (2022)
41. Rheu, M., Shin, J.Y., Peng, W., Huh-Yoo, J.: Systematic review: trust-building factors and implications for conversational agent design. Int. J. Hum.-Comput. Interact. **37**(1), 81–96 (2021)
42. Sanguinetti, M., et al.: Annotating errors and emotions in human-chatbot interactions in Italian. In: The 14th Linguistic Annotation Workshop, pp. 1–12. Association for Computational Linguistics (2020)
43. Seabold, S., Perktold, J.: Statsmodels: econometric and statistical modeling with Python. In: 9th Python in Science Conference (2010)
44. Sugiyama, H.: Dialogue Breakdown Detection Using BERT with Traditional Dialogue Features, pp. 419–427. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9323-9_39
45. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
46. de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., Noord, G.v., Nissim, M.: BERTje: a Dutch BERT model. arXiv:1912.09582 (2019). http://arxiv.org/abs/1912.09582
47. Weitzenboeck, E.M., Lison, P., Cyndecka, M., Langford, M.: The GDPR and unstructured data: Is anonymization possible? Int. Data Privacy Law **12**(3), 184–206 (2022)
48. Wester, J., Schrills, T., Pohl, H., van Berkel, N.: "As an AI language model, I cannot": investigating LLM denials of user requests. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 1–14 (2024)
49. Y.digital/Direct Research: Nationale voice monitor 2024 (2024). https://www.y.digital/resources/nationale-voice-monitor-2024, y.digital

50. Yeh, S.F., et al.: How to guide task-oriented chatbot users, and when: a mixed-methods study of combinations of chatbot guidance types and timings. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp. 1–16 (2022)
51. Zumstein, D., Hundertmark, S.: Communicating and transacting with chatbots: insights from public transport. In: Proceedings of the International Conference on WWW/Internet 2017 and Applied Computing, December 2017, pp. 55–62 (2017)