# MCCD: Generating Human Natural Language Conversational Datasets

Matheus F. Sanches[1], Jader M. C. de Sá[1], Allan M. de Souza[1], Diego A. Silva[2], Rafael R. de Souza[1], Julio C. dos Reis[1] and Leandro A. Villas[1]

[1]*Institute of Computing, University of Campinas, São Paulo, Brazil*

[2]

Keywords: Natural Language Processing, Data Wrangling, Data Acquisition, Human Conversation, Model Learning, Tool.

Abstract: In recent years, state-of-the-art problems related to Natural Language Processing (NLP) have been extensively explored. This includes better models for text generation and text understanding. These solutions depend highly on data to training models, such as dialogues. The limitations imposed by the lack of data in a specific language significantly limit the available datasets. This becomes worse as intensive data is required to achieve specific solutions for a particular domain. This investigation proposes *MCCD*, a methodology to extract human conversational datasets based on several data sources. MCCD identifies different answers to the same message differentiating various conversation flows. This enables the resulting dataset to be used in more applications. Datasets generated by MCCD can train models for different purposes, such as Questions & Answers (QA) and open-domain conversational agents. We developed a complete software tool to implement and evaluate our proposal. We applied our solution to extract human conversations from two datasets in Portuguese language.

## 1 INTRODUCTION

Pre-trained models are considered the backbone of several modern NLP systems, as they are one of the most prominent models at the moment (Qiu et al., 2020). These models often rely on large amounts of data to be trained (Qiu et al., 2020). Furthermore, data used during the training stage directly affects the quality of models and the relations and biases learned. In this sense, high-quality datasets are essential to developing human-like NLP systems (Bansal et al., 1993) (Mehrabi et al., 2021).

Some of the largest datasets are available primarily in English. They tend to be less moderated and may present negative biases as they grow. The lack of moderation occurs due to high costs and the time needed to be performed. The use of moderated data sources may significantly increase the quality of the final dataset. In this context, it lacks adequate methodologies to create conversational datasets to allow standardization of several stages performed during the identification of conversations and generation of a novel dataset.

State of the art language models led to remarkable progress in NLP tasks (Brown et al., 2020). However, given the challenges of obtaining massive datasets of unlabeled text from the Web, there is still a problem in creating conversational applications relying on such models. Despite the importance of data for models, it is hard to find complete datasets in multiple languages simultaneously. The Web is an essential source of texts as several websites are available in multiple languages, where there are many sites in different languages. However, in several languages, such as Portuguese, the total volume of data ready to be used is limited. If we add more constraints, such as informal speech or conversations with at least three turns, there is no available conversational dataset of such nature.

Creating conversation datasets from Web sources is a challenge because usually there is more than one way to answer each message, regardless of its language. We observed a gap in the literature regarding solutions to create conversational datasets. It affects consumers of pre-trained language models because it is unknown the influences of pre-training data on their systems. In this context, our investigation addresses the following research question: **how to create conversational datasets using existing Web data in place?** This includes data from dialogues with different language aspects, *i.e.*, formal, informal, slang, and internet abbreviations. These are essential to training models with different ways to understand



Sanches, M., C. de Sá, J., M. de Souza, A., Silva, D., R. de Souza, R., Reis, J. and Villas, L.
MCCD: Generating Human Natural Language Conversational Datasets.
DOI: 10.5220/0011077400003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 247-255
ISBN: 978-989-758-569-2; ISSN: 2184-4992

messages and communicate with distinct audiences.

This article proposes a new source-agnostic methodology for generating conversational datasets, called **Methodology for Creating Conversational Datasets (MCCD)**. The results obtained through our methodology allow the creation of a unique dataset. On this basis, it is possible to train models with language modeling techniques or even segment the interaction between two or more humans to train models with different answers to the same questions.

In our solution, we implemented a software tool that instantiates the methodology. The software is fed with data sources from online Web forums and automatically generates conversational datasets. We conducted a case study in which the software tool was used to acquire, anonymize, clean and identify conversations from data obtained from Web forums. Our study implies three main contributions: A novel methodology to generate conversation datasets in an automatic way; a developed completed software tool for mining data from online forums (implementing the methodology); and two completed datasets in Portuguese language generated from the application of our tool. We centralized the available tool and datasets into a single GitHub repository[1].

The remaining of this article is organized as follows: Section 2 presents the background, including relevant concepts and related studies. Section 3 presents the proposed methodology. Section 4 presents the tool implemented using the proposed methodology. Section 5 presents two datasets generated with the tool presented. Section 6 discusses the achievements, contributions, and limitations of this work. Section 7 concludes this work.

## 2 BACKGROUND

Several natural language applications rely on textual data for training and execution. We consider three main applications to use our generated datasets from the application of our proposal: language modeling, word embedding, and dialogue agents.

Language modeling is the process of predicting the chance of a specific sequence of words appearing in a determined sentence. Natural Language (NL) models that generate text as output perform language modeling as part of the training phase. Some state-of-the-art NLP models were trained using only language modeling techniques, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) and Generative Pre-Training Transformer 2 (GPT-2) (Radford et al., 2019). Although the process was slightly different between these two models, the goal of predicting a word was similar. BERT was fed with a sequence of words, with 15% of the words masked, and it should output the correct sequence without masked words. GPT-2 was fed with a sequence of words and predicted the next word to that sequence.

Word embedding represents words and phrases digitally, as a list of numbers. There are different techniques to create this representation, each with advantages and disadvantages. One of the most important is ELMo (Peters et al., 2018), where the representation depends on the context, which means that the same word may have a different representation according to the context being used. ELMo achieves state-of-the-art results in different tasks, such as sentiment classification and QA.

Dialogue agents are applications designed to interact with real users, having conversations with natural language. This application faces a variety of challenges while interacting with real users. For instance, during conversations, humans may refer to out-of-domain concepts. They can also use metaphors, irony, or rely on interlocutors, common sense, or general knowledge. However, in a dialogue, both parties need to be active in many turns, and it is context-dependent: concepts introduced at the beginning of a conversation may be referred to in different moments (Traum, 1999).

The data used has a crucial role in the three NL applications presented. Data quality directly influences the results as these applications require data to train and refine their underlying classification models. In the following, we present a list of a few problems that may occur in this context.

- **Language Modeling:** Trained models using data with problems can insert bias into the model (Wolf et al., 2017);
- **Word Embedding:** This technique may create similar representations for words that do not have any common characteristics;
- **Dialogue Agents:** If this application is trained on low-quality data, agents may not be able to deal with different situations or can incorporate problems related to language modeling or even word embedding;

A large amount of the available written text, such as the articles at Wikipedia, can be used to perform Language Modeling and Word Embedding. However, this text may not be enough to train a Dialogue Agent. In order to tackle this, we create conversation datasets.

[1] https://github.com/MatheusFerraroni/MCCD

Conversational datasets are composed of data about text exchanged between two or more agents, where ideally, these agents are humans. Conversations are composed of turns, where one agent responds to the other. Each response considers the latest messages, the conversation context, and even the main topic related to the data source.

We conducted an exploratory literature review to reach key studies correlated to our investigation. One of the central related studies found was the dataset Brazilian Portuguese Web as Corpus (BrWaC) (Wagner Filho et al., 2018). This dataset was constructed with a crawler acquiring different web pages in Portuguese. Although this is one of the largest datasets in Portuguese, there are important characteristics to be taken into account to use this dataset, such as its data varying from highly informal to highly formal and opinion pages, which may create bias to models using it. The BrWaC dataset lacks conversational data, as most of its pages are only informative pages, such as news and sales. Similar, there is Common Crawl (Smith et al., 2013), which is a dataset about web pages that is constantly updated. Due to the similarity with the BrWaC, the Common Crawl faces the same challenges and limitations.

The ubuntu dialogue corpus(Lowe et al., 2015) is an English dataset based on conversations obtained through a crawler used on operating systems forums. This dataset was used to construct a multi-turn conversational dataset. To prove their hypotheses about using the dataset, they used an Recurrent Neural Network (RNN) to select the best response for each turn. This dataset was complete enough to present good results in the test scenarios. Nevertheless, it presents a strong bias about technology theme questions and may need a large fine-tune processing or a complementary dataset during training to be used in other domains. This dataset is a Human-to-Human (H2H) dataset.

The MultiWOZ dataset (Budzianowski et al., 2018) is a popular English dataset that imitates a conversation between a user talking to a virtual assistant, where the user requests information about places and services, and requests the virtual assistant to book reservations at different places and times. This dataset follows the Wizard-of-Oz approach (Kelley, 1984) and uses crowd workers to construct the task-oriented dialog. Although this dataset is only 10k dialogues, it is fully annotated, significantly increasing its relevance. This dataset is a Human-to-Machine (H2M) synthetic dataset.

Corporations are maintaining datasets and competitions to promote the development of specific areas. Microsoft is currently maintaining a competition on Task-Oriented Dialog Systems, which periodically releases machine-generated dialogues (Williams et al., 2016). The main goal of this challenge is to keep track dialogue state at each turn, such as the user's goal. The data used in this challenge is also H2M.

Li *et al.* (Li et al., 2017) created a multi-turn dialog about daily life for everyday conversations in English. This dataset was created artificially by humans writing the conversations; the conversations cover a large variety of areas, such as sports, weather, mood, transportation, and more. This dataset simulated a conversation H2M and was artificially created. Similarly, Byrne *et al.* (Byrne et al., 2019) created a similar dataset, but for six specific domains.

Although there are different datasets available, we found limitations to where each dataset can be used. This is due to different limitations, such as language, variables included, size, processing aspects, and more. In our study, we tackle specifically the lack of conversational H2H datasets, in which the ubuntu dialogue corpus is the closest one. The main limitation of this dataset is how the message references work in the data source, which results in the lack of different answers to the same question. Our MCCD methodology (cf. Section 3) defines how to select data sources with proper message references to create complete conversation datasets with multiple answers to the same message. Table 1 presents an example of multiple messages replying to the same message.

Table 1: Example of multiple answers to the same message.

| ID | Answering | Content |
|---|---|---|
| 1 | - | What is the best route to go from A to B? |
| 2 | 1 | The fastest way is through a straight line. |
| 3 | 1 | You can go straight for a few blocks and turn right. |
| 4 | 2 | Than you! I will follow your suggestion |

## 3 MCCD METHODOLOGY

We present our proposed methodology with the designed stages to create conversational datasets with all possible conversation flows. The identification of conversation flow is the process of identifying all ramifications an online conversation may have based on the message references. Different conversation flows may start and finish with the same messages, but the messages exchanged are different. This allows us to create a QA dataset.

MCCD was designed to use any data source during the data acquisition stage. As a result of following each stage in our methodology, the final output contemplates a specific data structured, organized, and cleaned in a specific way to allow future researchers to use it for training NLP models.

The main goal of our methodology is the generation of conversational datasets. The results generated include a clear, structured, and chronologically ordered sequence of text messages exchanged among users who wrote online forum messages.

We defined how to structure messages among users from the data source (Web forum) under processing. We described recommendations to be executed during the cleaning process of messages, suggesting replacements for specific situations presented in texts. This includes identifying images or emojis and overwriting them with pre-defined tags.

Our methodology with the proposed stages applies to any language, regardless of the used character set, as long as the storage method chosen during implementation allows it. To this end, requirements need to be fulfilled for the processing stage to reach very rich datasets.

We designed the methodology to have its requirements as simple as possible. Their primary purpose is to ensure that the resulting dataset contains the writing marks from a data source, such as formal or informal writing. The solution must be suited to identify a conversation's possible flows. We present the requirements in the following:

- **Human Requirement:** The textual data obtained must have been written by humans talking to each other or answering a specific topic (open issue).
- **References between Messages:** The step of conversation identification in the processing stage requires the messages in the data source to reference zero or more messages as replying.

The human-to-human textual data ensures that the generated dataset contains specific characteristics that humans insert in writing texts by using Web systems, such as emojis while using smartphone apps or slangs in online forums. This kind of element allows the resulting dataset to contain informal language modeling.

As the cleaning process changes the final results, the generation of the final dataset must be accompanied by documentation that details how the data cleaning was performed. In this document, the replaced tags, removed elements, or any other transformation done in the data must be described, as already defined in the literature. (Gebru et al., 2021).

Figure 1 presents our methodology's designed flow of tasks organized into three major stages. Our methodology was designed to be as straightforward as possible, considering the complexity of the acquired and generated data.

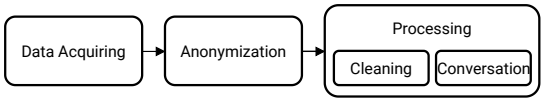


Figure 1: Designed stages in MCCD.

The **first** stage is data acquiring, in which the data source is accessed to retrieve the needed data (cf. Subsection 3.1).

The **second** stage is the Anonymization stage (cf. Subsection 3.2). At this stage, all data used to identify a user is removed.

The **last stage** is the processing (cf. Subsection 3.3). This stage is separated into two steps: cleaning and conversation identification. The cleaning step is responsible for removing or replacing elements from the data preparing it to be used. The conversation identification step identifies and saves the conversations and conversation flows. The conversation identification step may generate a large amount of data to be stored.

Once the three stages are completed, the generated files are cleaned, structured, and ready for use.

## 3.1 Data Acquisition Stage

Our methodology is source agnostic, which allows it to be used in further scenarios, as long as data fulfill the basic requirements mentioned.

The only required information for the data acquisition is a parameter indicating where the data is located. This stage is responsible for understanding how to access the data and how they are structured in the source. It may require the information of additional parameters, indicating whether data may be accessed using threads to speed up this stage or even credentials to access specific data sources.

Figure 2 presents the steps performed inside this stage.

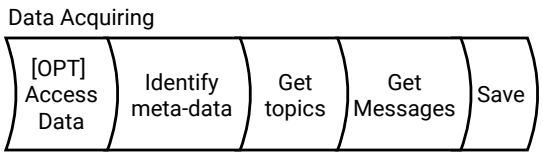


Figure 2: Steps in the Data Acquiring stage.

The **first** step is to access the data being. This methodology does not define how this access is made as long as the requirements are satisfied. The **second** step is the identification of categories, sub-categories, and meta-data from the data being acquired, such as

timestamps and structure.

The **third** step aims to get the topics from each sub-category. If the data is being acquired directly from a database, a single access for each category may already gather all information needed.

The **fourth** step gets all messages from a specific topic. This may encounter the same requirements as the previous step. The **last step** is responsible for persist the data acquired. Although this step is the last in the flow, it can save partial results while the data is acquired for each stage. Each file saves information about a specific item, such as a single topic and its message or a single sub-category and its topics.

This stage validates if the output directory already has a valid dataset from the same source. If there are old files from a previous acquisition, this step loads the old dataset and only searches for new data in the data source.

## 3.2 Anonymization Stage

This stage is responsible for removing all meta-data that may identify a user in the acquired dataset. The location and the amount of data that need to be anonymized may vary according to the data source and which data is saved in the acquisition stage.

This stage overwrites the original files, replacing the values with an empty value or setting an *id* to keep track of which user created messages, but without identifying this user outside the dataset. It is important to note that this stage is mandatory if the final dataset is released publicly.

## 3.3 Processing Stage

The processing stage encompasses the key steps in the methodology. Figure 3 shows the steps to complete the entire processing stage. As a result of this stage, files are generated, one for each topic and one for each conversation between two or more persons inside each topic.

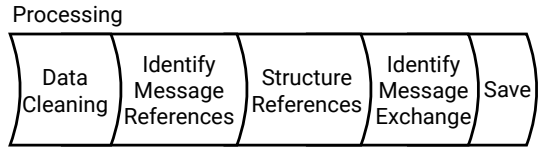


Figure 3: Steps in the processing stage.

Each stage depends on previous stages (cf. Figure 3), as it is impossible to identify the message exchange if the references were not identified. The final result can be severely degraded if the data cleaning is not completed correctly. The input for the processing stage is either the output from the acquisition stage or the output from the anonymization stage.

The step of data cleaning is responsible for identifying elements that should not be presented in a conversation dataset, such as images and videos. Once one of these elements is identified, it is removed or replaced with proper tags to indicate what was in that place before. This step must be utilized only on text messages.

The quality of this step directly impacts the final result because poorly cleaned texts may decrease the quality of the models using it. One example of this situation is to use a collection of PDFs as the data source. The images from the PDFs can be replaced with a particular tag indicating the presence of an image. It is also possible to replace tables with the text separated by a colon character.

The elements replaced or removed are changed according to the data source and must be explicit at the final documentation of the dataset generated. The last step creates the messages for a specific topic with the messages in chronological order.

Each conversation in the generated dataset from our solution is duly identified, and they are organized so that each different response to a message generates different conversation flows. This process allows identifying different responses to the same message; this happens in the real world, where there are many ways to answer the same question.

Figure 4 presents how the identifications of multiple conversation flows happens. The messages are organized in chronological order, wherein the instant $t0$ – the message 1 was created; and the instant $t4$ – the message 5 was created. Each message may reference 0 or more messages. In our example, message 4 references messages 3, 2, and 1.

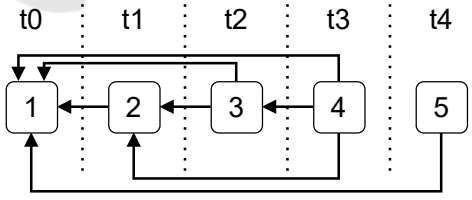


Figure 4: References between messages.

Table 2 shows the references in the example presented in Figure 4.

The process starts backward from the last message. In our example from Figure 4, the message number 5. For each reference, a recursive call occurs until it finds a message that does not reference any other message. Once the process finds this message, the solution returns to the initial message, creating one possible flow starting from message 5. This pro-

Table 2: References between messages.

| Message | References to |
|---|---|
| 1 | None |
| 2 | 1 |
| 3 | 2, 1 |
| 4 | 3, 2, 1 |
| 5 | 1 |

cess is similar to building a tree data structure, where the root node is the most recent message, and each way to a leaf node is a conversation flow. Algorithm 1 shows a pseudo implementation of how of our conversation flow identification solution.

Algorithm 1: Conversation flow identification.

```
procedure CONVERSATION_FLOWS(message)
    references ← get_references(message)
    for ref ← references do
      flow ← Conversation_Flows(ref)
    end for
    if references is empty then
      res ← []
        while message.parent is not empty do
      res.add(message)
      message ← message.parent
        end while
      return res
    end if
end procedure
```

## 3.4 Final Output

This section describes the final output generated by the methodology and the data that need to be persisted. The methodology MCCD does not define that the data must be saved with individual files, the character encoding, or extension. In order words, our methodology only explicitly defines the relevant be saved.

**Category data:** All data relating to the category and sub-category must be stored, such as category and sub-category list, names, creation time, and the number of topics.

**Topic data:** All topics identified and information about it must be stored. The content of each topic is saved separated.

**Messages/User Texts:** The raw content of all messages inside a topic must be stored. This includes at least the written text of the message, references, and creation time

**Clear texts:** The cleaned messages must be saved separated. This data allows the training of novel models.

**Conversations:** Each conversation flow identified must be saved, cleaned, and chronologically.

# 4 MINER-XenForo SOFTWARE TOOL

We implemented a complete software tool called *Miner-XenForo* that implements our proposed methodology. In addition, it specifies and implements functionalities to a specific domain. The *Miner-XenForo* used as data source Web forums that are constructed using *XenForo*[2], a platform for managing online forums. Our software tool gathers data from websites of online forums such that the acquisition step working refers to a web scraper.

Figure 5 shows the modules and execution flow of the software tool. The label "*R*" indicates the presence of requests to websites. The label "*OS Threads*" indicates the use of operating system threads. The Data Acquisition stage is performed with four steps such that each one is responsible for a specific task defined in the MCCD methodology.

The Anonymization stage is performed by a single module without using system threads but prepared to work with it. The Processing stage utilizes system threads. According to the dataset under treatment, this stage can be the slowest one (in terms of processing time). Four modules were created to complete the Processing stage. These modules run sequentially while the processing for each topic runs in parallel.

## 4.1 Miner-XenForo: Data Acquisiton

The acquisition stage is designed to acquire data from a website built using *XenForo*. The only required input regarding how to access the data source is the URL where the forum is hosted. The List below presents the structure of such forums.

1. Main Page
   (a) List with categories. (Reference to 2)
   (b) List with sub-categories. (Reference to 3)
2. Category
   (a) List with sub-categories. (Reference to 3)
3. Sub-category
   (a) List with topics. (Reference to 4)
4. Topic (Also named as "thread" by some forums).
   (a) List with messages.

[2]https://xenforo.com/

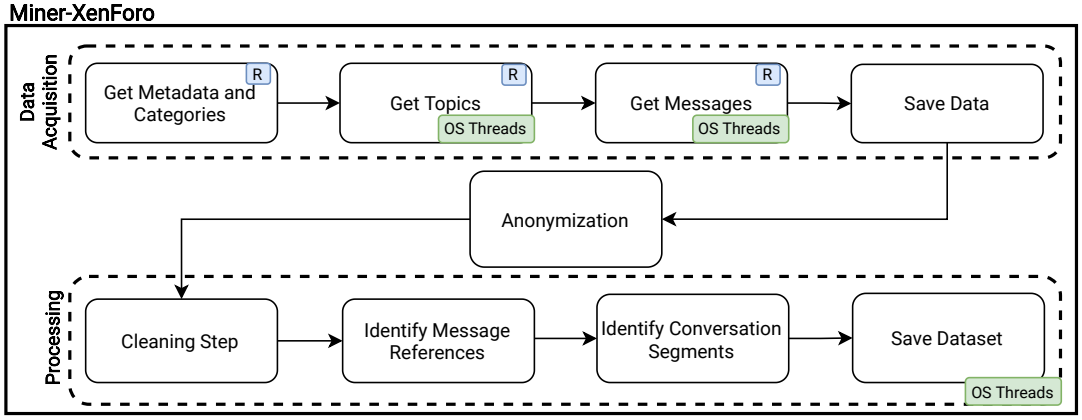


Figure 5: *Miner-XenForo* Software tool Modules.

In addition, we prepared the implementation to deal with the following optional parameters:

- **reload-threads:** Iterate each page inside each sub-category to search for new topics.
- **reload-posts:** Iterate over each topic page, searching for new messages.
- **max-request:** Set how many simultaneous requests the software can do to the website. Caution is required as the website can prevent multiple requests quickly.
- **cache-pages:** Cache mechanism to debug and adapt the acquisition process. Must not be used in production as the amount of data generated increases rapidly

In the acquisition, the software performs a single request to the main page to obtain a list of categories and sub-categories available. Afterward, the *Miner-XenForo* checks the local files to determine if a new dataset is being created or updating an existing one while acquiring information about the available topics and messages. During this process, our software tool checks the local files to update older versions of the same dataset.

## 4.2 Miner-XenForo: Anonymization

This module was implemented following all the requirements from the methodology. If this module is executed right after the acquisition stage, all results generated from examining or using the dataset are already anonymized. This is possible due to the overwriting of the original un-anonymized files with the ones generated by this module.

Due to the structure generated from the acquisition module, two types of files need to be anonymized: data about topics and data about the messages. These files contain the user who created each topic and each message in the dataset.

The JSON files containing unprocessed versions of topics and sub-categories may contain fields that the values identify a user. Due to this, the JSON files about sub-categories have the fields “user_href” and “user_name” replaced with an empty string. The JSON files with information about topics, the field “member_href” is replaced with an empty string, and the field “member_name” is replaced with a unique id for this user. This unique *id* used allows identifying all messages that this user has created without letting information about who the user is.

## 4.3 Miner-XenForo: Processing

The processing stage was specified to work with data acquired from the data source used during the acquisition stage. In this sense, in particular, the cleaning process is prepared to deal with HTML elements and specific elements presented in Web forums made using XenForo. All the elements being replaced, removed, or changed during the cleaning process can be viewed at the *Mine-XenForo* documentation.

The identification of conversation flows as close as possible from the original methodology specification. It differs only at ignoring references to messages in the future. This situation may happen as XenForo allows users to edit and cite messages created after theirs. This action needs to be done to keep the generated results consistent, as answering not already created messages may be a problem. This situation creates loops during processing that would require a specific approach.

# 5 CASE STUDY: DATASETS GENERATION

Our Software tool *Miner-XenForo* was applied and used to create two huge conversational datasets in Portuguese language.

The first one, called *"Adrenaline Dataset"*, used a website[3] about technology, hardware, and games. Our generated dataset based on the execution of *Miner-XenForo* reached a compressed file with about 2.0Gb of data; and a compressed file with the processed files is more than 70Gb. This dataset has more than 356K topics and 9.5M messages.

The second generated dataset was the *"OuterSpace Dataset"*. We applied *Miner-XenForo* to a Web forum[4] with content about games, generic themes, and a buy/sell category. As a result, the compressed version is 4.6Gb, and the compressed file with processed files is 5.4Gb. This dataset has more than 570K topics with more than 24M messages.

Figure 6 compares the number of messages per topic in the "*Adrenaline dataset*" and the "*OuterSpace dataset*".

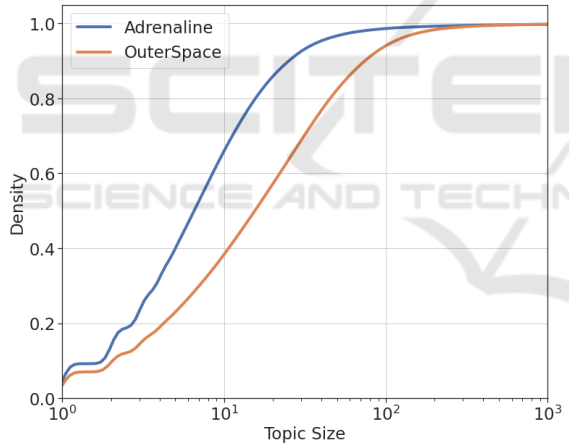


Figure 6: Topic size in Adrenaline and OuterSpace datasets.

We observed the same behavior regarding the size of topics in both datasets. We found most topics with less than ten messages and few topics with more than one hundred. About 9% of all topics in the "Adrenaline Dataset" have one message; this is 7% in the "OuterSpace dataset". Furthermore, the majority of the topics in both datasets have less than 100 messages. In both datasets, 99.9% of the topics have at most 1000 messages, but the largest topic in the "Adrenaline dataset" has more than 134K messages; and in the "OuterSpace dataset" has more than 324K messages.

Table 3: Messages density distribution.

| Max of # Messages | Adrenaline | OuterSpace |
|---|---|---|
| 1 | 9.2% | 7.0% |
| 10 | 67.7% | 39.5% |
| 100 | 98.6% | 94.1% |
| 1000 | 99.7% | 99.7% |

Both data sources utilize the same web forum platform, and they were acquired and processed with the *Miner-XenForo* software. However, the interaction among the users occurring on them was different enough to generate 13 times more data on the "Adrenaline dataset" compared to the "OuterSpace dataset".

# 6 DISCUSSION

This investigation contributed in the definition and evaluation of a methodology capable of structuring the requirements, stages, and steps as a way to develop software tools to extract conversational datasets. Furthermore, we implemented a complete software tool following our developed methodology MCCD, called Miner-XenForo. This tool was used to create two different conversational datasets already available publicly.

MCCD was created to be used in any data source that fulfills the defined requirements. In particular, our use case software tool, Miner-XenForo, was implemented to acquire data from forums built on top of XenForo. Despite this limitation, our obtained software tool covers many forums in different languages and MCCD may be applied to any language.

The contribution obtained in this investigation paves the way for the creation of novel datasets that can be used to train NLP models in different languages.

# 7 CONCLUSION

The generation of adequate conversational datasets from available Web data sources for model learning is still an open research challenge. This is even true and required for languages in addition to English, such as Portuguese. This study proposed a novel source agnostic methodology to generate conversational datasets automatically. On this basis, we implemented a complete software tool to acquire, anonymize, clean and identify conversations from

[3] forum.adrenaline.com.br
[4] forum.outerspace.com.br

forum-like online data sources. This study applied our software tool to generate two conversational datasets in Portuguese language. The obtained datasets from our study were made available for sharing and reuse purposes. We found that our designed methodology and software tool enables the creation of relevant datasets. This must leverage the training of language models. Indeed, our subsequent research steps involve using the generated conversational datasets to train new models and refine existing language models. We aim to apply and evaluate such models in constructing chatbots for customer services.

## REFERENCES


Bansal, A., Kauffman, R. J., and Weitz, R. R. (1993). Comparing the modeling performance of regression and neural networks as data quality varies: A business value approach. *Journal of Management Information Systems*, 10(1):11–32.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T. J., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018). Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Byrne, B., Krishnamoorthi, K., Sankar, C., Neelakantan, A., Goodrich, B., Duckworth, D., Yavuz, S., Dubey, A., Kim, K.-Y., and Cedilnik, A. (2019). Taskmaster-1: Toward a realistic and diverse dialog dataset. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Trans. Inf. Syst.*, 2(1):26–41.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. In *OpenAI*.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt cheap web-scale parallel text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria. Association for Computational Linguistics.

Traum, D. R. (1999). *Speech Acts for Dialogue Agents*, pages 169–201. Springer Netherlands, Dordrecht.

Wagner Filho, J. A., Wilkens, R., Idiart, M., and Villavicencio, A. (2018). The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Williams, J. D., Raux, A., and Henderson, M. (2016). The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Wolf, M., Miller, K., and Grodzinsky, F. (2017). Why we should have seen that coming: Comments on microsoft's tay "experiment," and wider implications. *The ORBIT Journal*, 1(2):1–12.