Text-to-SQL Oriented to the Process Mining Domain: A PT-EN Dataset for Query Translation

Bruno Y. Yamate[†], Thais R. Neubauer[†], Marcelo Fantinato[†], Sarajane M. Peres[†]

[*]Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, Av. Arlindo Béttio, 1000, São Paulo, 03828-000, São Paulo, Brazil.

[*]Corresponding author(s). E-mail(s): brunoyui@usp.br; sarajane@usp.br;
Contributing authors: thais.neubauer@usp.br; m.fantinato@usp.br;
[†]These authors contributed equally to this work.

Abstract

This paper introduces *text*$_{\mathbf{2}}$*SQL*$_{\mathbf{4}}$*PM*, a bilingual (Portuguese-English) benchmark dataset designed for the text-to-SQL task in the process mining domain. Text-to-SQL conversion facilitates natural language querying of databases, increasing accessibility for users without SQL expertise and productivity for those that are experts. The *text*$_{\mathbf{2}}$*SQL*$_{\mathbf{4}}$*PM* dataset is customized to address the unique challenges of process mining, including specialized vocabularies and single-table relational structures derived from event logs. The dataset comprises 1,655 natural language utterances, including human-generated paraphrases, 205 SQL statements, and ten qualifiers. Methods include manual curation by experts, professional translations, and a detailed annotation process to enable nuanced analyses of task complexity. Additionally, a baseline study using GPT-3.5 Turbo demonstrates the feasibility and utility of the dataset for text-to-SQL applications. The results show that *text*$_{\mathbf{2}}$*SQL*$_{\mathbf{4}}$*PM* supports evaluation of text-to-SQL implementations, offering broader applicability for semantic parsing and other natural language processing tasks.

Keywords: Dataset Benchmark, Text-to-SQL, Structured Query Language, Process Mining, Large Language Models, Prompt Engineering

1 Introduction

The text-to-SQL conversion task [1], a specialized area of semantic parsing, involves generating SQL (Structured Query Language) statements from natural language utterances. By enabling access to database information through natural language holds the potential to democratize information retrieval, allowing users without knowledge of SQL commands or syntax to make basic queries [**?**]. Furthermore, solving the text-to-SQL task through semantic parsing enhances developer productivity by generating SQL statements that closely align with ideal ones. Recent advances in implementing such a task have leveraged modern deep neural networks [2], either complementing or replacing traditional rule-based parsing and mapping techniques. Even more recently, large language models (LLMs), coupled with diverse prompt engineering strategies, have demonstrated remarkable performance on benchmark datasets for the task [3]. Benchmark datasets used to evaluate solutions in this area are typically cross-domain and comprise multi-relational databases [3–5]. These datasets serve a dual purpose: they are employed both for model fine-tuning and for assessing the effectiveness of text-to-SQL strategies.

Although a range of domains is addressed in these benchmark datasets, they typically represent classical information retrieval domains such as cars, flights, music, sports, academia, and pets. While this diversity of domains provides valuable vocabulary and insights into information retrieval requirements for model training and evaluation, it is not comprehensive enough to develop models that perform effectively in more specialized domains. These domains often feature data structures with characteristics uncommon in relational databases and possess specific needs for information extraction. In this paper, we introduce a benchmark dataset designed to support information retrieval tasks relevant to the process mining domain. The study of applying the text-to-SQL task in process mining is motivated by the strong interdisciplinarity inherent to this field. Data analysts, process analysts, and organizational managers all play crucial roles in the practical implementation of process mining. By considering these professionals, we highlight the benefits of text-to-SQL solutions in two key areas: enhancing productivity in information retrieval for data analysts and improving accessibility to information for process analysts and organizational managers.

In the process mining domain, the data of interest is stored in event logs, where each record corresponds to process executions [6, 7]. These logs serve as a foundation for extracting valuable insights, whether through specialized algorithms, process query languages, or more commonly used computational strategies such as SQL and descriptive statistics. In this context, while event logs are not inherently required to be stored in a relational database, they can be transformed for SQL-based information retrieval. The most common standard for storing event logs in process mining is the XES (eXtensible Event Stream) format [8], which, when converted for use in a relational database, generates a single, non-normalized table. This lack of normalization, combined with the specialized vocabulary and unique information needs of process mining, creates a domain in which text-to-SQL strategies tend to underperform when compared to classical domains. Despite the apparent simplicity of a context where information retrieval is performed via SQL from a single table in a relational

database, exploratory studies have shown that querying information in this scenario can be quite challenging.

The benchmark dataset $text_2SQL_4PM$ is introduced in this paper. $text_2SQL_4PM$ is an open, bilingual (Portuguese and English), and annotated benchmark dataset, designed for training and evaluating text-to-SQL solutions within the specific context of process mining. To the best of our knowledge, no existing dataset includes natural language utterances, SQL statements, and annotations tailored to the process mining domain. Thus, we assert that the contributions of this paper are:

1. A complete description of the proposed bilingual (Portuguese-English) benchmark dataset with 1,655 natural language utterances, 205 corresponding SQL statements, and ten qualifiers[1]. The dataset includes 205 distinct utterances and 1,450 paraphrases, all formulated by humans (without the use of language models in the construction of the utterances or SQL statements). The qualifiers are employed to facilitate a contextualized analysis of the complexity embedded in the dataset.
2. An example of applying a text-to-SQL solution based on prompt engineering and the large language model GPT-3.5 Turbo, accompanied by a detailed and contextual analysis of the results with respect to the dataset's qualifiers. In this analysis, we highlight both the complexity of the task and the feasibility and utility of applying text-to-SQL for information retrieval in the process mining domain.

The paper is structured as follows: Section 2 presents the fundamental theoretical concepts and related benchmark datasets; Section 3 describes the $text_2SQL_4PM$ benchmark dataset, including the methods used in its construction, the qualifiers used for annotation, and statistics and examples of the instances included; Section 4 discusses the study of the text-to-SQL task supported by the dataset, establishing a baseline solution associated with it; finally, Section 5 presents the conclusions, followed by the references.

2 Background and Related Work

In this section, we present a brief definition of SQL and the text-to-SQL task, and some concepts of the process mining field, which is the domain of the dataset created. We also present related works that provide publicly available datasets similar to ours, whether for text-to-SQL tasks or process mining tasks.

2.1 SQL and Text-to-SQL

Structured Query Language (SQL) is a declarative language used to organize, manipulate, and retrieve information from relational databases. This language is commonly employed to query data through a single request, known as an SQL statement, which

[1]In this paper, a qualifier is defined as a label that represents a specific perspective of analysis in evaluating the competence of the text-to-SQL solution.

consists of commands[2] that allow customization of the retrieved data. An SQL statement operates on the relations defined in a schema of a relational database and returns, as a result, a relation (temporary, existing at runtime) in relational format.

Definition 1 (**Schema, Relation, Relational Database** [9]). A relational database $\mathcal{D}$ is a collection of data organized according to the principles established in the relational data model, i.e., data organized as a collection of relations defined according to a schema. A relation schema $R(A_1, A_2, ..., A_n)$, consists of a relation name R and a list of attributes, $A_1, A_2, ..., A_n$. Each attribute A_i has a domain $dom(A_i, R)$ in the relation schema R. A relation r of the relation schema $R(A_1, A_2, ..., A_n)$ is a set of n-tuples $r = \{t_1, t_2, ..., t_m\}$. Each n-tuple t is a list of n values $t = \langle v_1, v_2, ..., v_n \rangle$, ordered as defined by the list of attributes $A_1, A_2, ..., A_n$ of the relation schema and according with the corresponding $dom(A_i, R)$.

Since SQL is a language for computational processing, it follows a well-known syntax based on a standard norm.

Definition 2 (**The syntax of a standard SQL statement**).

SELECT A_1, A_2, ..., A_n
FROM source relations r
WHERE condition(s)
GROUP BY A_1, A_2, ..., A_n
HAVING condition(s)
ORDER BY A_1, A_2, ..., A_n

Each clause of an SQL statement has a well-established function: *SELECT* is responsible for defining the attributes that should structure the resulting relation; *FROM* indicates the relations that serve as data sources for information retrieval; *WHERE* is a clause that applies filters to the data sources; *GROUP BY* is responsible for grouping data, usually to support statistical summaries; *HAVING* is a clause that filters the information from aggregations; and *ORDER BY* is a clause used to impose sorting on the data in the resulting relation. In addition to this basic structure, the following are also cited as basic commands: arithmetic operations, set operations, string processing operations, and nullity test conditions.

The text-to-SQL task aims to generate a SQL statement from a natural language utterance that can be executed on a database to retrieve information.

Definition 3 (**Text-to-SQL task**). Let $c : U \rightarrow S$ be the implementation of text-to-SQL task, in which U is a universe of natural language utterances, S a universe of SQL statements, and $c(u) \rightarrow s$ is a conversion procedure from elements $u \in U$ to elements $s \in S$. In this conversion procedure, the following assertions are valid, due to the expressiveness of the natural language and the SQL language:

- $U' \subset U \mid \forall\, (u'_i, u'_j) \in U'$ and $i \neq j$, u'_i and u'_j are paraphrases in natural language, meaning they express the same intent;

[2]The most common SQL commands are: SELECT with or without arithmetic operations (AVG, COUNT, MIN, MAX, SUM), INNER JOIN, WHERE, ALL, GROUP BY, HAVING, ORDER BY, BETWEEN, LIKE, IS NULL, IS NOT NULL, UNION, INTERSECT, EXCEPT, IN, NOT IN, ANY, SOME.

- $S' \subset S \mid \forall(s'_i, s'_j) \in S'$ and $i \neq j$, s'_i and s'_j are equivalent SQL statements, meaning that they produce the same result when executed;
- if $c(u'_i) \rightarrow s'_i$, then $c(u'_i) \rightarrow s'_j$, $c(u'_j) \rightarrow s'_i$ and $c(u'_j) \rightarrow s'_j$ for any i, j.

Solutions for implementing $c : U \rightarrow S$ aim to efficiently address two well-known problems: *schema representation* for R, which serves as input for the conversion procedure c along with the utterance u; and *schema linking*, which involves connecting the terms and concepts in the utterance u with the relations, attributes, and values in D. Building on the discussions outlined by Wang et al. [10], we assert that the former problem entails an appropriate encoding of schema elements (such as relation names, attribute names, their respective domains, and primary and foreign keys), while the latter involves aligning entity references extracted from u with the encoded schema elements.

Initially, the deep learning models fine-tuned for implementing c treated schema representation and schema linking as separate problems, striving to devise sophisticated methods for representing database schemas, e.g. through graphs [11]. However, with the advent of the cross-domain text-to-SQL task, these two challenges began to be addressed *in tandem* [10][12]. The emergence of large language models for the text-to-SQL task has rendered this separation nearly imperceptible. When the utterance u and database schema R are provided via a prompt, implementations using these models aim to efficiently extract and present only the schema information from R that is most relevant to the u specified in the prompt [13][14].

2.2 Process Mining Domain

Process mining brings together data and process sciences with the main goal of automatically extracting knowledge about business processes from *event logs*. An event log L is a sequential file that records *events* related to the execution of *activities* (i.e., well-defined steps in a process) within the business process under analysis. Typically, we assume that each *event* e is related to a particular process instance, referred to as *case* $c \in C$. Additional information, called *attributes*, may be recorded in event logs, such as the timestamp of the event, the person or resource executing the activity, or any other data elements recorded with the event [6].

Definition 4 (**Event, Attribute [6]**)**.** Let $\mathcal{E}$ be a universe of events, i.e. the set of all possible event identifiers. Events may have various attributes, such as timestamp, activity, resource, cost, and others. Let $\mathcal{AN}$ be a set of attribute names. For any event $e \in \mathcal{E}$ and name $n \in \mathcal{AN} : \#_n(e)$ is the value of attribute n for event e. Typically, for each existing attribute related to the events in $\mathcal{E}$, a domain is defined for its values, i.e., if we consider the *timestamp* attribute $\#_{timestamp}(e)$, we define the time domain T and $\#_{timestamp}(e) \in T, \forall e \in \mathcal{E}$.

Each event in the event log is globally unique, i.e., the same event cannot occur twice in a event log. An event log consists of *cases* and cases consist of events. The events for a case are organized in a *trace*, i.e., a sequence of unique events. Moreover, cases, like events, can have attributes.

Definition 5 (**Case, Trace, Event log [6]**)**.** Let $\mathcal{C}$ be the case universe, i.e. the set of all possible case identifiers. Cases, like events, have attributes. For any case $c \in \mathcal{C}$ and name $n \in \mathcal{AN} : \#_n(c)$ is the value of attribute n for case c. Each case has a special mandatory attribute *trace*: $\#_{trace}(c) \in \mathcal{E}^*$. $\hat{c} = \#_{trace}(c)$ is a shorthand for referring to the trace of a case. A *trace* $\sigma \in \mathcal{E}^*$ is a finite non-empty sequence of unique events ascending ordered by occurrence time, i.e., for $1 \leqslant i < j \leqslant |\sigma| : \sigma(i) \neq \sigma(j)$ and $\sigma(i)$ occurs before $\sigma(j)$. An event log L is a set of cases $L \subseteq \mathcal{C}^*$ where each event appears only once in the log, i.e. for any two different cases the intersection of their events is empty.

A definition that simplifies the concept of a trace is presented by Aalst [6] as a 'simple trace', and for practical purposes in process mining, 'simple trace' is referred to as 'variant' in the computational tools of the field. Both are used to represent an analytical perspective based on the control flow of a process instance, i.e., based on the notion of the execution of a sequence of activities.

Definition 6 (**Simple trace, Variant**)**.** A *simple trace* σ' is a sequence of activities related to the events that make up the trace σ [6]. The term *variant* is commonly used to refer to σ' in computational tools.

Table 1 shows an excerpt of an event log. Only three cases are shown, and their respective traces contain three, five, and four events, respectively. Each event has a unique identifier and several attributes. For example, the first event in the event log is an instance of the activity 'Create ticket' that occurred on February 20th at 10:30 was executed by Joana and cost 10 dollars. The second case starts with the third event in the event log and also refers to an instance of the activity 'Create ticket'.

Table 1 Example of a log excerpt

		Attributes				
Event identifier	Case identifier	Activity	Timestamp	Resource	Cost	...
1	1	Create ticket	20/02/2021 10:30	Joana	10	...
2	1	Activate ticket	20/02/2021 10:33	Paul	50	...
3	2	Create ticket	20/02/2021 10:33	Ana	10	...
4	2	Activate ticket	20/02/2021 10:40	Paul	50	
5	1	Await for user input	20/02/2021 11:10	Cris	100	...
6	2	Await for user input	20/02/2021 15:50	Cris	100	...
7	3	Create ticket	23/02/2021 16:01	Joana	10	...
8	3	Activate ticket	23/02/2021 16:09	Paul	50	...
9	3	Handle ticket	23/02/2021 16:12	Paul	50	...
10	3	Close ticket	23/02/2021 16:55	Paul	10	...
11	2	Handle ticket	25/02/2021 11:42	Cris	50	...
12	2	Activate ticket	25/02/2021 12:40	Cris	70	...
...	...	...	...	...	...	...

Several process mining tasks and techniques rely on the sequence of the events within the cases (e.g. process model discovery, conformance checking, process monitoring, and descriptive, predictive, and prescriptive analysis). Therefore, the events within a case must be ordered on the event log by the moment of occurrence [6]. In Table 1, the 'Timestamp' column provides the information of the moment when the activity was executed and the event log is ordered by it. It is also common to organize the event log grouping the events of each case, but the order of events within the case is always maintained.

The field of process mining has certain particularities that should be noted here, as they can influence the declarative extraction of information procedures:

- Partial ordering in event logs: In some information systems where event logs are recorded, no timestamp information is available. In other cases, timestamps can be too coarse, such as commonly seen in hospitals where information systems only record a date. In addition, when the event log results from merging data from different sources, timestamp-related problems can arise due to multiple clocks and delayed recording. However, in principle, such event log ordering does not require timestamps. One way to address ordering problems is to assume only a partial ordering of events (i.e., not a total order) and subsequently use dedicated process mining algorithms for this. Another way is to define the order based on domain knowledge or frequent patterns across days [6].
- Process query language: While the tabular nature of an event log allows the use of SQL, it is not the only declarative way to query information in an event log. Specialized languages for querying process data, called process query language (PQL) have been developed to offer process mining-specific operators. For example, PQL includes the SOURCE and TARGET operators to link events registered in different tuples in a relation, the VARIANT operator to aggregate cases event names into a single string, and the CONFORMANCE operator for conformance checking. Despite these specialized languages, choosing SQL for use promotes generalization and accessibility, due to its popularity. SQL's widespread use and familiarity far outweigh the benefits of a more specialized language like PQL, whose expertise is harder to find.

2.3 Related work

Several domain-specific datasets with pairs of English natural language utterances and the corresponding SQL statement exist and have been used by research community for decades, like ATIS [15], GeoQuery [16], Scholar [17], Yelp and IMDB [18].

With advancements in deep learning models and language models that generate a SQL statement given a natural language utterance, more robust datasets were published. Such datasets are considered cross-domain. In this context, 'cross-domain' refers to a dataset that consists of multiple databases, each with schemas and data related to distinct domains. For the text-to-SQL task, the training and testing sets should not share the same domain; in other words, models are expected to generalize across different domains. Examples of datasets in such category include:

- WikiSQL [4]: a large dataset consisting of 80,654 pairs of utterances and corresponding SQL statements on 24,241 databases. It is organized with only one relation and the SQL statements structure is very simple with just SELECT and WHERE clauses.
- Spider [3]: a robust dataset consisting of 10,181 utterances and 5,693 corresponding SQL statements on 200 databases covering 138 different domains. This database contains multiple relations and more complex SQL statements, including nested clauses and operators such as JOIN, GROUP BY, ORDER BY, and HAVING. It also incorporates more 1,659 utterances in the training dataset that come from other datasets.
- BIRD [5]: a robust dataset containing 12,751 utterances and SQL statements pairs across 95 databases with seven tables on average on 37 different domains. The SQL statements contains function operators such as DATE, YEAR, IIF, STRFTIME and CAST and the use of CASE conditions on SELECT. The utterances are challenging requiring knowledge domain to be answered. This dataset focuses on database values, with the databases containing dozens of rows, unlike other datasets that contain only a few rows.

When it comes to datasets in Portuguese for the text-to-SQL task, mRAT-SQL+GAP [19] is the only general-purpose dataset found. It is the result of a translation using Google Cloud Translation API[3] of 8,659 training utterances and 1,034 dev utterances (used for test purposes) from the Spider dataset. The SQL statements were kept in English. In the realm of process mining, Barbieri et al. (2022) presented a dataset with statements in the form of questions, in a style similar to the datasets used in question-answering tasks. Each question addresses a common information need in process mining, which can be resolved either through simple SQL statements or through complex specialized algorithms [20]. The dataset was used in the task of translating a natural language question to a logical query that could be run on existing process mining tools and it does not have gold standard outputs to be used in supervising learning methods. The original set of 250 questions was written in Portuguese and volunteers translated them to English, resulting in a total of 794 questions. Only the English questions are publicly available[4].

The dataset $text_2SQL_4PM$, introduced herein, differs from previous ones by featuring a unique set of characteristics: it is domain-specific (process mining), bilingual, designed for supervised learning methods, fully generated and reviewed by humans, and comes with a set of qualifiers and a baseline text-to-SQL solution based on a large language model. Although it is still small compared to datasets used for benchmarking large language models, it includes statements and SQL queries that cover a complete range of basic SQL commands, varying from simple statements to highly complex ones.

3 $text_2SQL_4PM$ Dataset

The dataset introduced in this paper, entitled $text_2SQL_4PM$, was designed to facilitate the development of natural language to SQL conversion engines, particularly those

[3]Cloud Translation API: https://googleapis.dev/python/translation/latest/index.html.
[4]https://ic.unicamp.br/~luciana.barbieri/pmquestions.csv

based on machine learning, for the field of process mining. Additionally, it serves as a benchmark for assessing any conversion engine designed to tackle the Text-to-SQL task within the process mining domain.

The contents of this dataset[5] include utterances for information extraction from a business process context, written in Portuguese and English, SQL statements that solve the requested information retrieval, and descriptive information that allows for generating some statistics about the dataset. The dataset is located in the domain of a business process concerning the dynamics of authorization requests for reimbursement of academic travel expenses incurred by university staff. The event log associated with this business process is well-known in the process mining community and was the subject of the BPI Challenge competition in 2020[6].

This section presents the method followed to construct the $text_2SQL_4PM$ dataset and a series of descriptive statistics that describe its complexity.

3.1 Method

The three-phased method followed in the generation of the dataset is depicted in Figure 1. In Phase 1, initial dataset content was generated by undergraduate and graduate students as part of their coursework in Data Mining and Process Mining classes. A scoring system was implemented to incentivize students to produce high-quality content for this exercise. Subsequently, Phase 2 and Phase 3 were conducted by three researchers with extensive expertise in process mining and SQL. These phases involved an evaluation of the initial content, a domain adaptation of the complete dataset, and a data augmentation process. The remaining of this section provides a detailed description of each phase.

Phase 1 - Dataset content generation:

The goal of the first phase was the generation of the initial content, maximizing the variability of utterances and SQL statement types that would compose the $text_2SQL_4PM$ dataset. To achieve this, the following strategies were applied:

- **Participants**: 29 undergraduate students and 13 graduate students, enrolled in courses whose syllabus focused on process mining topics and who had previous knowledge of SQL, were invited to participate in an exercise to generate pairs (NL-PT utterance/statement SQL)[7] that could appropriately express the extraction of useful information about a business process from an event log.
- **Score system and guidelines**: A scoring scheme was assigned to the exercise, considering criteria related to the quantity of pairs created, their correctness and the variety of commands used in the SQL statements (see Section 2.1). Two guidelines were provided to the students to enhance the quality of their content: i) the utterances must be useful for a process manager to request information from the

[5]The dataset, as well as excerpts from the event log, containing content in Portuguese and English, enabling the exploration of the dataset's SQL statements, are available at https://github.com/pm-usp/text-2-sql. The excerpts from the event log have been slightly modified from the original event log to facilitate a more accurate analysis of the correctness of the SQL statements.

[6]https://icpmconference.org/2020/bpi-challenge/

[7]NL-PT: natural language in Portuguese.

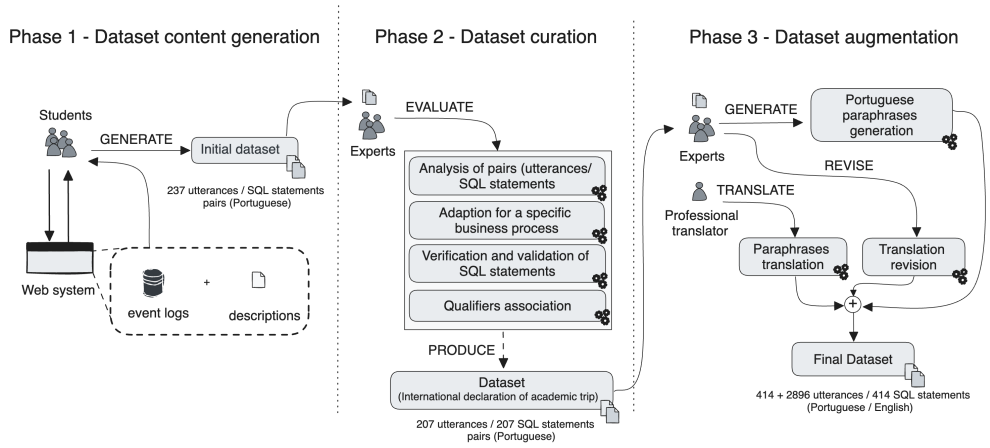

Fig. 1 Overview of $text_2SQL_4PM$ dataset generation process

event log, and ii) only pure SQL statements should be elaborated, meaning that cursors and control flow could not be used.

- **Business process**: Six different business processes, each represented by a descriptive text and at least one excerpt from an associated event log, were randomly presented to the students.[8] Thus, different real-world situations were available for the creative process of the student group, increasing the chances of generating statements with different information extraction objectives.
- **Data collection**: An *ad hoc* WEB system guided the students in producing the pairs (NL-PT utterance/statement SQL). Through this system, students had access to three (of six) descriptions of business processes and one of their respective excerpts from event logs. From this information, they were required to register at least one utterance-statement pair for each business process. Additionally, non-normalized database tables[9], each containing an excerpt from an event log, were made available within a relational database management system for students to test the SQL statements produced. A total of 237 utterance-statement pairs were produced through this exercise for the initial dataset content.

Phase 2 - Dataset curation:

The goal of the second phase was the improvement of the dataset content. The initial content of the dataset underwent detailed data curation involving three process mining experts (a senior researcher, a doctoral student, and a master's student with experience in the software industry hereafter referred to as *expert A*, *expert B*, and *expert C*, respectively) in the following actions:

[8] The business processes and their respective event logs were related to: incident management [21], open and close problems management [22], judicial performance of Brazilian justice [23], financial loan requests [24], authorization requests for civil construction [25], and international declaration of academic trip [26].

[9] The event log schema consists of a single non-normalized table, lacking foreign keys, and containing duplicated information in the 'id-case' column.

- Analysis of each NL-PT utterance aimed at evaluating if they were correctly formulated and meaningful in the context of information retrieval in process mining, conducted by *expert A* and *expert C*. Correctness was evaluated regarding the appropriate use of process mining jargon concerning its basic concepts (see Section 2.2). Meaningfulness was evaluated concerning the usefulness of the information to be retrieved from that utterance. If an error or inadequacy was observed, either the utterance and its respective SQL statement were adapted, or the pair under analysis was discarded.
- The adaptation of the entire dataset to focus on a specific business process, carried out by *expert A* and *expert C*. Although several business processes were used in the content generation of the dataset (phase 1), we determined that only one of them would be the focus of attention in the dataset to enable its use in exploring text-to-SQL issues in the process mining domain, isolating problems that could arise from the variability of vocabulary resulting from the use of different business processes. The utterances and SQL statements were adapted according to the vocabulary used in the business processes related to authorization requests for reimbursement of academic travel expenses [26] (see examples in Table 2).
- Verification and validation of the SQL statements by *expert A* and *expert C*, aiming to ensure correctness in the use of the SQL language and accuracy in information retrieval, according to what was specified in the utterance.

Table 2 Examples of adaptation applied to the dataset in order to align it with the domain of international academic travel declarations. The changes are shown in bold. Note that this concerns the alteration of vocabulary inherent to the business process.

	Original	Adapted
Example 1	*Utterance*: How many activities **'closed'** do we have? *SQL*: SELECT count(*) FROM events_log WHERE activity = **'Closed'**	*Utterance*: How many activities **'end trip'** do we have? *SQL*: SELECT count(*) FROM events_log WHERE activity = **'End trip'**
Event log:	incident management	international declaration of academic trip
Example 2	*Utterance*: How many events are associated with the **'Petition joined'** activity? *SQL*: SELECT COUNT (*) FROM events_log WHERE activity = **'Petition joined'**	*Utterance*: How many events are associated with the **'declaration rejected by director'** activity? *SQL*: SELECT COUNT (*) FROM events_log WHERE activity = **'Declaration rejected by director'**
Event log:	judicial performance of Brazilian justice	international declaration of academic trip

The dataset curation also involved associating the utterance-statement pairs with a series of qualifiers, carried out by the *expert* B and *expert C*, in order to organize the

dataset's content regarding relevant aspects from the perspective of process mining (PMp), natural language (NLp), and Structured Query Language (SQLp):

- **Qualifier 1** (PMp): refers to whether the utterance can be answered considering each event independently or if the events need to be aggregated by case. Values: *event level - case level.*
- **Qualifier 2** (PMp): it refers to the perspective from which the process is analyzed given the information request in the utterance, and whether it is a statistical information extraction or conformance verification. Values: *perspective (control flow, temporal, resource, cost) - descriptive statistics (control flow, temporal, resource, cost) - conformance.*
- **Qualifier 3** (PMp): it refers to the process mining concepts that occur as the objective of externalizing information, i.e., concepts that are used in the SELECT clause. If an aggregation occurs, then it is also indicated along with the aggregate concept. Values: *case - event - timestamp - activity - resource - cost.*
- **Qualifier 4** (PMp): it refers to the process mining concepts that occur as the objective of filtering information, i.e., concepts that are used in the WHERE clause. Values: *case - timestamp - activity - resource - cost - none.*
- **Qualifier 5** (NLp): it refers to the classic wh-question classification of the utterances. Values: *how - what - which - when - who - none.*
- **Qualifier 6** (SQLp): it determines whether the SQL statement involves an aggregation function in the SELECT clause. Values: *aggregation - none.*
- **Qualifier 7** (SQLp): it determines whether a condition on the GROUP BY clause is required to answer the utterance, i.e., it indicates the presence of a HAVING clause. Values: *having - none.*
- **Qualifier 8** (SQLp): it refers to the hardness criteria provided by Spider[3]. The hardness criteria consider four levels of difficulty based on number of SQL components (selections and conditions) present in SQL statements. Values: *easy - medium - hard - extra hard - no hardness.*

After curatorial actions were performed, the dataset consisted of 205 revised and qualified utterance-statement pairs in Portuguese.

Phase 3 - Dataset augmentation:

The Phase 3 of dataset generation was dedicated to data augmentation. Two actions were carried out in this phase: building paraphrases and translating the utterances/statements into English. The paraphrases were created manually by the *expert A* and *expert B*. For paraphrase creation, the original utterances were either completely rewritten or had some elements replaced by linguistic or technical synonyms. Table 3 shows three cases of paraphrase creation, illustrating the three strategies used.

Once the paraphrases were created, two additional qualifiers were established, one regarding the process mining perspective and the other regarding the natural language perspective:

- **Qualifier 9** (PMp): The meaning of each value for this qualifier is as follows. Values: *value- generic - domain.*

Table 3 Strategies used for paraphrase creation

Base utterance	Paraphrase	Strategy
Which cases had their first log record before March 2017?	Processing records began occurring before March 2017 for which declarations?	completely rewriten
Which cases **arrived at** the 'end trip' activity between 2016 and 2017?	Which cases **went through** the 'end trip' activity between 2016 and 2017?	replace by linguistic synonyms
Which **cases** went through the 'declaration rejected by supervisor' activity? Sort ascending by start date.	Which **process instances** went through the 'declaration rejected by supervisor' activity? Sort the answer in ascending order by start date.	replace by technical synonyms

  - *value*: the vocabulary of the utterance has an explicit connection with the schema of database tables/columns and column values, and in most cases, values are enclosed in single quotes in the utterance, allowing them to be easily replaced with other values.
  - *generic*: the vocabulary of the utterance has an explicit connection with the schema of database tables/columns and with values with specific columns (case ID, event ID and timestamp), therefore it is process domain independent.
  - *domain*: the vocabulary of the utterance primarily considers the natural vocabulary used in the process domain and has little or no explicit connection with the schema of database tables/columns and column values.

- **Qualifier 10** (NLp): it refers to derived utterances known as paraphrases from an initial utterance known as the base. Values: *base - paraphrase*.

Since all individuals involved in generating the dataset are Portuguese native speakers, an English native professional translator was hired to translate all utterances into English. The translator was informed about the dataset's objective and was guided on the use of process mining field jargon and the importance of maintaining the integrity of the paraphrases during the translation. The translation was reviewed by *expert C* to ensure that the process mining jargon and the original meaning of the utterances were maintained.

To complete the process of creating the dataset in two languages, we established two event logs: one with the original values in English and another with the values translated into Portuguese. Consequently, the SQL statements also have two versions to align with their respective event logs. The decision to create both versions was important so that the evaluation of text-to-SQL conversion capability could be performed equally for both languages.

As a result, the dataset comprises 1,655 quadruples (NL-PT utterance, NL-EN utterance, SQL statement - PT, SQL statement - EN), consisting of 205 originals and 1,450 paraphrases associated with ten qualifiers.

3.2 Statistics and examples

In this section we present statistics of the dataset according to the series of qualifiers defined in Section 3.1. For each dataset analysis perspective (PMp, NLp, and SQLp), we present examples of utterances or statements that illustrate the classifications established under each qualifier, overviews of how many quadruples are associated with each class, and relationships between the classes of different qualifiers, whenever relevant.

Process mining perspective (PMp):

Table 4 shows examples for the qualifiers 1, 2, 3, 4 and 9. Observing the examples for each qualifier mentioned in the table clarifies the interpretation used when they were applied to the dataset:

- for Qualifier 1, the examples clarify what we consider as an information request at the event level or at the case level. In the former example, all filters (in this case, filters related to the activity name, resource name, and timestamp) are applied within each row of the table, meaning a specific event is being analyzed. In the latter example, it is necessary to analyze two subsequent rows, including the row where the filter on the activity name occurs, the analysis goes beyond the event to involve a case context; to make this possible in a standard SQL statement, the table (event log) is recursively wrapped in an INNER JOIN command.
- for Qualifier 2, the first example concerns a request in which a user is interested in a feature about the workflow of process instances, more specifically when it involves passing through an activity, meaning some perspectives of the process dynamics is being explored; furthermore, a restriction regarding the person (resourve) involved in the work is specified. The second example illustrates a request in which a simple count is requested. The third example can be seen as an audit activity (a conformance verification task), as it requests cases with the highest numbers of activities executed and their duration.
- for Qualifier 3, the utterance in the first example requests information about 'events' that occurred within a specific time period; to address this, the SQL statement must include this information in the SELECT clause. In the second example, the focus is on 'cases', thus, information about the 'case' concept is associated with the SELECT clause.
- for Qualifier 4, in the first example, the utterance refers to event information related to a specific activity, so a filter related to the 'activity' concept should be applied in the WHERE clause. In the second example, the process mining concept of interest pertains to the 'resource' concept.
- for Qualifier 9, in the first example, the text-to-SQL converter should identify which attribute, in the table schema, the explicitly mentioned values can be found. The second example is an utterance that applies to any event log, regardless the underlying business process domain of the event log. Finally, in the third example, specific domain words are used and they must be indirectly related to the table schema, so the text-to-SQL converter should understand that 'interventions' refers to 'events' and 'declaration' refers to 'case'.

Table 4 Examples for qualifier categories from a process mining perspective

	Example	Qualifier value
Qualifier 1	What were the activities carried out by the resource named Thomas in the first semester of 2018?	event level
	What are the activities that preceded a 'send reminder' occurrence?	case level
Qualifier 2	Show the identifiers of the cases that went through the 'declaration rejected by pre-approver' activity, except those in which the activity was performed by the Douglas 'resource'.	perspective (control-flow and resource)
	How many resources worked on each case?	descriptive statistics (resource)
	What were the five cases with the most performances of activities and what was the duration of each of them?	conformance
Qualifier 3	What events occurred in the year 2017?	event
	In which declarations were up to 20 activity occurrences performed	case
Qualifier 4	Which resources were responsible for the 'declaration rejected by pre-approver' activity?	activity
	List the cases in which the 'Wayne' resource was allocated, ordering the response by case identifier.	resource
Qualifier 9	How many times were 'start trip' and 'end trip' activities performed?	value
	Which cases were entirely handled by the same resource?	generic
	How many interventions were carried out in the processing of each declaration?	domain

Table 5 shows the number of quadruples classified in each class of qualifiers 1, 2, 3, 4 and 9. The charts organized in Figure 2 show the information about the distribution of the quadruples over the qualifiers. For qualifier 2, there is more than one value associated with the same quadruple, which causes the total number of associated values to be greater than the total number of quadruples in the dataset.

Some values for the qualifiers are infrequently present in the dataset. This is mainly due to the creative bias of the original utterances created by the students who participated in Phase 1 of the dataset generation process. Some highlights include the low number of requests that: are formulated with reference to the most likely discourse for business managers (value *domain* for Qualifier 9); allude to compliance verification tasks; deal with cost analysis in general; deal with temporal analysis as information resulting from the SQL request; and deal with case-based filtering.

Several factors may have contributed to the low occurrence. Among them, we noted: the students training involved in the utterance formulation being entirely focused on the computing area and not on business; the difficulty of considering auditing as information to be extracted from SQL requests; and the fact that some event

logs used in the initial generation of utterances did not contain information about event cost. Specifically regarding the 'domain' value in Qualifier 9, there is a higher number of utterances among the paraphrase subset. This is because *expert A* created utterances with a vocabulary different from that commonly used by SQL programmers or process mining analysts. These utterances are composed of language that more closely resembles what would be used by business managers, considering domain-specific nomenclature rather than the standard terminology present in event logs and associated relational tables.

Table 5 Process mining perspective: number of quadruples per qualifier/class. Legend: B - base; P - paraphrase; BP - base + paraphrase

		B	P	BP			B	P	BP
Qualifier 1	event level	110	802	912	Qualifier 3	case	87	593	680
	case level	95	648	743	(SELECT)	event	75	586	661
						resource	62	413	475
Qualifier 2	perspective	165	1130	1295		activity	39	281	320
	descriptive statistics	129	894	1023		timestamp	18	120	138
	conformance	24	180	204		cost	15	114	129
Qualifier 9	value	115	623	738	Qualifier 4	activity	86	608	694
	generic	84	484	568	(WHERE)	none	78	561	639
	domain	6	343	349		timestamp	25	184	209
						resource	23	133	156
						cost	11	84	95
						case	8	44	52

Natural Language (NLp):

Table 6 shows some examples for the qualifiers 5 and 10. To apply the qualifications related to natural language processing, the following interpretations were assumed:

- for Qualifier 5, the classification of utterances as wh-questions can result in three situations that warrant attention: an utterance may have an imperative form, as in the first example in Table 6, in which case the value 'none' is chosen; an utterance may involve more than one condition for requesting information, leading to the assignment of multiple classifications (second example in Table 6); semantically equivalent utterances (including paraphrases) may have different linguistic constructions and therefore receive different classifications, as shown in the three last examples in Table 6.
- for Qualifier 10, the examples demonstrate the construction of paraphrases based on conceptual equivalence ('to perform an activity' means 'to be involved in in an event' and is related to 'the employee's workload'). Other strategies include replacing date formats with written-out dates, changing interrogative sentences to imperative sentences, or using topicalization.

Table 7 shows the number of quadruples classified in each class of the Qualifier 5. The referenced distribution of paraphrases is shown in Figure 3. The chart in Figure 4 shows the distribution of base and paraphrase utterances present in the dataset. In this

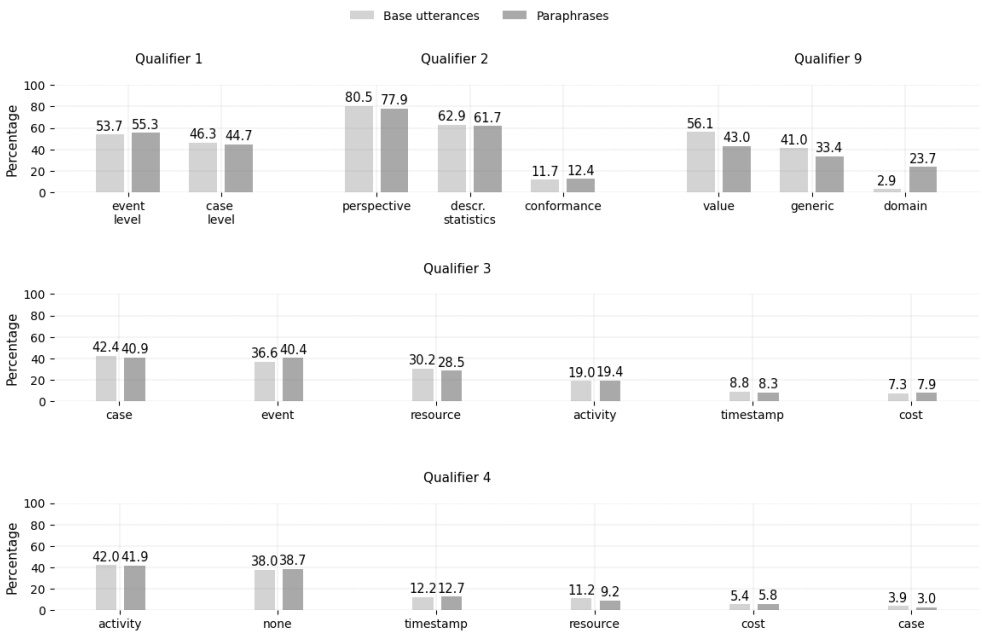

Fig. 2 Process mining qualifiers comparison

distribution, we observe the number of paraphrases created for each base utterance, noting that most base utterances have between six to nine paraphrases.

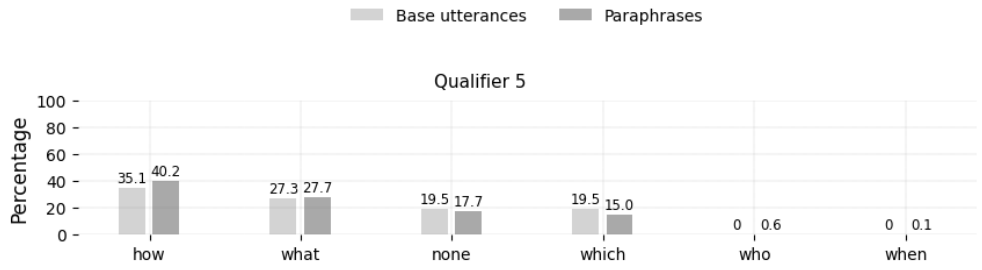

Fig. 3 WH-question classification comparison

Structured Query Language perspective (SQLp):

Table 8 shows examples for the Qualifier 6, 7 and 8. The qualifier 6 indicates whether an aggregation operation is present or not in the SELECT clause, and Qualifier 7 specifies whether a filter is applied after the GROUP BY operation. The class assignment for the qualifier 8 was carried out through automatic analysis of SQL

Table 6 Examples for qualifier categories from the natural language perspective

	Utterance	Qualifier value
Qualifier 5	Show all the cases that ended in March 2018.	none
	Which resources are related to more events and how many events are they related to?	which;how
	Return the five resources requested in the greatest number of cases.	none
	What are the top 5 resources with the most cases?	what
	Report the five employees who made the most declarations.	who
Qualifier 10	How many times did the 'Thomas' resource perform an activity in 2017?	base
	How many events was the 'Thomas' resource involved in in 2017?	paraphrase
	What was the workload of the employee Thomas in terms of performing actions for processing declarations in 2017?	paraphrase

Table 7 Natural language perspective: number of quadruples per values for Qualifier 5. The sum of the counts exceeds the number of utterances in the dataset because some receive more than one classification.

		Base	Paraphrase	Base + Paraphrase
Qualifier 5	how	72	401	473
	what	56	257	313
	none	40	583	623
	which	40	217	257
	who	0	8	8
	when	0	2	2

statements, according to the criteria adopted by the Spider dataset [27]. Essentially, these criteria are based on the number of SQL components, selections, and conditions contained in the SQL statement. For example[10],

- the easy class contains only one projection in the SELECT clause and no more than one condition in either the WHERE clause or GROUP BY HAVING clause, but not both;
- the medium class contains two projections in the SELECT clause and one condition on the WHERE clause with a GROUP BY clause;
- the hard class contains nested subqueries;
- the extra class contains GROUP BY HAVING clause with nested subqueries.

The 'no hardness' class concerns the impossibility of classifying some gold SQL statements. This impossibility is due to limitations arising from the scope of SQL commands used in the Spider dataset. Such limitation refers to the nonexistence of the following SQL commands or operators:

[10]For the complete set of rules, see the Spider [3].

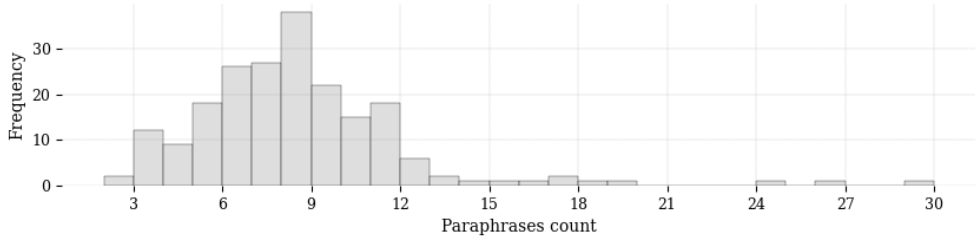

Fig. 4 Paraphrases count distribution. X-axis: Number of paraphrase utterances per base utterance; Y-axis: Number of base utterances containing a specific number of paraphrase utterances. Example: there are 8 paraphrase utterances for 38 base utterances.

- 'NULL' and 'NOT NULL';
- function *strftime*;
- WITH clause;
- table alias on FROM clause used on SELECT (ex.: SELECT *p*.attribute FROM table p ...);
- alias on FROM clause resulting from a SUBSELECT;
- clauses after a FROM clause with SUBSELECT;
- parentheses on WHERE clause; alias on SELECT clauses (ex.: SELECT attribute *as c*);
- window functions (ex.: lead).

The number of quadruples classified in each class of qualifiers 6, 7 and 8 is shown in Table 9. The respective distributions of these quadruples over the qualifier are illustrated by the charts organized in Figure 5.

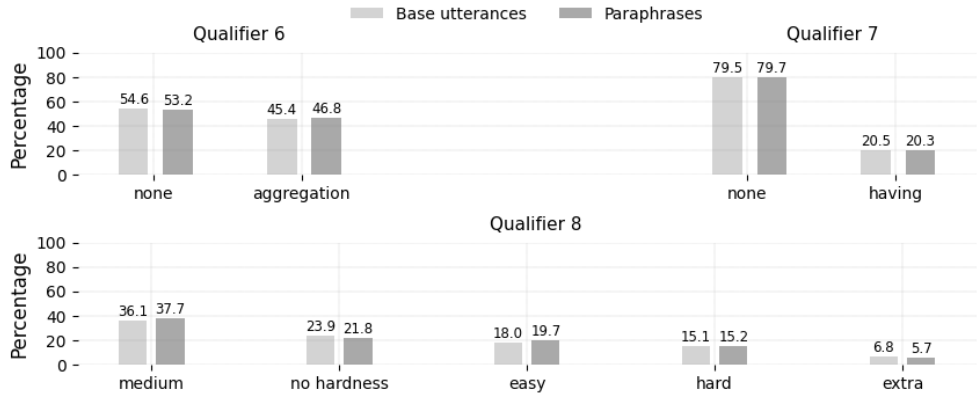

Fig. 5 SQL qualifiers comparison

Table 8 Examples qualifier categories from a structured query language perspective

	Statement SQL	Qualifier value
Qualifier 6	SELECT **DISTINCT idcase** FROM event_log WHERE cost > 2225	none
	SELECT **count(*)** FROM event_log WHERE resource = 'Peter'	aggregate
Qualifier 7	SELECT COUNT(*), resource FROM event_log **GROUP BY resource**	none
	SELECT idcase, resource from event_log group by idcase **HAVING COUNT(DISTINCT resource) = 1**	having
Qualifier 8	SELECT count(*) FROM events_log WHERE activity = 'End trip'	easy
	SELECT resource, count(DISTINCT idcase) FROM events_log WHERE timestamp BETWEEN '2017-01-01' AND '2017-05-31' GROUP BY resource	medium
	SELECT COUNT (DISTINCT resource) FROM events_log WHERE idcase IN (SELECT idcase FROM events_log GROUP BY idcase HAVING COUNT(*) > 3)	hard
	SELECT idcase from events_log GROUP BY idcase HAVING COUNT(*) IN (SELECT COUNT(*) FROM events_log GROUP BY idcase ORDER BY COUNT(*) desc LIMIT 10) ORDER BY COUNT(*) DESC	extra
	WITH RankedEvents AS (SELECT id, activity, timestamp, LEAD(timestamp) OVER (PARTITION BY idcase ORDER BY timestamp) AS next_timestamp, idcase FROM events_log), DurationEvents AS (SELECT activity, (strftime('%s', next_timestamp) - strftime('%s', timestamp)) as duration FROM RankedEvents WHERE next_timestamp IS NOT NULL) SELECT activity, AVG(duration) AS average_duration FROM DurationEvents GROUP BY activity	no hardness

Table 9 Structured Query Language perspective: number of quadruples per qualifier/class. Legend: B - base; P - paraphrase; BP - base + paraphrase

		B	P	BP			B	P	BP
Qualifier 6	none	112	772	884	Qualifier 8	medium	74	546	620
	aggregation	93	678	771		no hardness	49	316	365
						easy	37	285	322
Qualifier 7	none	163	1156	1319		hard	31	220	251
	having	42	294	336		extra	14	83	97

4 $text_2SQL_4PM$ Baseline

This section presents the results of using the $text_2SQL_4PM$ Dataset as a basic test case for implementing the text-to-SQL task. Based on these results, a baseline is established to assess the feasibility and potential value of applying this task in the process mining domain. The baseline is derived from the use of a large language model from the GPT family. The following sections provide a detailed overview of the model application method and the results obtained.

4.1 Method

The following resources and procedures were applied in the test for building the $text_2SQL_4PM$ baseline[11]:

Large Language Model: the GPT-3.5 Turbo model was used to provide SQL statements for each utterance in the dataset. The company OpenAI is one of the leaders in providing large language models and is the provider of the chosen model. The model used was selected for this purpose mainly due to its recognized performance in the text-to-SQL task [28][13] and its affordable cost.

Database Management System (DBMS): the DBMS SQLite we used for verification and validation of the SQL statements provided by the GPT-3.5 Turbo model. For this purpose, two database instances were generated for storing the event log 'international declaration of academic trip' [26], each containing a single relation (table) with attribute names in English and values in either Portuguese or English. In both cases the instance values are case insensitive.

Prompt Engineering: The first prompt used in OpenAI's official Text-to-SQL demo, known as the OpenAI Demonstration Prompt, was used. We tested prompts in both Portuguese and English. For Portuguese utterances, the entire prompt was translated to Portuguese, except the database schema, which was kept in English. Example of prompts in English are depicted in the Listing 1, and examples in Portuguese, in the Listing 2. The prompts were designed using a zero-shot[12] approach, with: (i) the task specification, (ii) the database schema to be used in the task, and (iii) a text completion prompt that provides the first word of the model's response.

Listing 1 Prompt example (English)

```
### Complete sqlite SQL query only and with no explanation
### SQLite SQL tables, with their properties:
#
# event_log(id, activity, timestamp, resource, cost, idcase)
#
### What events associated with the 'end trip' activity did not
    take place on December 12, 2017?
SELECT
```

Listing 2 Prompt example (Portuguese)

```
### Complete somente a consulta sqlite SQL e sem explicação
### Tabelas SQLite SQL, com suas propriedades:
#
# event_log(id, activity, timestamp, resource, cost, idcase)
#
```

[11] All scripts and files used in the experiment are available in https://github.com/pm-usp/text-2-sql.

[12] According to Liu et at. [29], zero-shot prompt is a strategy where a pre-trained language model is applied to a task without any additional training specific to that task. The model uses predefined cloze or prefix prompts to generate the desired output, and this approach is called 'zero-shot' because no task-specific training data is used.

```
### Quais eventos associados à atividade 'fim da viagem' não
    aconteceram no dia 12 de dezembro de 2017?
SELECT
```

Evaluation: To evaluate the performance of the GPT-3.5 Turbo model on the text-to-SQL task using the $text_2SQL_4PM$ dataset, two indicators were used: 'exact set match without values' and 'execution accuracy' [27], referred to here as the structure indicator and run indicator, respectively. These indicators are widely used in related literature.

- **Structure indicator**: evaluates only the structure of the SQL statement, ignoring values in clauses condition. According to this indicator, a response is considered valid if the SQL statement generated by the model exactly matches a gold-standard SQL statement that was previously associated with the input utterance. This gold-standard statement serves as the reference for the correct information retrieval requested in the utterance. On the one hand, the success rate for this metric can be underestimated since a simple alias added on SELECT clause of the SQL statement generated by the GPT-3.5 Turbo model is considered a failure. Table 10 shows examples on this matter. On the other hand, this metric does not validate the DISTINCT keyword on SELECT clause, which can result in an overestimation of success. In addition, due to limitations of the implementation used related to 'no hardness' class, out of the 1,655 utterance-SQL statement pairs, 365 are not included in the calculation of the total result percentages for this metric, resulting in 1,290 pairs to be analyzed.
- **Run indicator**: evaluates the results of SQL statements when executed in a DBMS. An SQL statement generated by the model is considered valid if the results obtained by its execution match the results obtained by executing the gold SQL statement previously associated with the input utterance. For this indicator, a case-insensitive strategy was adopted for the values of attributes involved in the SQL statements conditions.

Table 10 Examples of failure on structure indicator just by simple addition of alias on SELECT clause of the SQL statement generated

	Generated	Gold
English	SELECT resource, **COUNT(*) as num_activities** FROM event_log WHERE activity = 'declaration approved by administration' GROUP BY resource	SELECT **count(*)**, resource FROM event_log WHERE activity = 'Declaration approved by administration' GROUP BY resource
Portuguese	SELECT **COUNT(DISTINCT resource) AS total_recursos** FROM event_log	SELECT **COUNT(DISTINCT resource)** FROM event_log

4.2 Results

The results forming the proposed $text_2SQL_4PM$ baseline are presented from four perspectives: performance under the structure indicator; performance under the run indicator; comparative performance analysis between the indicators; and general challenges.

Structure indicator: Among the 1,290 utterance-SQL pairs of utterance-statements SQL analyzed for each language, the success rate for the structure indicator was 31.8% for Portuguese and 32.7% for English, corresponding to 410 and 422 correct answers, respectively. Some characteristic factor of the process mining domain influence the low success rates achieved by this indicator. Table 11 presents an example of such factors:

- the implicit reference to the event concept through its definition (activity execution), applied in the gold SQL statement, in contrast to the explicit reference to the attribute 'activity' in the generated SQL statement. Although both SQL statements are equivalent, the indicator points to an error.

Table 11 Example of utterance complexity to which the structural indicator is sensitive. The differences between the gold and generated SQL statements are highlighted in blue.

EN	**Utterance**: Which process instances have more than 20 activities performed?
PT	**Utterance**: *Em quais instâncias de processo há mais de vinte ocorrências de atividades executadas?* **Gold SQL**: SELECT idcase FROM event_log GROUP BY idcase HAVING count(*) > 20 **Generated SQL - for both natural languages**: SELECT idcase FROM event_log GROUP BY idcase HAVING COUNT(DISTINCT activity) > 20

However, the values the indicator yield different magnitudes depending on the group of utterance-SQL pairs being analyzed. Table 12 presents detailed results for three PMp qualifiers. This closer, qualifier-level analysis enables us to identify where challenges arise within each category, offering insights not apparent from the overall success rates:

- the lower success rate observed in the 'case level' class of Qualifier 1 compared to 'event level' suggests that utterances involving cases are more challenging. This is likely because case-level process mining concepts are often implicit in utterances, providing fewer contextual cues for the models;
- the qualifier-level breakdown highlights challenges within the 'domain' class of Qualifier 9 for both languages. Here, low performance is attributed to difficulties in linking utterance terms to the database schema, especially when domain-specific vocabulary is used; for example, terms like 'declarations' in the utterance often lack a clear association with items in the schema.
- although English generally performed slightly better than Portuguese on average, closer inspection of each qualifier class reveals situations where English faced greater challenges. For instance, in the 'descriptive analysis' class of Qualifier 2, English

shows a significant 26.7 drop in the structure indicator for paraphrased statements compared to Portuguese, underscoring the complexity of handling paraphrased utterances in this context.

Table 12 Results for the structure indicator considering the qualifiers from the process mining perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	**Qualifier values**	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
1	event level	43.2 (41/95)	34.9 (244/699)	47.4 (45/95)	36.1 (252/699)
	case level	32.8 (20/61)	24.1 (105/435)	27.9 (17/61)	24.8 (108/435)
2	perspective	50.8 (61/120)	42.0 (345/821)	49.2 (59/120)	41.7 (342/821)
	descriptive statistics	31.0 (31/100)	23.7 (171/721)	33.0 (33/100)	24.1 (174/721)
	conformance	33.3 (5/15)	16.5 (18/109)	40.0 (6/15)	15.6 (17/109)
9	value	50.0 (43/86)	45.8 (226/493)	50.0 (43/86)	47.1 (232/493)
	generic	25.8 (17/66)	22.6 (86/381)	28.8 (19/66)	25.7 (98/381)
	domain	25.0 (1/4)	14.2 (37/260)	0.0 (0/4)	11.5 (30/260)

Tables 13 provides detailed results for the Qualifier 5 of NLP perspective. The structure indicator for Qualifier 5 highlights that the 'who' class presents a challenge in both languages. All utterances in this class also fall under the 'domain' category of Qualifier 9, which, as previously discussed, suffers from domain-specific vocabulary challenges. The same utterances complexities aforementioned negatively impacted 'what' and 'which' classes. Since most types of 'what' and 'which' statements are retrieving the complete tuple in projections, some generated SQL statements are considered unsuccessful because they explicitly mention each attribute in the PROJECT clause, while the gold SQL uses the alias '*'.

Tables 14 provides detailed results for the Qualifier 8 from SQL perspective. This qualifier reveals a negative correlation between the complexity and structure indicator success. The 'hard' and 'extra-hard' classes have the lowest structure indicator values, as expected, due to their increased complexity.

Run indicator: Across all utterance-SQL pairs, the success rate for the run indicator was 44.5% for Portuguese and 47.6% for English, corresponding to 737 and 788 correct answers, respectively. The improvement in the run indicator relative to the structure indicator suggests that GPT-3.5 Turbo model generates SQL statements that may fail structural checks but still produce accurate results, as shown in the examples of Table 15.

Table 13 Results for the structure indicator considering a qualifier from the natural language process perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	**Qualifier values**	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
5	how	39.6 (21/53)	28.5 (86/302)	37.7 (20/53)	25.8 (78/302)
	what	37.2 (16/43)	27.9 (55/197)	39.5 (17/43)	32.5 (64/197)
	none	51.6 (16/31)	35.1 (163/465)	48.4 (15/31)	36.3 (169/465)
	which	25.0 (8/32	25.0 (44/176)	31.2 (10/32)	27.3 (48/176)
	who	0.0 (0/0)	0.0 (0/7)	0.0 (0/0)	0.0 (0/7)
	when	0.0 (0)	100.0 (2/2)	0.0 (0/0)	100.0 (2/2)

Table 14 Results for the structure indicator considering a qualifier from the structured query language perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	**Qualifier values**	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
8	easy	64.9 (24/37)	50.2 (143/285)	73.0 (27/37)	53.3 (152/285)
	medium	43.2 (32/74)	30.6 (167/546)	39.2 (29/74)	30.2 (165/546)
	hard	9.7 (3/31)	12.7 (28/220)	12.9 (4/31)	14.5 (32/220)
	extra	14.3 (2/14)	13.3 (11/83)	14.3 (2/14)	13.3 (11/83)
	no hardness	-	-	-	-

Table 15 Examples of success on run indicator but fail on structure indicator

	Generated	Gold
EN	SELECT AVG(cost) FROM event_log WHERE activity = 'payment handled' AND strftime ('%Y', timestamp) < '2018'	SELECT AVG(cost) FROM event_log WHERE timestamp < '2018-01-01' AND activity = 'Payment handled'
PT	SELECT activity, COUNT(*) FROM event_log WHERE activity IN ('início da viagem', 'fim da viagem') GROUP BY activity;	SELECT count(*), activity from event_log where activity = 'Início da viagem' OR activity = 'Fim da viagem' GROUP BY activity

The results for the run indicator detailed per qualifier are depicted in tables 16, 17, and 18. With regard to the analysis of the PMp qualifiers, it is observed that there was a significant improvement in the value of the run indicator, in relation to the structure indicator, for utterances at the 'event level', for 'descriptive statistics', and for general-purpose ones ('generic'). This may indicate both an ease in the correct interpretation of the utterances by the GPT-3.5 Turbo model and the possibility of

task resolution with a broader variety of equivalent statements (although syntactically different). It is also observed that higher values in the run indicator are obtained for utterances that address the value 'how' for Qualifier 5, and for all levels of SQL statement complexity (Qualifier 8). Furthermore, in this indicator, it is possible to evaluate the SQL classifiers as 'no hardness', for which the GPT-3.5 Turbo model shows the greatest difficulty in resolving the text-to-SQL problem.

Table 16 Results for the run indicator considering the qualifier organization of utterances from the process mining perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	**Qualifier values**	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
1	event level	64.5 (71/110)	53.2 (427/802)	68.2 (75/110)	56.2 (451/802)
	case level	38.9 (37/95)	31.2 (202/648)	40.0 (38/95)	34.6 (224/648)
2	perspective	47.3 (78/165)	34.3 (388/1130)	50.3 (83/165)	38.2 (432/1130)
	descriptive statistics	61.2 (79/129)	52.1 (466/894)	62.0 (80/129)	53.5 (478/894)
	conformance	37.5 (9/24)	23.3 (42/180)	37.5 (9/24)	21.7 (39/180)
9	value	50.4 (58/115)	51.8 (323/623)	53.9 (62/115)	57.0 (355/623)
	generic	58.3 (49/84)	56.2 (272/484)	60.7 (51/84)	59.5 (288/484)
	domain	16.7 (1/6)	9.9 (34/343)	0.0 (0/6)	9.3 (32/343)

Table 17 Results for the run indicator considering the qualifier organization of utterances from the natural language process perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	**Qualifier values**	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
5	how	68.1 (49/72)	53.9 (216/401)	68.1 (49/72)	54.6 (219/401)
	what	42.9 (24/56)	34.2 (88/257)	48.2 (27/56)	40.9 (105/257)
	none	57.5 (23/40)	47.3 (276/583)	57.5 (23/40)	50.6 (295/583)
	which	30.0 (12/40)	25.3 (55/217)	37.5 (15/40)	29.5 (64/217)
	who	0.0 (0/0)	0.0 (0/8)	0.0 (0/0)	0.0 (0/8)
	when	0.0 (0)/0	100.0 (2/2)	0.0 (0/0)	100.0 (2/2)

Table 18 Results for the run indicator considering the qualifier organization of utterances from the structured query language perspective. The results are presented as a percentage of correct answers along with the corresponding absolute numbers of each corresponding qualifier class, for Portuguese and English utterances.

Qualifier	Qualifier values	Portuguese		English	
		Base	**Paraphrase**	**Base**	**Paraphrase**
8	easy	73.0 (27/37)	54.7 (156/285)	86.5 (32/37)	64.2 (183/285)
	medium	73.0 (54/74)	63.2 (345/546)	74.3 (55/74)	65.8 (359/546)
	hard	35.5 (11/31)	30.0 (66/220)	32.3 (10/31)	33.6 (74/220)
	extra	35.7 (5/14)	30.1 (25/83)	35.7 (5/14)	27.7 (23/83)
	no hardness	22.4 (11/49)	11.7 (37/316)	22.4 (11/49)	11.4 (36/316)

Comparative analysis: An analysis based on the GPT-3.5 Turbo model's performance from the perspective of Qualifier 8 (SQL perspective) confirms that although more complex SQL statements are harder for the GPT-3.5 Turbo model to generate, there are also more ways to design favoring the evaluation under the run indicator (there is a significant difference between the values obtained in the structure indicator and the run indicator for queries at the hard and extra complexity levels).

Grouping each base utterance with their paraphrases and segregating by hardness classes, the distribution on figure 6 shown that the GPT-3.5 Turbo model fails according to both indicator for only 27 and 23 utterance-SQL pairs (sections shaded in light gray on the bars in the figure), for Portuguese and English respectively. Specifically for the unclassified pairs ('no hardness'), evaluated only under the run indicator, failures are observed more frequently (37 and 36 for Portuguese and English, respectively).

General challenges: The process mining domain has specific characteristics that pose challenges for text-to-SQL solutions. In the presented $text_2SQL_4PM$ baseline, three characteristics stand out, and they are illustrated in tables 19, 20, and 21:

- in the example in Table 19, the understanding regarding the number of instances that can be associated with the object to be retrieved (suggested by the plural term 'cases' – more than one case can have the 'largest' size in terms of the number of events) is correctly expressed in the gold SQL statement but not addressed in the generated SQL statement (which necessarily returns a single instance as the result). Although this problem can be found in other application domains, it becomes especially relevant in process mining when retrieving information at the 'case level', the main object of interest in process mining.
- in the example in Table 20, the understanding of the temporal order of events ('permit final approved by supervisor' must occur before 'start trip'), correctly expressed in the gold SQL statement, but incorrectly represented in both generated SQL statements. In this case, for English, no order was required in the result, and for Portuguese, the reverse order was required. Information related to the sequence of

Fig. 6 Distribution of success rates for translating natural language utterances into SQL statements, grouped by base utterance and corresponding paraphrases. Each bar in the chart represents a group. The bar size indicates the number of utterances in each group (base utterance + corresponding paraphrases). Bars entirely in light gray indicate that the translation failed for all utterances in the group. Dark gray in a bar indicates the number of utterances for which the translation succeeded in both indicators. Blue and red colors indicate that the translation succeeded in only one indicator, structure indicator or run indicator, respectively. The qualifier 8, related to the complexity of the expected SQL statement, is used to organize the groups under analysis.

events is especially important in process mining due to the interest in understanding the execution dynamics of a business process.

- in the examples of Table 21, the GPT-3.5 model provides SQL statements that partially address the expected answer (first example on table); or does not deal with the DISTINCT command (second example on table). The latter problem stems from the fact that the data in the event log is not normalized ('case' attribute) or not unique ('activity' attribute). In both cases, one could say that there is a problem of dubiety in natural language, as the result obtained from executing the generated SQL statements can be considered correct under some level of abstraction in the interpretation of the utterance.

5 Conclusion

In this paper, we present the benchmark dataset $text_2SQL_4PM$. $text_2SQL_4PM$ has been established as a benchmark because it offers both the dataset and a baseline solution for the text-to-SQL task. This dataset is specifically tailored to the process mining domain – a prominent area of data exploration where MANY stakeholders lack

Table 19 Example of utterance with difficult for the text-to-SQL solution: number of instances to be recovered. The main differences between the gold and generated SQL statements are highlighted in blue.

EN	**Utterance**: Retrieve the identifiers of the cases that have the greatest number of events.
PT	**Utterance**: *Recupere os identificadores dos casos que possuem a maior quantidade de eventos.* **Gold SQL**: SELECT idcase FROM event_log GROUP BY idcase HAVING COUNT(*) = (SELECT COUNT(*) FROM event_log GROUP BY idcase ORDER BY count(*) DESC LIMIT 1) **Generated SQL - for both languages**: SELECT idcase FROM event_log GROUP BY idcase ORDER BY COUNT(id) DESC LIMIT 1

Table 20 Example of utterance with difficult for the text-to-SQL solution: temporal ordering. The main differences between the gold and generated SQL statements are highlighted in blue.

EN	**Utterance**: In how many cases does the 'start trip' activity eventually follow the 'permit final approved by supervisor' activity?
PT	**Utterance**: *Em quantos casos a atividade 'início da viagem' segue a atividade 'permissão final aprovada pelo supervisor' eventualmente?* **Gold SQL**: SELECT COUNT(DISTINCT e1.idcase) FROM event_log e1 INNER JOIN event_log e2 ON e1.idcase = e2.idcase WHERE e1.activity = 'Permit final approved by supervisor' AND e2.activity = 'Start trip' AND e1.timestamp < e2.timestamp **Generated SQL - for English**: SELECT COUNT(DISTINCT idcase) FROM event_log WHERE activity = 'start trip' AND idcase IN (SELECT idcase FROM event_log WHERE activity = 'permit final approved by supervisor') **Generated SQL - for Portuguese**: SELECT COUNT(DISTINCT idcase) FROM event_log e1 JOIN event_log e2 ON e1.idcase = e2.idcase WHERE e1.activity = 'início da viagem' AND e2.activity = 'permissão final aprovada pelo supervisor' AND e1.timestamp < e2.timestamp

Table 21 Example of utterance with difficult for the text-to-SQL solution: incompleteness or dubiety. The main differences between the gold and generated SQL statements are highlighted in blue.

EN	**Utterance**: What are the events that are not associated with resources?
PT	**Utterance**: *Quais são os eventos que não estão associados a recursos?* **Gold SQL**: SELECT * FROM event_log WHERE resource IS NULL **Generated SQL - for both languages**: SELECT activity, timestamp, cost, idcase FROM event_log WHERE resource IS NULL
EN	**Utterance**: Which activities contain the word 'declaration' in their label?
PT	**Utterance**: *Quais atividades contém a palavra 'declaração' em seu rótulo?* **Gold SQL**: SELECT DISTINCT activity FROM event_log WHERE activity LIKE '%declaration%' **Generated SQL - for both languagems**: SELECT activity FROM event_log WHERE activity LIKE '%declaration%';

technical expertise in SQL but seek to retrieve information related to business process management.

The primarily use case for $PM_{text2sql}$ dataset was for fine-tuning and evaluation of text-to-SQL implementations in process mining context for request information; therefore, it can be used for other natural language tasks because their richness features such as: i) bilingual, ii) a curated and assessment process of the generated utterances and corresponding SQL statements, iii) a careful enrichment with paraphrases and iv) qualifiers from diverse perspectives; and (v) careful curation by humans. The bilingual nature of the dataset, featuring natural language utterances and enriched paraphrases generated by native Portuguese speakers, along with their corresponding English versions translated by a professional, makes it a valuable resource for semantic parsing tasks and potentially for other natural language processing tasks such as machine translation or paraphrase generation. Specifically, within the process mining domain, the carefully crafted paraphrases created by researchers with extensive expertise in this area represent an especially valuable resource.

The particular features of the $text_2SQL_4PM$ dataset for the text-to-SQL task are:

- $text_2SQL_4PM$ contains Portuguese natural language utterances and the corresponding values in SQL statements also in Portuguese, so that can be used to fine-tune and evaluate text-to-SQL implementations in Portuguese language. $text_2SQL_4PM$ joins the mRAT-SQL+GAP dataset created using automatic translations of the Spider dataset from English to Portuguese [19], serving as another resource for automatic processing of Portuguese.
- although the simple one-table schema structure of a event log has been used, the request information that needs timestamped sequence of events to be answered, so much common in process mining context, can be a quite challenge to construct a SQL statement. Thus, some of the natural language utterances of the dataset imposes a real challenge to text-to-SQL implementations and serves as a precise benchmark in such a domain.
- the utterances manual qualification can be used for classification tasks or a separation of concerns when assessing a model implementation with the $text_2SQL_4PM$ dataset.

The limitations of the $text_2SQL_4PM$ benchmark dataset are:

- it can only be used to train text-to-SQL models for exploratory information retrieval tasks. Other types of information requests, which are important in process mining but require advanced event log processing - such as discovery, optimization, and process monitoring — are not covered. This limitation can only be partially overcome with future research efforts due to inherent constraints of the SQL language itself. An alternative to extend the utility of semantic parsing in process mining, though not fully overcoming the limitation at hand, could be to focus on creating datasets using a process query language [30].
- the SQLite DBMS was used for verification and validation of the statements SQL. Therefore, some pairs of utterance-SQL statement in the dataset use specific SQLite functions, mainly date time and window functions.

References

[1] Yu, T., Li, Z., Zhang, Z., Zhang, R. & Radev, D. Walker, M., Ji, H. & Stent, A. (eds) *TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation.* (eds Walker, M., Ji, H. & Stent, A.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 588–594 (Association for Computational Linguistics, New Orleans, Louisiana, 2018). URL https://aclanthology.org/N18-2093.

[2] Katsogiannis-Meimarakis, G. & Koutrika, G. A survey on deep learning approaches for text-to-sql. *The VLDB Journal* **32**, 905–936 (2023).

[3] Yu, T. *et al.* Riloff, E., Chiang, D., Hockenmaier, J. & Tsujii, J. (eds) *Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task.* (eds Riloff, E., Chiang, D., Hockenmaier, J. & Tsujii, J.) *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3911–3921 (Association for Computational Linguistics, Brussels, Belgium, 2018). URL https://aclanthology.org/D18-1425.

[4] Zhong, V., Xiong, C. & Socher, R. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR* **abs/1709.00103** (2017).

[5] Li, J. *et al.* Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs (2023). 2305.03111.

[6] van der Aalst, W. M. P. *Process Mining: Data Science in Action* 2 edn (Springer, Heidelberg, 2016).

[7] van der Aalst, W. M. P. & Carmona, J. (eds) *Process Mining Handbook* Vol. 448 of *Lecture Notes in Business Information Processing* (Springer, 2022). URL https://doi.org/10.1007/978-3-031-08848-3.

[8] IEEE. Ieee standard for extensible event stream (xes) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016* 1–50 (2016).

[9] Elmasri, R. & Navathe, S. *Fundamentals of Database Systems* (Pearson Education, 2010).

[10] Wang, B., Shin, R., Liu, X., Polozov, O. & Richardson, M. Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J. (eds) *RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers.* (eds Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7567–7578 (Association for Computational Linguistics, Online, 2020).

[11] Bogin, B., Gardner, M. & Berant, J. Korhonen, A., Traum, D. & Màrquez, L. (eds) *Representing Schema Structure with Graph Neural Networks for Text-to-SQL Parsing*. (eds Korhonen, A., Traum, D. & Màrquez, L.) *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4560–4565 (Association for Computational Linguistics, Florence, Italy, 2019). URL https://aclanthology.org/P19-1448.

[12] Li, H., Zhang, J., Li, C. & Chen, H. RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL (2023).

[13] Gao, D. *et al.* Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *CoRR* **abs/2308.15363** (2023).

[14] Pourreza, M. & Rafiei, D. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction (2023). URL https://arxiv.org/abs/2304.11015. 2304.11015.

[15] Dahl, D. A. *et al.* Allan, J. (ed.) *Expanding the Scope of the ATIS Task: The ATIS-3 Corpus*. (ed.Allan, J.) *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994* (1994). URL https://aclanthology.org/H94-1010.

[16] Zelle, J. M. & Mooney, R. J. Clancey, W. J. & Weld, D. (eds) *Learning to parse database queries using inductive logic programming*. (eds Clancey, W. J. & Weld, D.) *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, 1050–1055 (AAAI Press, 1996).

[17] Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J. & Zettlemoyer, L. Barzilay, R. & Kan, M.-Y. (eds) *Learning a Neural Semantic Parser from User Feedback*. (eds Barzilay, R. & Kan, M.-Y.) *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 963–973 (Association for Computational Linguistics, Vancouver, Canada, 2017). URL https://aclanthology.org/P17-1089.

[18] Yaghmazadeh, N., Wang, Y., Dillig, I. & Dillig, T. SQLizer: query synthesis from natural language. *Proc. ACM Program. Lang.* **1** (2017). URL https://doi.org/10.1145/3133887.

[19] José, M. A. & Cozman, F. G. Brito, A. & Delgado, K. V. (eds) *mRAT-SQL+GAP: A portuguese text-to-sql transformer*. (eds Brito, A. & Delgado, K. V.) *Anais da X Brazilian Conference on Intelligent Systems* (SBC, Porto Alegre, RS, Brasil, 2021).

[20] Barbieri, L., Madeira, E. R. M., Stroeh, K. & van der Aalst, W. M. P. Munoz-Gama, J. & Lu, X. (eds) *Towards a Natural Language Conversational Interface for Process Mining*. (eds Munoz-Gama, J. & Lu, X.) *Process Mining Workshops*, 268–280 (Springer International Publishing, Cham, 2022).

[21] Amaral, C. A. L., Fantinato, M., Reijers, H. A. & Peres, S. M. Ziemba, E. (ed.) *Enhancing completion time prediction through attribute selection.* (ed.Ziemba, E.) *Information Technology for Management: Emerging Research and Applications*, 3–23 (Springer International Publishing, Cham, 2019).

[22] Steeman, W. Bpi challenge 2013, incidents (2013). URL https://data.4tu.nl/articles/_/12693914/1.

[23] Unger, A. J. *et al.* Maranhão, J. (ed.) *Process mining-enabled jurimetrics: analysis of a brazilian court's judicial performance in the business law processing.* (ed.Maranhão, J.) *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, 240–244 (Association for Computing Machinery, New York, NY, USA, 2021).

[24] van Dongen, B. Bpi challenge 2017 (2017). URL https://data.4tu.nl/articles/_/12696884/1.

[25] van Dongen, B. B. Bpi challenge 2015 (2015). URL https://data.4tu.nl/collections/_/5065424/1.

[26] van Dongen, B. BPI Challenge 2020 (2020). URL https://data.4tu.nl/collections/_/5065541/1.

[27] Zhong, R., Yu, T. & Klein, D. Webber, B., Cohn, T., He, Y. & Liu, Y. (eds) *Semantic Evaluation for Text-to-SQL with Distilled Test Suites.* (eds Webber, B., Cohn, T., He, Y. & Liu, Y.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 396–411 (Association for Computational Linguistics, Online, 2020). URL https://aclanthology.org/2020.emnlp-main.29.

[28] Dong, X. *et al.* C3: Zero-shot Text-to-SQL with ChatGPT (2023). 2307.07306.

[29] Liu, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* **55** (2023).

[30] Vogelgesang, T. *et al. Celonis PQL: A Query Language for Process Mining*, 377–408 (Springer International Publishing, Cham, 2022). URL https://doi.org/10.1007/978-3-030-92875-9_13.