MATE, a Meta Layer Between Natural Language and Database

Irene Sucameli[1], Alessandro Bondielli[2,1], Lucia C. Passaro[2], Edoardo Annunziata[3], Giulia Lucherini[3], Andrea Romei[3] and Alessandro Lenci[1]

[1]*Department of Philology, Literature and Linguistics, University of Pisa, Pisa, Italy*
[2]*Department of Computer Science, University of Pisa, Pisa, Italy*
[3]*BNova srl, Massa, Italy*

Abstract

Nowadays, the knowledge of query languages like SQL is mandatory for accessing the most relevant information concerning a company's business, stored in richly structured databases. The advancement of Natural Language Processing and Deep Learning research has made it possible to develop different models for the conversion of natural language questions into formalized queries. Although the performance of these models is very satisfactory on internationally established benchmarks for the English language, it is undoubtedly necessary to investigate their portability with respect to the types of databases and natural languages used for formulating the questions. For this reason, we realised MATE (Meta lAyer between naTural language and databasE), a framework for interfacing humans and databases, thus facilitating the access in natural language to the data stored in databases. Indeed, MATE is developed with the aim of accessing information in a very simple way, from the perspective of a human-centered AI.

Keywords

Natural Language Understanding, Text to Query, Data Warehouses

1. Introduction

It is well known that most relevant factual information about a company is stored in some form of data management systems such as databases and data warehouses (DWH). This typically serves the purpose of enabling and supporting business intelligence and analytic tasks within the company. However, direct access to this information is often precluded to many users who would directly benefit from it, due to the "know-how" required to transform their desired requests into formal query languages, such as SQL. This limits the potential of analysis at various levels within the company, especially for less tech-savvy users, and may compel the use of pre-determined analyses that yield sub-optimal results for understanding the trends and nuances of the available data.

NL4AI 2022: Sixth Workshop on Natural Language for Artificial Intelligence, November 30, 2022, Udine, Italy [1]
irene.sucameli@fileli.unipi.it (I. Sucameli); alessandro.bondielli@unipi.it (A. Bondielli); lucia.passaro@unipi.it (L. C. Passaro); edoardo.annunziata@bnova.it (E. Annunziata); giulia.lucherini@bnova.it (G. Lucherini); andrea.romei@bnova.it (A. Romei); alessandro.lenci@unipi.it (A. Lenci)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3287/paper15.pdf

Recent years have seen an important growth in the research area concerning models for the translation of natural language questions into formalized queries. These models typically leverage neural architectures in which pre-trained Language Models are used to interpret queries expressed in natural language. Such systems are known in the literature as *seq2sql* models [2], since they exploit architectures similar to those used in *seq2seq* (encoder-decoder) systems for machine translation [3, 4], considering natural language as the translation source and a query language as the target. This approach is possible also thanks to the availability of natural language – SQL parallel datasets [2, 5].

Such models have shown impressive performances on benchmark datasets. However, a key aspect must be taken into account: these models are typically learned on specific language pairs (e.g., English to SQL), thus limiting their out-of-the-box portability to other query languages. Information about the schema of the specific target database must be directly encoded by the model as well. Moreover, while performances on gold-standard data are definitely crucial in pushing the literature forward, we can argue that these models are fundamentally limited in real-world scenarios. Further, in real-world business intelligence applications DWH are often used instead of simpler operational databases. Although DWHs are formally similar, they are organised in such a way as to ensure the analysis of business facts by facilitating filtering and aggregation.

For these reasons, in this work we propose **MATE** (**M**eta l**A**yer between na**T**ural language and databas**E**). It is a meta layer able to act as a bridge between users' questions expressed in natural language and a more structured query language of choice. We propose to leverage Natural Language Understanding (NLU) techniques for sequence labelling, in order to obtain a textual representation that is more easily translatable to a structured query. In particular, we chose a set of labels that can be directly associated with actions and operations (e.g., selection, filtering, and sorting) performed to obtain knowledge from transactional data. Compared to current state-of-the-art solutions, the main benefit of our meta layer is that MATE is independent of the formal language used, since the learning part takes place directly within the Language Understanding module of the model.

MATE is being developed in partnership with BNova,[1] an Italian data intelligence company, within the Text2Query project (POR-FESR 2014-2020). MATE is implemented inside a conversational agent within BNova's proprietary analytic platform, NIKY, in order to act as a natural language-based bridge between NIKY users (e.g., company executives) and the data concerning their company they are most interested in. However, we can argue that MATE can be easily adapted and integrated within different systems and languages, both natural and query ones.

This paper is organized as follows. In Section 2 we highlight related work in the field, including academic and commercial proposals. Section 3 details the proposed approach. Further, Section 4 provides a case study that serves as a preliminary evaluation of the approach, described in 5. Finally, Section 6 draws some conclusions and describes future works.

[1] www.bnova.it

2. Related work

In recent years, research on Natural Language Interface to Databases (NLID) has seen many advances. Several approaches to the problem of converting text into queries have been proposed, varying from seq2seq and seq2sql models [6, 7, 2] up to more recent approaches, which use pre-trained Neural Language Models to link natural language questions to a database schema by exploiting attention mechanisms [8, 9, 10]. A number of recent papers have in fact proposed to leverage Transformer-based models for this task, either by incorporating BERT-style contextual information into a problem dependent structure [9], or by jointly learning textual, contextual, and schema-level representations in the same model in order to align the various modalities, e,g., the textual questions and the relational schema [8, 10]. These models have shown good performances on established benchmarks for the English language. Among such benchmarks, we can consider WikiSQL [2], a dataset built via crowdsourcing which includes over 80k annotated questions, SQL tables and queries extracted from Wikipedia, and Spider [5], a complex cross-domain dataset which consists of more than 10k questions and 5k SQL queries over 138 different domains. However, it is still necessary to further investigate the portability of systems which refers to different types of databases or natural languages used for formulating queries.

As for commercial systems, augmented analysis and natural language querying are becoming key topics in helping analysts to explore their data [11, 12]. Such systems typically permit to query data by using business and domain-related terms that can be expressed in a text box, or spoken. Among these, Tableau [13] is worth mentioning. It uses a text box with drop down lists on the metadata domain to help users to compose their queries. In contrast, our model works on a free-form text and allows users to easily express their request using natural language. The NL4DV tool [14] has instead a strong emphasis on visualization: it takes as input a tabular dataset and a natural language request about that dataset, and automatically selects the more appropriate visualization.

Finally, as for the development of intermediate representations, which are of particular interest for the present work, the literature is less extensive. Most systems propose to convert user input expressed in natural language into a query expressed in an intermediate representation language. Subsequently, the intermediate query is transformed into a database language query [15, 16]. The advantage of this approach is the adaptability of the metalanguage and its portability with respect to different databases [16]. For example, authors in [15] proposed an intermediate representation (NatSQL) which simplifies the SQL structure while retaining its keywords and syntax (e.g., SELECT, WHERE and ORDER BY clauses). This facilitates NatSQL in generating queries from text.

Our proposed approach, Mate, is developed under similar assumptions, since it constitutes an intermediate layer between the questions expressed in natural language and the queries to the database. However, NatSQL is designed to serve as an alternative target for neural text-to-SQL models. Because our proposed approach decouples model learning from the target formal language, we instead structure our intermediate representation in a more general way, choosing elementary operations whose composition reflects typical online analytical processing (OLAP) workloads. The use of our meta layer would then allow to better control the data flow transmitted during the translation of textual input to query, simplifying the communication between NLU models and DWHs.

```
"intent": "request related to DB Table",
"focus": [ "focus of the query"],
"filters": {
    "measures": ["values to aggregate"],
    "filtering slots": ["name of the slot"] },
"operations": {
    "sort_order": string,
    "count": boolean,
    "group_by": string,
    "avg": string }
```

Figure 1: MATE 's structure.

3. MATE

MATE is an intermediate meta layer which stands between a natural language and a query language. It allows developers to integrate more easily database data and natural language-based agents. MATE is based on a frame-slot semantic, which is typical of NLU systems.

A *frame* is a data structure which contains a collection of slots whose value can be used to encode different kinds of semantic representations [17]. The NLU system would then parse user's utterance in order to extract both *intents*, which describe the goal(s) the user is trying to accomplish, and *slots*, which convey the information required to fulfil the intent. For example, from the request "Show me which Japanese restaurants are open in Florence", the intent and slots extracted will be the following ones: `INTENT: SHOW-RESTAURANT`; `SLOT-TYPE: Japanese`; `SLOT-CITY: Florence`.

The idea behind MATE is to create a layer that organises the intents and slots extracted in a structure and a format suitable for being interpreted as database queries, by means of rules and heuristics. MATE is therefore an intermediate meta layer between the textual input and its transformation into a query. To this end, the meta layer must include all the pieces of information, extracted from the conversational flow, which are required to "build" a query to a database. In order to do so, we propose a slot-specific semantics which allows us to connect slots to a query language. As illustrated in Figure 1, the detection of the main fact of the DWH, which corresponds to the **FROM** statement, in MATE coincides with the identification of the **intent** of the request, while the detection of the object of interest, which corresponds to the **SELECT** statement, is represented in MATE as the **focus**. The focus is followed by a list of **filters**, which represent the dimensions constraining the search (**WHERE**). Filters include nested data which can be i) **measures**, corresponding to the values that have to be aggregated, or ii) a series of optional **filtering slots**, corresponding to the columns of the database tables on which the search is performed.

Finally, MATE's structure includes a list of possible **operations** to be carried out within the database according to the focus and the extracted filters. We have decided to use a limited list of possible SQL operations that can be performed, although MATE can be fast adapted to different needs and implemented with further operations. Currently, the operations supported

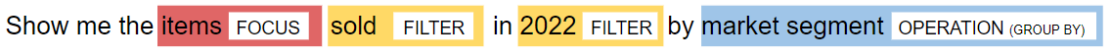

Figure 2: Example of a tagged sentence in RASA NLU module.

are **ORDER BY** (stored as **SORT ORDER** in the meta layer), **COUNT**, **GROUP BY** and **AVG**. When translating our intermediate structure to SQL, as a first step each word is mapped to its corresponding value, column and table in the database via a similarity score. The tables required to perform the query are then automatically joined with the implicit foreign-key constraints. Although, as is typical for such transpilers, the resulting SQL is not human-readable, the resulting query plan is still effective and does not differ meaningfully from what would result from hand-written SQL.

MATE's structure, which consists of intent, focus, filters and operations, constitutes a domain-independent layer. The adaptation of the proposed meta layer to different data domains can be done easily by modifying the possible values for intent, focus and filters. For example, with data related to the booking-flight domain, a possible intent could be "request-booking", the focus could be "flight" while acceptable filters could be "departure", "arrive", "date" and so on. Similarly, if the domain is related to market sales, the intent will be "request-sales", the focus will be "product", while possible filters' values will be "product-type" and "store-category".

4. A case study

We implemented MATE as a conversational agent in RASA[18], an open source machine learning framework based on intent recognition and slot classification.

RASA has three main functions: 1) Natural Language Understanding, which customises and trains language models for domain-specific terms, 2) Dialogue Management, which allows training a new conversational agent-supervised machine learning and 3) Integration, which provides built-in connectors to some common messaging and voice channels, such as Facebook Messenger and Telegram, as well as to external platforms via APIs. Using RASA, we were able to easily manage question/answer interactions with the user, which was sent/received from/to the NIKY front-end. NIKY is BNova's advanced analytics platform; it is based on artificial intelligence algorithms and has been developed to improve the customer experience and simplify interactions between users and the database. The role of NIKY's front end is twofold. On the one hand, it contains a simple interface to get the request of the user and to show the response messages of RASA. The goal is to activate a simple conversation between the user and our natural language interpreter. On the other hand, it visualises query results by means of easily interpretable charts.

Suppose that the user asks "Show me the items sold in 2022 by market segment". This message is then sent to RASA, the core component of the natural language processing task. Then, using RASA NLU module, all the relevant information contained in the sentence are tagged, as illustrated in Figure 2. Next, the marked data are organised in the MATE structure and implemented as a JSON file, as described in Section 3. The resulting output will be represented

```
"intent": "request_sales",
"focus": ["product"],
"filters": {
    "measures": ["sales"],
    "year": ["2022"] },
"operation": {
    "sort_order": null,
    "count": null,
    "group_by": ["market_segment"],
    "avg": null }
```

Figure 3: An example of the population of the MATE layer with the query "Show me the items sold in 2022 by market segment".

Table 1
Weighted averaged values for the slot-filling task.

Measures	Values
Precision	0.94
Recall	0.93
F1 score	0.93
Accuracy	0.98

as shown in Figure 3.

The JSON structure is then sent to NIKY, which converts the intermediate query produced by MATE into a database language and queries the database to return the data requested and its graphical representation (e.g., a bar or a sunburst chart).

5. Evaluation

MATE has been evaluated with respect to its accuracy in the tasks of slot-filling and query composition. In this Section, we present these two evaluations and we discuss their results.

5.1. Slot-filling

To test MATE on the task of slot recognition and filling, we evaluated RASA NLU model using a dataset consisting of 101 user sentences related to the market sales domain and written by native speakers. The dataset provided by BNova was then divided into training and test sets, which include respectively the 80% and the 20% of the sentences. Based on this test set we computed precision, recall, F1 score and accuracy of the model in the recognition and extraction of the entities. The values obtained, reported in Table 1, show that the NLU model enriched with MATE is able to effectively capture the information contained within the textual input that is necessary to query the database.

Analysing the results, we noticed that the entities more frequently confused by the model are *product-type*, classified as *market-segment* (both in the role of *focus* and of *filtering slot*),

Table 2
Errors recorded in query composition.

Type of error	Occurrences
Focus mismatch	3
Missing measure	1
Implied information	1

and *customer* as *filtering slot*, which is confused with *customer* with the role of *focus*. The misclassification between *product-type* and *market-segment* it is due to the fact that *product-types* constitute a sub-class of *market-segment*; for example, "shoes" is marked as *market-segment* while "sport shoes" or "closed shoes" fall under *product-type*. As future developments of the work, we therefore intend to resolve these ambiguities and increase the training data in order to improve the correct identification of focus and filtering slots. The misclassification between *customer-slot* and *customer-focus* is due to the fact that this keyword tends to appear, within the training set, more frequently as focus than as filtering slot; for this reason the model tends to associate the keyword to the focus with a higher degree of confidence.

5.2. Query composition

After this first evaluation, we validated the quality of the meta layer, verifying how effective the information produced by MATE was to generate an SQL query. Thus, we verified how many times NIKY was able to positively answer the following question: "based on the information provided by MATE, is it possible to correctly query the DWH, answering the user's question?". MATE's accuracy in query composition is **0.73**. Analysing the results, three types of errors were recorded, as shown in Table 2. As described in the Table above, in 3 cases it was recorded a **focus mismatch**; in fact, in all the three queries it was erroneously indicated as focus the keyword *customer*. The reason why this error occurs has been already explained in the discussion of the first evaluation experiment. In addition to this, in one case a **missing measure** was found. This was due to the small number of training data; thus, the error could be resolved by implementing the training set. Finally, the last error in the translation of the query concerns the **extraction of implied information**. This error was recorded for the question "How much do customers buy online?". In this query it has been identified as focus "customer" but the real focus of the request was related to "products". However, since "product" is implicit, it is not easy to annotate the correct entity and extract this data from the textual input; the error identified in this query was therefore considered as a borderline case, which will be better investigated in future works.

The two evaluations we conducted represent only a preliminary analysis and, as such, will be extended in future works to compare MATE with other state-of-the-art solutions. Moreover, in future studies we would like to verify to what extent the questions currently used for the training and test sets are similar to real questions of typical users.

Nevertheless, the results of our preliminary evaluations are positive and show that 1) the NLU model enriched with MATE is able to accurately identify and extract the informative slots embedded in the user's request, and that 2) on the basis of the information extracted it is possible to effectively query external data warehouses.

6. Conclusion

We presented Mate, a new meta layer that provides an intermediate representation between users' questions expressed in natural language and a structured query language. Mate, developed by a joint team made up of an academic partner (UNIPI) and a business partner (BNova), leverages Natural Language Understanding techniques for sequence labelling to organise the information of the textual request into an intermediate structure which will be easily translatable to a database query. Mate is based on the recognition, within the textual request provided by the user, of key concepts such as *focus*, *operations* and *filters*. Then, from these key concepts, the relevant information for querying the database are extracted. We implemented Mate as a conversational agent integrated into RASA, and we tested the meta layer proposed by two different evaluations. The results derived from the evaluations conducted proved that our meta layer can correctly identify and extract the relevant information within a textual query, as well as be effectively used for querying an external data warehouse.

As future works we aim to make Mate adaptable to the user's refinement research; by doing so, whether the user decides to continue the search with a further question, the focus, filters, and operations will be kept in the meta layer structure and pieces of information will be then eventually updated or added. On the other hand, if the user decides not to continue with the query, it would be possible to empty the fields previously kept in memory and proceed with a completely new query. In addition to this, we will work to improve the integration between RASA, within which the meta layer has been implemented, and NIKY; this would be possible, first and foremost, by increasing the training data available, which will allow improving both system entity recognition and its performance on ambiguous and linguistically complex cases.

By increasing the performance of the NLU model enriched with Mate, it will be possible to develop a useful application that will allow users to access the information stored in databases in a simple and effective way.

Acknowledgments

Mate has being developed thanks to the partnership between the University of Pisa and BNova srl. It has been realised in the context of Text2Query, a research project partially founded by Regione Toscana (POR-FESR 2014-2020), whose aim is the development of natural language interfaces for query languages and Big Data via Deep Learning models.

References

[1] D. Nozza, L. Passaro, M. Polignano, Preface to the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), November 30, 2022, CEUR-WS.org, 2022.

[2] V. Zhong, C. Xiong, R. Socher, Seq2sql: Generating structured queries from natural language using reinforcement learning, ArXiv abs/1709.00103 (2017).

[3] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 464–468. doi:10.18653/v1/N18-2074.

[4] W. Wang, Y. Tian, H. Xiong, H. Wang, W.-S. Ku, A transfer-learnable natural language interface for databases, ArXiv abs/1809.02649 (2018).

[5] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, D. Radev, Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3911–3921. doi:10.18653/v1/D18-1425.

[6] L. Dong, M. Lapata, Coarse-to-fine decoding for neural semantic parsing, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 731–742. doi:10.18653/v1/P18-1068.

[7] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, W. Cohen, Open domain question answering using early fusion of knowledge bases and text, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4231–4242. doi:10.18653/v1/D18-1455.

[8] W. Hwang, J.-Y. Yim, S. Park, M. Seo, A comprehensive exploration on wikisql with table-aware word contextualization, ArXiv abs/1902.01069 (2019).

[9] P. He, Y. Mao, K. Chakrabarti, W. Chen, X-sql: reinforce schema representation with context, ArXiv abs/1908.08113 (2019).

[10] B. Wang, R. Shin, X. Liu, O. Polozov, M. Richardson, RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7567–7578. doi:10.18653/v1/2020.acl-main.677.

[11] K. Affolter, K. Stockinger, A. Bernstein, A comparative survey of recent natural language interfaces for databases, The VLDB Journal 28 (2019) 793 – 819.

[12] J. Richardson, R. Sallam, K. Schlegel, A. Kronz, J. Sun, Gartner Magic quadrant for analytics and business intelligence platforms, Technical Report, 2020.

[13] Salesforce, Tableau homepage, Last accessed 28 Sept. 2022. URL: https://www.tableau.com/.

[14] A. Narechania, A. Srinivasan, J. T. Stasko, NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries, IEEE Transactions on Visualization and Computer Graphics 27 (2021) 369–379.

[15] Y. Gan, X. Chen, J. Xie, M. Purver, J. R. Woodward, J. Drake, Q. Zhang, Natural SQL: Making SQL easier to infer from natural language specifications, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2030–2042. doi:10.18653/v1/2021.findings-emnlp.174.

[16] E. Smith, K. A. Crockett, A. Latham, F. J. Buckingham, Seeker: A conversational agent as a natural language interface to a relational database, 2014.

[17] D. Jurafsky, J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 1st ed., Prentice Hall PTR, USA, 2000.
[18] Rasa homepage, Last accessed 28 Sept. 2022. URL: https://rasa.com/.