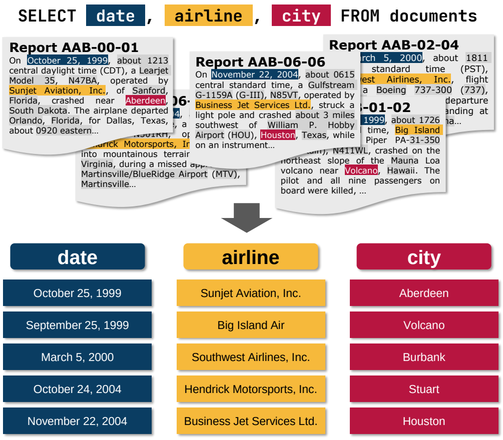
WannaDB: Ad-hoc Structured Exploration of Text Collections Using Queries

Just tell me what you want, what you really, really want!

Benjamin Hättasch

Technical University of Darmstadt (TU Darmstadt), Karolinenplatz 5, 64289 Darmstadt, Germany



date	airline	city
October 25, 1999	Sunjet Aviation, Inc.	Aberdeen
September 25, 1999	Big Island Air	Volcano
March 5, 2000	Southwest Airlines, Inc.	Burbank
October 24, 2004	Hendrick Motorsports, Inc.	Stuart
November 22, 2004	Business Jet Services Ltd.	Houston

Figure 1: Aim: Query a text collection and receive an approximate structured result without manual extraction

Motivation In many domains, users face the problem of needing to quickly extract insights from large collections of textual documents. For example, imagine a journalist who wants to write an article about airline security that was triggered by some recent incidents of a well-known US airline. For this reason, the journalist might decide to explore a collection of textual accident reports from the *National Transportation Safety Board* in order to answer questions like 'What incident types are the most frequent ones?' or 'Which airlines are involved most often in incidents?'. And clearly, there are many more domains where end users want to explore textual document collections in a similar fashion.

Yet, existing approaches to answer such queries over new text collections force users to either read through vast amounts of text and manually extract the relevant information before they can compute an answer to their query or to build extraction pipelines (e.g., when using [1]) which however require substantial efforts.

Hence, we advocate for a different route where users can extract structured data relevant to satisfy an information need from a collection of text documents without the need to program, train or specify extraction systems.

Instead, the aim is to provide a system that allows users to explore new (unseen) text collections by simply issuing a query to receive structured information from the corpus. In contrast to [2] this should not require data already in tabular form, rather the idea is to automatically identify the relevant target structure and then, again automatically, fill it from unstructured text.

Contributions Therefore, we propose WannaDB, a system for ad-hoc structured text exploration. The main idea of WannaDB is that a user specifies their information need by composing SQL-style queries over the text collection. For example, in Figure 1, the user issues a query to extract information about dates, airlines, and cities of incidents. WannaDB then takes the query and evaluates it over the given document collection by automatically populating the table(s) required to answer the query with information nuggets from the documents.

To do this, WannaDB uses a novel pipeline as shown in Figure 2 which first extracts a superset of information nuggets from texts (e.g., all named entities), then

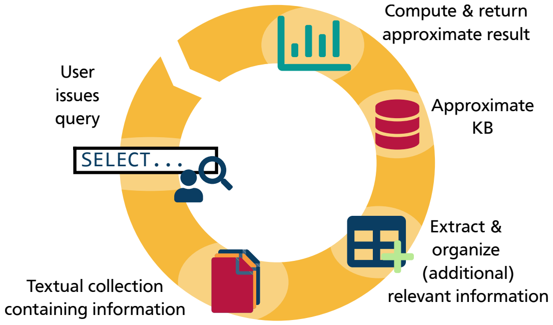

Compute & return approximate result
User issues query
SELECT...
Approximate KB
Extract & organize (additional) relevant information
Textual collection containing information

Figure 2: Pipeline & Usage of WannaDB

DESIRES 2021 – 2nd International Conference on Design of Experimental Search & Information REtrieval Systems, September 15–18, 2021, Padua, Italy

benjamin.haettasch@cs.tu-darmstadt.de (B. Hättasch)
https://benjaminhaettasch.de/ (B. Hättasch)
0000-0001-8949-3611 (B. Hättasch)

CEUR Workshop Proceedings (CEUR-WS.org)

determines the information need from the query, and finally matches nuggets to the relevant attributes of the user's query. As a result, WannaDB allows to answer the queries, even if the information is not explicitly stated in the corpus but has to be calculated (e.g., when the query contains aggregation functions like AVG or SUM). A main observation here is that in many cases a sample of extractions (i.e., a table with partially missing or incorrect values) is sufficient to produce approximate results to answer the user's query.

Architecture & Initial Evaluation WannaDB contains components to determine the information need from queries, aggregate the relevant information, and compute the actual query result. For the extraction of possibly relevant information nuggets, WannaDB relies on off-the-shelf extractors like Stanza [3]. The key contribution of WannaDB, however, is a new matching approach that uses a novel embedding space exploration algorithm incorporating interactive user feedback: The matching process is done separately for each relevant attribute. It starts by selecting information embeddings close to the attribute embedding. Afterwards, other embeddings that might be matches are searched by applying several selection rules based on the closeness of embeddings to known matches. Each candidate is presented for feedback (yes/no) to the user. The algorithm balances between exploration and exploitation to select those information nuggets for feedback that quickly allow identifying the areas in the embedding space relevant for the attribute with as little feedback as possible. This area can then be used to populate the remaining rows in the target table. Previously extracted information (and user feedback) can be reused for follow-up queries. A detailed description of the matching process can be found in [4].

Our experiments on different text-collections each focused around certain topics lead to promising results: $10 - 25$ quick iterations of feedback for each attribute (i.e., confirming whether an information nuggets belongs in a certain column of a table) sufficed for high matching scores for both textual and numeric attributes.

The Road Ahead In the next steps, we want to enlarge the scope, i.e., support more general corpora. We also want to leverage certainties from the extraction and matching process for the computation of the approximate result and provide useful interfaces for the end users, not only through a standalone application but also e.g., in form of a Jupyter Notebook extension.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research as part of the Project Software Campus 2.0 (TUDA), Microproject INTEXPLORE, under grant ZN 01IS17050, by the German Research Foundation as part of the Research Training Group *Adaptive Preparation of Information from Heterogeneous Sources (AIPHES)* under grant No. *GRK 1994/1*, as well as the German Federal Ministry of Education and Research and the State of Hesse through the National High-Performance Computing Program.

References

[1] C. De Sa, A. Ratner, C. Ré, J. Shin, F. Wang, S. Wu, C. Zhang, Deepdive: Declarative knowledge base construction, SIGMOD Rec. 45 (2016) 60–67. URL: https://doi.org/10.1145/2949741.2949756. doi:10.1145/2949741.2949756.

[2] S. Zhang, K. Balog, Ad hoc table retrieval using semantic similarity, in: Proceedings of the 2018 World Wide Web Conference, WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, p. 1553–1562. URL: https://doi.org/10.1145/3178876.3186067. doi:10.1145/3178876.3186067.

[3] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, ArXiv abs/2003.07082 (2020).

[4] B. Hättasch, J.-M. Bodensohn, C. Binnig, Aset: Adhoc structured exploration of text collections, in: 3rd International Workshop on Applied AI for Database Systems and Applications, Copenhagen, Denmark, 2021. URL: https://sites.google.com/view/aidb2021/home/accepted-papers.