NLP Relational Queries and Its Application

Andrei Stoica
Ontario Tech University
2000 Simcoe Street North
Oshawa, Ontario L1G 0C5
Email: andrei.stoica@ontariotechu.ca

Ken Q. Pu
Ontario Tech University
2000 Simcoe Street North
Oshawa, Ontario L1G 0C5
Email: ken.pu@ontariotechu.ca

Heidar Davoudi
Ontario Tech University
2000 Simcoe Street North
Oshawa, Ontario L1G 0C5
Email: heidar.davoudi@ontariotechu.ca

***Abstract*—Recent advances in natural language processing have shown the effectiveness of statistical and neural network-based algorithms in a deep understanding of textual data. We demonstrate that the result of NLP analysis on text documents can enrich relational data in a way so that structured queries can be used to derive further value from text data.**

In this paper, we present how we can perform analytics on a scientific research dataset based on both the relational data and NLP topic modeling. The integrated NLP features together with the classical relational query constructs allow one to explore the topic structure of the DBLP dataset with flexibility and precision.

I. Introduction

We are motivated by advances in two areas: natural language processing (NLP) and data analytics. In the area of text mining, we have seen highly effective approaches such as LDA [1] for topic modeling and transformers [2] for language modeling. On the other hand, the field of data analytics has witnessed equally and yet quite different advancements in systems [3], [4] that support Internet scale analytics. Given the amount of text-based data in structured databases, there are many benefits to augment relational database engines with NLP specific operators.

In this paper, we propose several NLP operators to relational queries so that one can harness the power of both worlds. The NLP toolbox is vast. In this paper, we limit our work to lexical analysis, topic modeling and finally, next-word prediction. These operators are added to the relational query language.

To demonstrate the effectiveness of our approach, we have analyzed the entire DBLP [5] datasets, containing over 2.5 million Computer Science related publications from 1959 to 2020. The DBLP dataset is highly structured with entities of persons, journal/conferences and publications, which are connected via different relations such as authorship and publication venues.

Using query languages such as the Structured Query Language (SQL), one can perform statistical and quantitative analytics on the datasets. However, there are many queries of interest that require some level of NLP understanding of the corpus, and they cannot be simply expressed in SQL.

Like many datasets, a large portion of DBLP consists of text data in the form of natural language, namely the paper titles. Consider some typical publications are shown in Figure 1. The three papers are on different topics, and some are more relevant to the recent surge of interest in deep learning. One may argue that the paper titled "*Neural Network Architecture Designed for High-Speed and Complex Pattern Recognition*" is arguably more relevant publication given the keyword *Neural Network* that appears in the title. We observe that such analysis is lacking in the traditional query constructs, and would require additional query operators to enable text processing at the natural language level.

II. NLP Extension to Relational Databases

In this section, we describe the NLP extension to the relational data model. We also demonstrate how the relational operators are integrated into the relational operator.

A. Basic NLP Constructs

We need to enrich the representation `text` data type to support reasoning at the NLP level.

A s : `text` is an arbitrary piece of text, e.g.

$s =$ *Scale Equalization of Higher-order Neural Networks.*

A x : `token` is an indivisible string that we use for analysis. For example, $x =$ *Networks* is a token in s. However, NLP typically analyzes the data in terms of *normalized* tokens. So, we may choose to normalize x to its root $\hat{x} =$ network by removing the plural form and the capitalization. Some tokens such as $y =$ *of* may be omitted altogether in the NLP analysis.

The tokenization can be represented by a function:

$$\texttt{tokenize} : \texttt{text} \rightarrow \texttt{token}+$$

We assume two special tokens:

- $\langle$`start`$\rangle$: is the start of a text data.
- $\langle$`end`$\rangle$: is the end of a text data.

Note: We denote non-empty finite sequences of tokens as `token`+. This can be implemented in relational databases as `Array`[`token`].

It's possible that `tokenize` performs both token normalization and filtering.

While analysis of data at the sentence-level is too coarse, and at the `token`-level to granular, we may form *phrases* as an intermediate unit of analysis. A phase is a (short) sequence of one or more tokens:

$$\texttt{phrase} = \texttt{token}+$$

We note that a `phrase` may not be a substring of `text` data because there may be token normalization and filtering performed.

There are many methods from NLP literature in dealing with phrases. *Language models* are models that evaluates

978-1-7281-1054-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IRI49571.2020.00064

Authorized licensed use limited to: Ontario Tech University. Downloaded on November 10,2026 at 19:36:52 UTC from IEEE Xplore. Restrictions apply.

Adaptive Function of Utility Driven Distribution for Mobile Agent Scheduling.	IRI	2001
Discovery of Temporal Knowledge from Databases.	IRI	2001
Neural Network Architecture Designed for High-Speed and Complex Pattern Recognition.	IRI	2001

Fig. 1: Sample paper titles in IRI 2001

the probability of occurrences of tokens in some contexts. Typically a context is a preceding phrase of the token to be predicted, but Word2Vec [6] also considers other contexts. A language model is represented by a function:

$$\texttt{likelihood} : (\texttt{phrase}, \texttt{token}) \rightarrow \mathbb{R}$$

where $\texttt{likelihood}(w)$ is the probability distribution over all tokens that can follow a phrase w. For our purpose, we have trained a simple feed-forward neural network language model [7] to perform next-token prediction with a MLP+softmax activation. The training is done offline, but the inference is performed during the query-time. We note that more complex network architectures such as LSTM are also suitable for learning a language model.

B. Topic Models

Topic modeling [1] has been widely applied to text analysis. A topic model is derived from collection of documents, with each document being represented as a bag of tokens. A topic model with k topics is a function that map documents to a mixture of the k topics.

$$\texttt{topicmodel} : \texttt{token}+ \rightarrow \mathbb{R}^k$$

where k is the number of topics, and for $\vec{v} = \texttt{topicmodel}(D)$, we have $\sum_{i=0}^{k} v_i = 1$. Each coefficient of v_i is the topic weight for topic i.

There are several algorithms for constructing the topic model. The most widely used is the Latent Dirichlet Allocation (LDA) [1]. LDA relies on a generative model. It is assumed that tokens are sampled according to a Dirichlet-distributed topic-word distribution.

We have applied LDA to generate the topic model. For LDA, it also provides token-topic weights as:

$$\texttt{topicmodel} : \texttt{token} \rightarrow \mathbb{R}$$

where $\sum_{x:\texttt{token}} \texttt{topicmodel}(x) = 1$.

C. NLP + SQL processing of DBLP dataset

In this section, we demonstrate the fluidity of performing NLP enriched relational data analytics. The combination of NLP and SQL provides the user a powerful declarative language to perform ad-hoc and yet complex processing of text data. We illustrate with examples of a typical workflow.

Augmenting the relational model

We define the user-defined custom types: `token` and `phrase`. We also define the NLP row-level functions:

- `tokenize` : `text` $\rightarrow$ `Array`[`token`] performs tokenization with normalization and filtering
- `likelihood` : (`phrase`, `token`) $\rightarrow \mathbb{R}$ computes the likelihood of next word.

For topic modeling, we have two functions: model construction, and model inference. Model construction is a table function

$$\texttt{LDA} : \texttt{Table}(\texttt{phrase}) \rightarrow \texttt{Table}(\texttt{token}, \mathrm{Integer}, \mathbb{R})$$

Model inference is done at row level:

$$\texttt{topicmodel} : \texttt{phrase} \rightarrow \texttt{Array}[\mathbb{R}]$$

Build the LDA topic model

```
CREATE VIEW corpus AS
  SELECT id, tokenize(title) as tokens
  FROM inproceedings

CREATE VIEW lda(token, topic, weight) AS
  LDA(corpus)
```

Topic inference on text

The constructed topic model allows us to obtain the topic mixtures from any tokenized text. Together with the power of relational algebra, we can generate topic vectors on ad-hoc selection of the publication data.

```
SELECT id, topicmodel(tokenize(title))
FROM inproceedings
WHERE booktitle = 'IRI'
  AND year > 2000
```

Topic aggregation

We can perform aggregation and time-series analysis of the topic models for free thanks to the power of SQL.

```
SELECT year,
       SUM(topicmodel(tokenize(title)))
FROM inproceedings
WHERE booktitle = 'IRI'
GROUP BY year
```

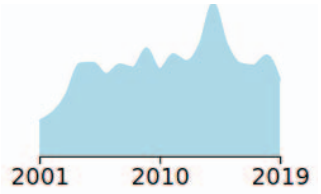
D. Topic labeling

Labeling topics is always a challenging task. In this section, we demonstrate that unsupervised topic bigram labelling can be done with the NLP+SQL. The objective is the find the most appropriate bigrams to label each of the topics. We use both the topic model as well as a language model function `likelihood` to construct the most suitable bigrams per topic.

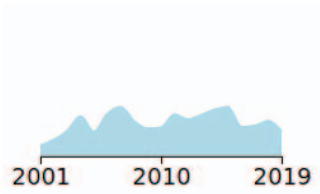
First, we can generate the top-10 tokens for each topic based on their topic weights.

```
CREATE VIEW keyword AS
 SELECT DISTINCT token
 FROM
 (SELECT token,
   rank() OVER (PARTITION BY
          topic ORDER BY weight DESC) as i
   FROM LDA(corpus))
 WHERE i <= 10
```

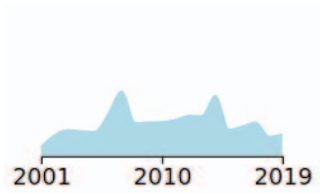

Authorized licensed use limited to: Ontario Tech University. Downloaded on November 10,2026 at 19:36:52 UTC from IEEE Xplore. Restrictions apply.



(a) Topic 19: data mine; model analysis; model approach

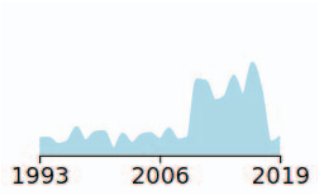


(b) Topic 2: automatic generation; natural language; model analysis

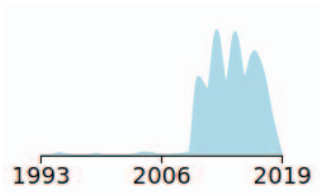


(c) Topic 10: video stream; knowledge graph; scalable video

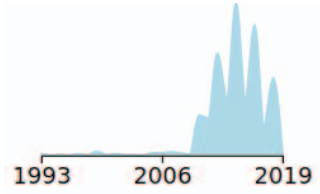
Fig. 2: Top-3 topics for IRI conference



(a) Topic 19: data mine; model analysis; model approach



(b) Topic 17: extended abstract; deep learning; machine learning



(c) Topic 13: case study; computer system; computer environment

Fig. 3: Top-3 topics for ICCS conference

We can generate bigrams by performing maximal likelihood estimation using the language model.

```
WITH V1 AS (
 SELECT S.token AS x,
        T.token AS y,
        likelihood(S.token, T.token) AS p
 FROM keyword S, keyword T
), V2 AS (
 SELECT x, MAX(p)
 FROM V1
 GROUP BY x
)
CREATE VIEW bigrams
 SELECT [x, y] AS phrase
 FROM V1 JOIN V2 ON (x, p)
```

Finally, we pick the top bigrams in each topic by selecting the top 3 bigrams based on their relevance score to the topic.

```
SELECT topic,
       phrase,
       rank() OVER (PARTITION BY topic
                    ORDER BY weight DESC)
FROM
 (SELECT phrase,
         unnest(array_range(0, K)) as topic
         unnest(topicmodel(phrase)) as weight
  FROM bigrams)
WHERE rank <= 3
```

III. DBLP Analysis and Visualization

We applied the proposed framework to analyze the DBLP dataset. The flexibility and ease of the declarative NLP+SQL language allows us to rapidly gain insights and generate meaningful visualization on many different aspects of the dataset.

We have included selected plots illustrating the type of analysis possible: ranging form topics comparison of different conferences to individual paper or author topic distribution, both at the individual paper level (Figure 5), as well as at the aggregated level (Figure 4a, Figure 4b).

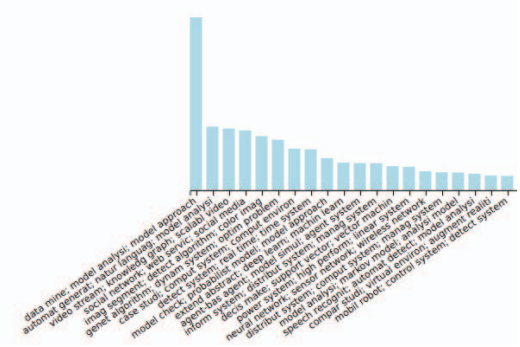
IV. Related Works

AMETHYST [8] is an existing system that employs NLP to organize and help find topics in heterogeneous data. It focuses on the organization of topics and keywords into hierarchies, while our system focus on the temporal trends of topics. In contrast Itoh et al. [9] propose a system to visualize the change in topics of a blog over time. Their approach visualizes a slice of time on one plane and a stack of these planes show the change of topics across time.

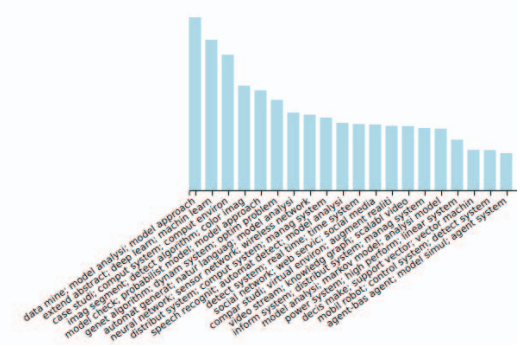
Tang et al. [10] proposed a system to compute similarity between two domains. Using the same dataset they track both changes in similarity and co-authorship between domains. They manually identified time periods and domains for the segmentation of data. Song et al. [11] use clustering to identify trends in bioinformatics field. Using the same dataset, they partition it into 4 non-overlapping periods and use Markov Random Field-based topic clustering for topic extraction. They analyse both within-period and between-period cluster similarity. Doumit et al. [12] apply LDA along with NLP to extract the bias of news sources. Through this combination of LDA and NLP they show that you can discern each news outlets bias in their coverage of an event and identify the themes that where presented by their coverage. Mei et al. [13] take a probabilistic approach to identifying the evolution of a theme over time. They us a probabilistic mixture model to extract themes and Hidden Markov Models to model the transition between themes.

Jelodar et al. [14] provide a comprehensive review of LDA topic modeling methods. Their work covers many methods for parameter estimation, as well as applications and adaptations for specific domains. One such adaptation is to deal with short

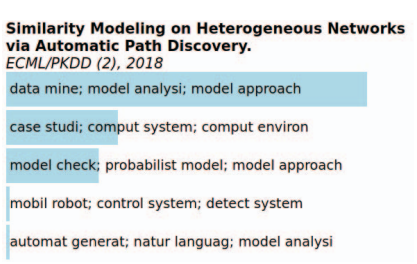
Authorized licensed use limited to: Ontario Tech University. Downloaded on November 10,2026 at 19:36:52 UTC from IEEE Xplore. Restrictions apply.



(a) Aggregated topic distribution for IRI



(b) Aggregated topic distribution for ICCS



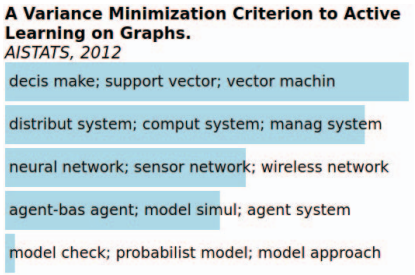


Fig. 5: Top 5 topics for 2 papers

documents such as Tweets and others address other issues such as enriching topic vectors with timestamped data.

V. Conclusion and Future Work

We presented a framework that embeds natural language processing to the relational model. We integrated language modeling and topic modeling to the SQL query language. The combined language supports powerful and flexible ad-hoc text analytics. We have utilized this framework to gain some insights through data analysis and visualization of the DBLP scientific publication dataset.

The future research will be focused on additional NLP algorithms including semantic analysis and deep language understanding with neural networks.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

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

[3] M. Armbrust, D. Bateman, R. Xin, and M. Zaharia, "Introduction to spark 2.0 for database researchers," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 2193–2194.

[4] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, "Dremel: interactive analysis of web-scale datasets," *Communications of the ACM*, vol. 54, no. 6, pp. 114–123, 2011.

[5] M. Ley, "The dblp computer science bibliography: Evolution, research issues, perspectives," in *International symposium on string processing and information retrieval*. Springer, 2002, pp. 1–10.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[7] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*. Association for Computational Linguistics, 2012, pp. 20–28.

[8] M. Danilevsky, C. Wang, F. Tao, S. Nguyen, G. Chen, N. Desai, L. Wang, and J. Han, "AMETHYST: a system for mining and exploring topical hierarchies of heterogeneous data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*. Chicago, Illinois, USA: ACM Press, 2013, p. 1458. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2487575.2487716

[9] M. Itoh, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa, "Analysis and visualization of temporal changes in bloggers' activities and interests," in *2012 IEEE Pacific Visualization Symposium*, Feb. 2012, pp. 57–64, iSSN: 2165-8773.

[10] X. Tang, C. C. Yang, and M. Song, "Understanding the evolution of multiple scientific research domains using a content and network approach," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 5, pp. 1065–1075, 2013, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22813. [Online]. Available: http://onlinelibrary.wiley.com/doi/abs/10.1002/asi.22813

[11] M. Song, G. E. Heo, and S. Y. Kim, "Analyzing topic evolution in bioinformatics: investigation of dynamics of the field with conference data in DBLP," *Scientometrics*, vol. 101, no. 1, pp. 397–428, Oct. 2014. [Online]. Available: http://link.springer.com/10.1007/s11192-014-1246-2

[12] S. Doumit and A. Minai, "Online News Media Bias Analysis using an LDA-NLP Approach," p. 16.

[13] Q. Mei and C. Zhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ser. KDD '05. Chicago, Illinois, USA: Association for Computing Machinery, Aug. 2005, pp. 198–207. [Online]. Available: https://doi.org/10.1145/1081870.1081895

[14] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15 169–15 211, Jun. 2019. [Online]. Available: https://doi.org/10.1007/s11042-018-6894-4

Authorized licensed use limited to: Ontario Tech University. Downloaded on November 10,2026 at 19:36:52 UTC from IEEE Xplore. Restrictions apply.