Matching Metadata on Blockchain for Self-Sovereign Identity

Frederico Schardong[1,2(✉)], Ricardo Custódio[1], Láercio Pioli[1], and João Meyer[1]

[1] Universidade Federal de Santa Catarina, Florianópolis, Brazil
ricardo.custodio@ufsc.br, {laercio.pioli,joao.meyer}@posgrad.ufsc.br
[2] Instituto Federal do Rio Grande do Sul, Campus Rolante, Rolante, Brazil
frederico.schardong@rolante.ifrs.edu.br

Abstract. Self-Sovereign Identity (SSI) is a privacy-preserving identity paradigm where users own and manage their digital identities. SSI is also referred to as blockchain identity, as it is commonly implemented using distributed ledger technologies. In this work, we describe the problem of schema matching on blockchain-based SSI implementations, systematically review the literature for tools that attack this problem, introduce a novel solution, and empirically compare it with the works reported from our literature review. Our solution uses pre-trained word vectors to find semantic similarities between user queries and schemas on the blockchain. Experimental evaluation shows that it outperforms existing solutions regarding queries that approach natural language.

Keywords: Self-Sovereign Identity · SSI · Blockchain · Schema matching · Metadata matching

1 Introduction

Privacy is perhaps the biggest concern that people have when using electronic services on the Web. Ensuring that information is securely transferred, stored, and processed by Service Providers (SPs) are complex problems that have been studied for decades by computer scientists. In this context, manipulating personal data is critical and requires even more care, as it can cause irreparable damage to their owners if unduly revealed or altered. To this end, a new identity management model named Self-Sovereign Identity (SSI) [2] was introduced based on the idea of reducing the presence of Identity Providers (IdPs) by having users storing and managing their data. This new identity model is also called blockchain identity as it is commonly implemented using distributed ledger technologies [10]. The use of blockchain provides high availability, immutability, and no centralized authoritarian control for identity.

This study was supported by the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS).

A. Marrella and B. Weber (Eds.): BPM 2021 Workshops, LNBIP 436, pp. 421–433, 2022.
https://doi.org/10.1007/978-3-030-94343-1_32

The objective of this work is to study the problem of matching metadata on blockchain-based SSI systems. Generally, a blockchain stores transactions describing new users joining the network, metadata[1] describing how users should organize their data to be exchanged with others, among other operations.

As a study case, we adopt Sovrin [18], one of the first SSI offerings made available to the general public and perhaps the most studied [10,11,14]. However, for the use of Sovrin, there are some challenges. The main one is related to the monetary cost. It happens that most transactions are free, but some have a monetary cost to be handled. Registering a new block describing a new schema is currently the most expensive, costing 50 USD [19]. Therefore, reusing existing schemas reduces the capital expenditure of users. Figure 1 presents the sequence of actions in a toy example where a Human Resources (HR) company joins the ledger and publicizes that it wants to receive Curriculum Vitae (CV) in a specific format. Due to the blockchain's structure, where one block is placed after another, searching if similar metadata was published in the ledger is not a trivial task. One would have to go through all the blocks and inspect their contents. Manually matching schemas is not only time-consuming but also error-prone.

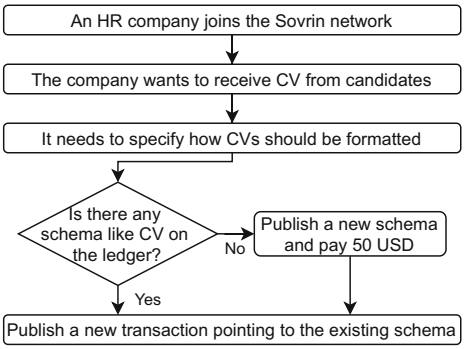

Fig. 1. Action flowchart of an HR company from joining the Sovrin network to publishing a schema that describes how candidates should format their CV.

Our contribution in this work is two-fold: (i) systematically review the blockchain-based SSI scientific literature to identify and investigate which research materials have been published that present any proposal, technique, or tool for retrieving metadata from blockchain-based identity solutions; and (ii) based on the analysis of these existing proposals, we introduced a novel tool to perform metadata matching on Sovrin that outperforms existing solutions by finding semantic similarities between user queries and schemas on the blockchain. Additionally, it can be easily expanded to other blockchain-based SSI systems. The rest of this paper is organized as follows. In Sect. 2 we provide the fundamentals of SSI needed to follow this study. Next, in Sect. 3 we detail the

[1] The terms metadata and schemas are used interchangeably in this paper.

systematic literature review, while in Sect. 4 we describe our proposed solution, and Sect. 5 presents some experiments carried out with our proposal showing that it outperforms current solutions. Finally, in Sect. 6 our final observations are made.

2 Self-Sovereign Identity

Perhaps the most naive and straightforward way to authenticate users of electronic services on the Web is to register and establish access credentials for each service. In this model, each SP has its own private identity system. Therefore, users have the arduous task of accessing a new service, registering, and defining their access credentials. In this model, the user is responsible for memorizing their access credentials for each of the services. This model is suitable when users use a few services. However, if the number of services is high, users tend to reuse their credentials, posing security challenges [7]. Likewise, the greater the number of services, the greater the user's inability to keep track of their sensitive data. In this scenario, users tend to forget which services keep their data. If data leaks, it is challenging to know which SP was responsible for the problem.

One way to minimize these difficulties, inherent to the one identity per service model, is using a specialized service, where identity management tasks, such as registration and authentication, are performed by an IdP [8]. In this model, users are no longer registered with SPs. Thus, whenever a user needs to authenticate to a service, the SP redirects to their identity provider. Once the user's IdP authenticates it, the SP is communicated that the user is whom they claim to be [16]. From a user perspective, having just one username and password for all SPs dramatically reduces the risk of incorrectly managing access credentials and improves security. However, the IdP-based model creates silos of personal information, which are high-value assets for attackers.

In practice, however, it is common to have many identity providers. Users register with one or more of these providers, those they trust. When the number of identity providers increases, it is reasonable to organize them in a federation [4]. This federation, using some "where are you from" scheme, allows the user to be associated with an IdP, which the SP accepts. So, instead of having multiple options for IdPs in service, there would be only one link to the federation.

The manipulation of private and sensitive user data by IdPs can raise privacy issues. Recently, the possibility of users keeping their identity with themselves has been discussed [10]. In this model, the user holds credentials with its attributes issued by the IdPs it is associated with. Although the user can be its own IdP, SPs might not trust self-issued credentials. Nonetheless, credentials are said to be anonymous [3], as they are not directly shared with SPs. Instead, the user presents mathematical proofs regarding the data contained in credentials, such as being older than 18 years old, along with proof that the IdP did not revoke this credential. This model is known as Self-Sovereign Identity (SSI) [2].

The advantages of SSI are manifold. We list a few: (i) users are not tracked by IdPs, because SPs do not contact IdPs to get users' data; (ii) users have total

control of their data, revealing only the minimum necessary to those who wish, thus helping to avoid that their sensitive information is misused or sold; and (iii) multiple self-sovereign digital identities can be created, allowing the use of one identity for each transaction, anonymizing the owner of these identities.

Most SSI systems are implemented using distributed ledger technology (*i.e.* blockchains) because of their immutability and fault tolerance [10,14]. As data stored in a blockchain cannot be changed, credentials or other critical information are usually not saved on-chain. What goes on the ledger are revocation registries (or pointers to the registries), end-points for communication, and schemas describing what information each type of credential has [18]. The latter is the subject of this research. Effectively searching and finding registered schemas to reuse them in new credential formats results in not needing to register new schemas, which is the most expensive kind of operation on Sovrin [19].

3 Systematic Literature Review

The main idea of secondary studies is to follow a systematic method to gather evidence and answer specific research questions [9]. In this paper, we conduct a Systematic Literature Review (SLR) to discover the state of the art regarding searching schemas on blockchain-based SSI.

3.1 Research Method

The review protocol we followed has five steps [9]: (i) research questions definition; (ii) search strategy for primary studies; (iii) definition of inclusion criteria; (iv) classification of the papers; and (v) data extraction.

To accomplish the first step, we follow the suggestion of Petticrew and Roberts [15] and adopt the PICOC (Population, Intervention, Comparison, Outcome, Context) guideline to construct the research question. *Population* refers to a set of elements that we are investigating, blockchain. *Intervention* refers to the element that addresses the study, schema matching. *Comparison* seeks to compare the intervention with some element, which is not used in this study. *Outcome* describes the obtained results, including a practice point of view of them. Our study uses the terms technique, technology, strategy, method, approach, and solution. Finally, *context* is the context in which the comparison takes place, self-sovereign identity. As a consequence of the PICOC guideline, we define the research question guiding this SLR as follows.

Which techniques and technologies were used to search or match schema in blockchain-based self-sovereign identity solutions?

Next, our search strategy consists of using the terms defined in the PICOC guideline in addition to synonyms to create the search string. It is important to note that when running the search string, we noticed that the terms referring to

the *outcomes* reduced considerably the number of papers returned by the search libraries. Therefore, we omitted the *outcome* terms to increase the number of papers. The final search string is presented below.

(blockchain **OR** ledger)
AND
(ontology **OR** retrieve **OR** matching **OR** similarity **OR** crosswalk **OR** mapping)
AND
(self-sovereign **OR** self-sovereignty **OR** self sovereign **OR** self-sovereignty **OR** decentralized identity **OR** decentralised identity **OR** distributed identity)

The third step is to define a set of inclusion criteria. These criteria aim to select research studies that fit the research question. In this SLR, we defined three inclusion criteria: (i) IC1: articles written in English; (ii) IC2: articles that necessarily have a title and abstract; and (iii) IC3: articles that propose a technique to search or match schemas in blockchain-based identity management solutions. Papers should meet all IC to be selected for data extraction.

The fourth step, namely the classification of papers, was not performed as any article that meets all IC has equal value in this SLR. Lastly, the data extraction step consisted of reading the selected papers and identifying: (i) what algorithm/technique/technology was used to carry out the schema matching; and (ii) if the proposed solution is an algorithm, tool, framework, or other.

3.2 Execution

We conducted our search string on ACM Digital Library, Engineering Village, IEEE Xplore, ScienceDirect, Scopus, and SpringerLink on March 28, 2021, and retrieved a total of 34 primary studies. All the libraries were configured to execute the search string considering only the metadata fields (*i.e.* title, keywords, and abstract). However, in the ACM library, we had to run the search string only on the abstract field, and in Science Direct, we searched on the entire paper. In the second stage, we identified and removed 11 duplicated articles. Next, our IC were applied. As a result, IC1 removed no paper, IC2 removed four books of proceedings. Finally, we read all the 19 papers' metadata (*i.e.* keyword, title, and abstract) and applied IC3, which discarded 18 articles.

The remaining paper was read and analysed [12]. Then, to increase the scope of our SLR, a snowballing process was carried. We reviewed its references (backward snowballing) and other articles that cited it (forward snowballing) to identify other potential primary studies. Fifty-five references were reviewed in the backward and forward snowballing. Two new works [17,21] were included in our set of final results. Figure 2 depicts the execution of our SLR protocol.

3.3 Report

With regards to the research question driving this SLR, our findings are summarized in Table 1. It lists the tools systematically discovered and what technologies

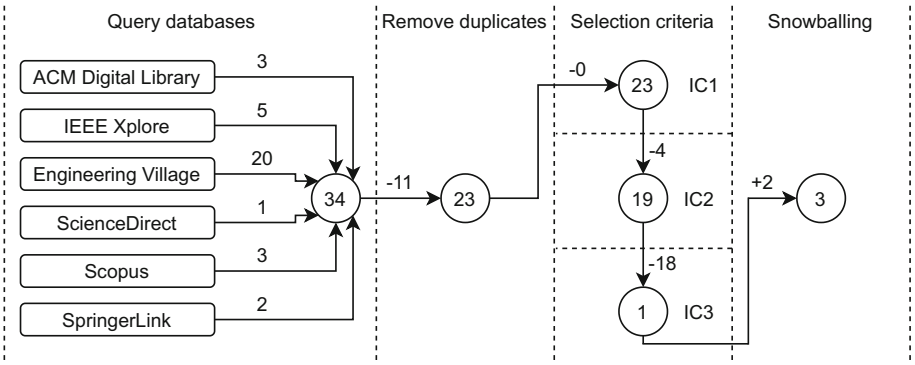

Fig. 2. Number of articles in each stage of the execution of our protocol.

were used to search for information in the blockchain. Furthermore, we included our tool in the last row for comparison and detailed it in the next section. Next, we report the three tools found in this SLR.

Table 1. Comparison of the works found in our SLR and this work.

Work	Technique	Proposal	Search technology
[12]	Full-text Search	Tool	Apache Solr
[17]	Full-text Search	Tool	Elasticsearch
[21]	Basic Search [12]	Tool	Apache Solr
This work	Semantic Search	Tool	spaCy

The authors of [12] proposed a tool to perform a full-text search on Hyperledger Indy, which is the blockchain technology underlying Sovrin. This tool performs schema matching based on textual input. It stores a local copy of the ledger on a database and integrates it with Apache Solr[2] to do the matching.

Next, the authors of [17] proposed a tool, named Indyscan, that stores the transactions on a local database and uses Elasticsearch[3] to search for information on the Sovrin ledger based on user-inputted search terms. According to [12], Indyscan uses MongoDB for storing the blockchain's transactions. Moreover, an HTTP API wrapper was developed to display the ledger transactions as HTML pages and allow users to execute the searches.

Finally, the last work found in our SLR [21] also implemented an HTTP API to display and search for transactions on the Sovrin ledger. According to [12], this application performs a "basic search". For instance, when searching for a transaction with a person's name, which is the case of the first transaction of

[2] https://solr.apache.org/.
[3] https://www.elastic.co/.

the ledger, any changed letter failed to find it. This tool also uses Apache Solr to perform the searches, and the transactions are stored in a PostgreSQL database.

4 Semantic-Based Schema Matching

In this section, we describe our proposed solution to the problem of matching schemas in blockchain-based SSI. We use as a study case the Sovrin ledger, which is the first publicly available SSI system [18,20]. The Sovrin ledger is a public-permissioned blockchain where only registered members can write to it, *i.e.*, create digital identities, and declare credential formats, but anyone can download and read them [20].

Following the design principles introduced by the authors of [12], which pointed out that sending requests to an online ledger for every system query could impact the performance of the system, we maintain a local copy of the ledger to perform schema matching. As of April of 2021, there are almost 60 thousand transactions registered in the Sovrin ledger. However, only 149 of these are transactions that register new schemas. Therefore, our prototype's persistent storage has less than 100 kilobytes, as only transactions registering schema are persisted. To ensure that the local storage is up-to-date, a pooling service was created and is executed at a constant interval.

Our approach to attack the schema matching problem consists of using a natural language processing tool named spaCy [6]. Through the usage of pre-trained word-vectors [13], semantic similarities between user-inputted terms and existing schemas in the blockchain can be calculated. For instance, a pre-trained word-vector of the English language is likely to find a high similarity between the terms *surname* and *last name*. This sort of matching enables our solution to find similar schemas with high accuracy. We used spaCy because it outperforms other offerings [1] and has detailed documentation and active development [6].

Figure 3 shows an overview of our solution. Users can query existing schemas by inputting either natural language queries or presenting schemas formatted in JSON, as represented in Fig. 3 by the queries of Alice and Bob to the Command Line Interface (CLI). Both the pooling service and CLI are built using Python 3 and embedded into Docker containers. The transactions are stored in an instance of RocksDB, a persistent key-value store optimized for low latency queries [5]. Our implementation is open-source and publicly available[4].

For every query made to the CLI, the three most similar schemas are returned by default along with their score of similarity (calculated by spaCy) and transaction number as referenced in the Sovrin ledger. Two examples are provided: (i) the output of the query "money" is shown in Table 2; and (ii) the results for the JSON-formatted query `["student", "university", "degree"]` is shown in Table 3. These two concrete examples help illustrate the usefulness of semantic search. On the results of the former query, the term `money` is only present in the first result, which is the only schema in the ledger at this point to have

[4] https://github.com/fredericoschardong/sovrin-schema-matching/.

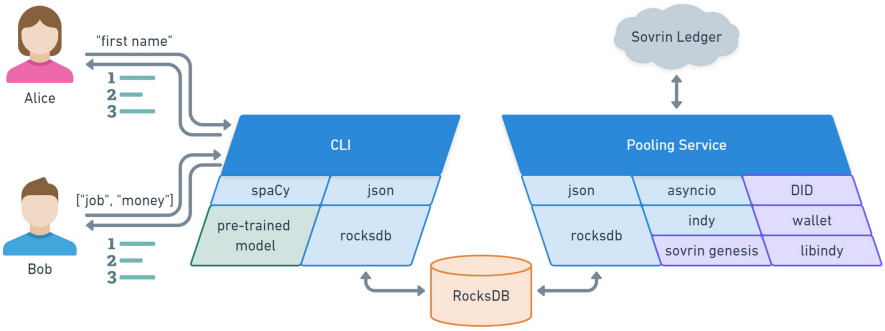

Fig. 3. An overview of the proposed solution. The CLI and the pooling service are written in Python 3, where blue parallelograms represent the modules. On the CLI, a green parallelogram shows the pre-trained model of the English language made available by spaCy. The purple parallelograms of the pooling service (DID, wallet, etc.) represent Sovrin-related components.

this term. Nonetheless, the other two schemas have related terms such as `loan amount`, `interest rate`, `credit score`, `total deposits`, and others. Regarding the results of the latter query, the first and second schemas have the same score as they are composed of the same terms but arranged in a different order. None of the returned schemas have the terms `student` or `university`, but all of them have `degree` and other fields related to education such as `institution` and `GPA`.

Table 2. Results for the query "money"

Score	Trans. #	Schema values
0.595	59556	`State, Listing Agent, Lot Number, Buyer First Name, Contract Signed Date, Purchase Price, Postal code, Buyer Last Name, Subidivision, City, Street Address, Buyer Agent, Estimated Completion date, Model Name, Earnest Money`
0.590	59555	`Application Status, Loan Number, Loan Amount, Date of Approval, First Name, Subdivision, Lot Number, Last Name, Earliest Closing Date, Loan Term in Months, APR, Interest Rate, Lender Name`
0.569	59551	`Credit Score, Account Type, Institution Name, DOB, Total Deposits, SWIFT BIC, IBAN, Last Name, First Name, Statement Period, Average Montly Balance Last 12 months, Total Withrawwals, Account Number`

Table 3. Results for the query `["student","university","degree"]`

Score	Trans. #	Schema values
0.660	54788	`degree, last_name, axuall_proof_id, institution, status, year, first_name`
0.660	54802	`first_name, institution, axuall_proof_id, last_name, degree, year, status`
0.620	33627	`DEMO-GPA, DEMO-Major, DEMO-Degree, DEMO-College Name, DEMO-Student Name`

5 Empirical Evaluation

In this section, we describe the method developed to conduct experiments and compare our tool with the solutions found on the SLR, then report the results of its execution and conduct a performance analysis.

5.1 Experiment Plan

We have defined three schema matching queries to evaluate our proposed solution and how it compares to the tools in the literature. For each query, a set of correct schemas, *i.e.* schemas that the evaluated technique should return, were manually selected by the fourth author and revised by the first author.

The first query is "address", and the 32 schemas that contain the terms `address`, `country`, `state`, `street`, `zip`, or `city` were manually selected. Next, the second query is "first name". The correct results are the 67 schemas that include `first name`, `given name`, `member name`, `full name`, or `student name`. The reasoning behind these two queries is to evaluate how our solution compares to the existing literature with regards to straightforward queries that might not gain significant advantage using a semantically aware search algorithm. On the other hand, our third query is "company job", and aims to evaluate the gains of performing semantic-aware searches. Correct results included the 23 schemas with any of those two terms or `organization`, `employer`, `department`, `role`, `contract`, or `payment`. All manually selected schemas were not chosen a priori, but rather during the selection of correct schemas on May 16, 2021.

To measure and compare the performance of our tool with the related work regarding those three queries, we have used the f-score, which is the harmonic mean of precision and recall. A confusion matrix for each query and tool was created to calculate those measurements, where the gold standard is the correct schema matching following our manual selection. Regarding our tool, only the results with a score greater than 0.6 were considered schema predictions.

5.2 Experimental Evaluation

The experimental evaluation happened on May 16, 2021, when the Sovrin ledger had 149 schemas. We ran the three queries on [17] and [21] through their publicly

accessible websites and in our tool. It was not possible to empirically evaluate the work of Lux *et al.* [12] as their solution is not publicly available.

Having the correct and incorrect classification of each tool, we used `sklearn`'s `classification_report`[5] to assemble the confusion matrices, which are omitted for brevity, and to calculate the f-score. The measurements are made for the two prediction classes: correct and incorrect prediction of schemas, respectively abbreviated to `True` and `False`. Table 4 presents the f-score for each class and their average, along with the support, which is the absolute value of schemas in the `True` and `False` classes.

Table 4. The number of correct and incorrect schemas based on our manual selection (support) and the f-score of predictions.

Query	Technique	Support		f-score		
		True	False	True	False	Avg.
"address"	[12]	36	112	–	–	–
	[21]			0.65	0.92	0.79
	[17]			0.46	0.90	0.68
	This work			0.29	0.84	0.56
"first name"	[12]	67	81	–	–	–
	[21]			0.66	0.83	0.74
	[17]			0.48	0.78	0.63
	This work			0.76	0.84	0.80
"company job"	[12]	23	125	–	–	–
	[21]			0.00	0.92	0.46
	[17]			0.00	0.92	0.46
	This work			0.55	0.94	0.74

The existing tools produced better results (*i.e.* higher f-score) for the single-word query "address". Our solution returned few and incorrect predictions, which resulted in a low f-score of 0.29 for the `True` class. However, with regards to the query "first name", in which many of the correct schemas do not have the terms `first` and `name`, our tool had a precision of 0.88 and recall of 0.67, thus resulting in an f-score of 0.76 for the `True` class, surpassing the other offerings. Moreover, the advantage of our solution is evidenced in the last query, "company job", in which the other solutions were unable to predict a single schema correctly. That happened because no schema has the two terms `company` and `job`.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html.

5.3 Performance

In addition to the production-ready ledger called *live*, which currently has 149 schemas, Sovrin also offers two ledgers for development, *builder* and *sandbox*, which currently have, respectively, 875 and 34543 schemas. We evaluated the performance of our solution on the three ledgers running different amounts of queries and measuring how long they took on commodity hardware. Figure 4 shows how long, on average, each test took. In total, 11 tests composed of $[2^0, 2^1, 2^2, ..., 2^{10}]$ unique two-term queries were performed for each ledger. Although the running time of our tool has scaled linearly with the number of schemas on a ledger, it executed 10 queries per second on *live* and 4 on *builder* but took 20 s to run one query on *sandbox*. Therefore, we conclude that it has adequate performance for the short and middle-term future. Many improvements can be made, for instance, to use multi-threading on a CPU level or to use a Graphics Processing Unit (GPU).

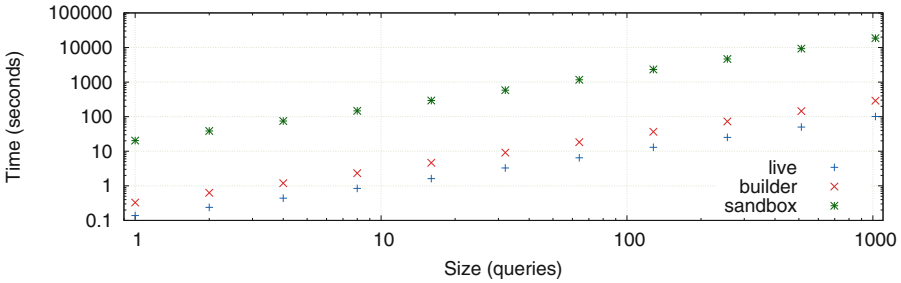

Fig. 4. The performance of our semantically-aware schema matching tool.

6 Conclusion and Future Works

In this study, we investigated the problem of searching metadata in blockchain-based SSI. An SLR was conducted to identify the research materials that have been published considering this issue. Moreover, we proposed a novel tool to find similar schemas in this context. As a study case, our tool performs searches on the Sovrin [18] ledger, a publicly available blockchain-based SSI system. Nonetheless, it can be easily expanded to other SSI systems. We have conducted three experiments to measure and compare the proposed solution with the works found in the literature and a performance analysis regarding its scalability.

The experimental results show that it outperforms the existing works in scenarios where the query approaches natural language, which benefits non-specialists, thus popularizing SSI. Both the research method adopted in our SLR and experiments can be reused in future studies. In this sense, we intend to continue this research and: (i) experiment with a more extensive set of queries; (ii)

evaluate how different pre-trained word-vectors impact the result; and (iii) add pre-trained word vectors of different languages to capture schemas in languages other than English.

References

1. Al Omran, F.N.A., Treude, C.: Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments. In: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), pp. 187–197. IEEE (2017). https://doi.org/10.1109/MSR.2017.42
2. Allen, C.: The path to self-sovereign identity (2016). http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html. Accessed 24 Feb 2021
3. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_7
4. Chadwick, D.W.: Federated identity management. In: Aldini, A., Barthe, G., Gorrieri, R. (eds.) FOSAD 2007-2009. LNCS, vol. 5705, pp. 96–120. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03829-7_3
5. Facebook Database Engineering Team: Rocksdb (2013). https://rocksdb.org/. Accessed 06 Apr 2021
6. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: industrial-strength natural language processing in python (2020). https://doi.org/10.5281/zenodo.1212303
7. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. Commun. ACM **47**(4), 75–78 (2004)
8. Jones, M.B., McIntosh, M.: Identity metasystem interoperability version 1.0. OASIS Standard (2009)
9. Keele, S., et al.: Guidelines for performing systematic literature reviews in software engineering. Technical report (2007)
10. Kuperberg, M.: Blockchain-based identity management: a survey from the enterprise and ecosystem perspective. IEEE Trans. Eng. Manag. **67**, 1–20 (2019). https://doi.org/10.1109/TEM.2019.2926471
11. Lim, S.Y., et al.: Blockchain technology the identity management and authentication service disruptor: a survey. Int. J. Adv. Sci. Eng. Inf. Technol. **8**(4–2), 1735–1745 (2018). https://doi.org/10.18517/ijaseit.8.4-2.6838
12. Lux, Z.A., Beierle, F., Zickau, S., Göndör, S.: Full-text search for verifiable credential metadata on distributed ledgers. In: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pp. 519–528. IEEE (2019). https://doi.org/10.1109/IOTSMS48152.2019.8939249
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
14. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Comput. Sci. Rev. **30**, 80–86 (2018). https://doi.org/10.1016/j.cosrev.2018.10.002
15. Petticrew, M., Roberts, H.: Systematic Reviews in the Social Sciences: A Practical Guide. Wiley, Hoboken (2008)
16. Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., Mortimore, C.: Openid connect core 1.0. The OpenID Foundation, p. S3 (2014)

17. Staš, P.: Hyperledger indy transaction explorer (2019). https://indyscan.io/. Accessed 21 Mar 2021
18. The Sovrin Foundation: Sovrin (2016). https://sovrin.org. Accessed 24 Feb 2021
19. The Sovrin Foundation: Write To The Sovrin Public Ledger (2019). https://sovrin.org/issue-credentials/. Accessed 24 Feb 2021
20. Tobin, A.: WriteSovrin: what goes on the ledger? (2018). https://sovrin.org/wp-content/uploads/2017/04/What-Goes-On-The-Ledger.pdf. Accessed 24 Feb 2021
21. Whitehead, A.: Sovrin main net (2019). https://sovrin-mainnet-browser.vonx.io/. Accessed 21 Apr 2021