Smart Contracts Search Engine in Blockchain

Hien TRAN
UMANIS
France
htran@umanis.com

Tarek MENOUER
UMANIS
France
tmenouer@umanis.com

Patrice DARMON
UMANIS
France
pdarmon@umanis.com

Abdoulaye DOUCOURE
UMANIS
France
adoucoure@umanis.com

François BINDER
UMANIS
France
fbinder@umanis.com

ABSTRACT

Recently blockchain technology has attracted increasing attention. It provides a distributed peer-to-peer network. It also allows to enlarge the contracting space using smart contracts. Smart contract is a numeric protocol which define a promises between parties. In a blockchain platform, we can find millions of smart contracts which are created by different users (developers). The smarts contracts created by users and saved in a blockchain can be similar in functionality, even if different users use different wording. This paper presents a new search engine in the blockchain. The novelty of our search engine is to help users to checks their smart contracts by referencing some similar existing smart contracts created and saved in a blockchain platform. Our search engine allows to give each user an adaptive set of similar smart-contracts based on the user's similarity needs. Our search engine is developed in the Ethereum blockchain platform. Experiments demonstrate the potential of our search engine system under different scenarios.

CCS CONCEPTS

• **Computer systems organization → Blockchain**.

KEYWORDS

Blockchain technology, Smart contracts, Similarity, Engine search, Elasticsearch

ICFNDS '19, July 1–2, 2019, Paris, France

ACM ISBN 978-1-4503-7163-6/19/07…$15.00
https://doi.org/10.1145/3341325.3342015

ACM Reference Format:
Hien TRAN, Tarek MENOUER, Patrice DARMON, Abdoulaye DOUCOURE, and François BINDER. 2019. Smart Contracts Search Engine in Blockchain. In *3rd International Conference on Future Networks and Distributed Systems (ICFNDS '19), July 1–2, 2019, Paris, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3341325.3342015

1 INTRODUCTION

Recently, the blockchain technology has been used in different sectors of activities such as industry and research. The reason for this explosion of interest is that applications can operate in a distributed fashion, without the need or help of some authority. The blockchain allows to store digital data at minimal cost and can enforce agreements while enlarging the contracting space using smart contracts.

Smart contracts represent self-executing scripts that reside on the blockchain and allows to execute promises described by parties.

In a blockchain, we can have millions of smart contracts which are termed by users and saved in the blockchain platform. The different smarts contracts saved in the platform can be similar. However, with the current blockchain techniques, a new user cannot improve his smart contracts by comparing it to other smarts contracts created and saved in the platform. To resolve this problem, we propose in this paper a new smart contracts search engine in blockchain. The novelty of our search engine is that it allows to help a new user to improve its smart contracts by giving him a set of similar smarts contracts saved in the blockchain platform. The number of the returned smarts contracts is adapted according to the need of users and the availability of the smart contracts in the platform.

Our search engine is developed in the Ethereum blockchain platform. To select a set of similar smart contracts, we propose to use a textual search algorithm based on Elasticsearch scoring approach [5]. The goal is to compare between the new user smart contracts and all smart contracts saved in the platform. The used textual search algorithm give a score for each smart contract, then according to the need of the user,

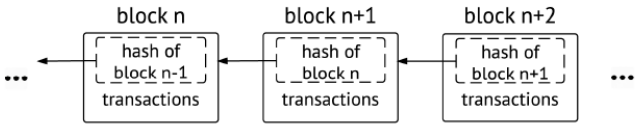

Figure 1: Blockchain [3]

in term of similar smart contracts, our algorithm returns from the platform the most similar smart contracts with the highest scores. This discrete syntactic similarity score is very precise as machine code doesn't carry ambiguities.

The paper is organized as follows. Section 2 gives some preliminaries related to blockchain technology. Section 3 presents some related works. Section 4 shows the principle of our smart contracts search engine. Section 5 shows an example of using of our search engine. Finally, a conclusion and some future works are given in section 6.

2 PRELIMINARIES

In this section we gives some preliminaries related to blockchain technology in subsection 2. Then, we present the smart contracts in subsection 2.

Blockchain technology

Blockchain is a distributed datastore [2, 10] of ordered records, or events, called blocks which are shared among the members of a network. Its novelty resides in the decentralized database management system. As presented in [3], each block is identified by its cryptographic hash and it references the block that came before it. This connection between blocks creates a blockchain, as presented in Figure 1.

Smart contracts

As presented in [11] and [1], a smart contract represents a numeric protocol that allows executing promises between parties. Smart contracts describe four objectives as privily, observability, enforceability and verifiability. Some blockchains implement such functionalities through programs that are executed on the different nodes that form the blockchain platform. Each node participating in the platform has a local Virtual-Machine (VM) which allow to execute contracts.

3 RELATED WORK

In this section, we will start by presenting briefly some blockchain platform (sub-section 3). Then, giving a short overview about some studies related to blockchain (subsection 3). Finally, concluding by a positioning in sub-section 3.

Blockchain Plateform

Currently in the Blockchain technology there is no standard concerning the implementation or practical creation of this technology type. Indeed, the technology behind the blockchain uses: (i) advanced cryptography; (ii) ad hoc network protocols; and (iii) complex performance optimizations. All of this is far too sophisticated to be developed again every time a project needs blockchain. Fortunately, there are many implementations of open-source blockchains platform, in the following we cite an example of two blockchain platforms [6, 7].

Ethereum [6] is an open-source blockchain platform promoted by the Ethereum Foundation and which defines itself as the "first true global computer", and makes it possible to build on its platform decentralized applications. The advantage of Ethereum is that all kinds of applications can be created. It proposes the solidity language which implements the smart contracts.

Hyperledger [7] is also an other open-source implementation which is proposed by Linux Foundation. The goal is to identify and realise an open standard platform.

Overview about blockchain studies

In the following, we present some studies presented in the blockchain context [2, 3, 9, 12].

Christidis et al. [3] present a description of how blockchains and smart contracts work, to identify the advantage and the inconvenience that their approach brings to a system, and highlight the ways the blockchains and Internet of Thing (IoT) can be used together. The authors' study allows finding new use cases for their IoT work.

Unterweger et al. [12] implement a smart contracts from the energy domain initially proposed in [8], which is both, privacy-preserving and of a practical level of complexity. This implementation solves a problem in a way that is representative of the state-of-the-art privacy enhancing technologies.

Benchoufi et al [2] present how the blockchain technology is used to clinical trials and illustrate its general principle in the context of consent to a trial protocol. In this study, authors show also the potential impact of blockchain in the clinical trials context. Finally, authors present how modern clinical trial methods can be used with blockchain technologies to tackle the aforementioned challenges.

Kosba et al. [9] present Hawk, a distributed smart contracts system which does not save the financial transactions in the clear on the blockchain, thus retaining transactional privacy from the public's view. In this paper, authors say that Hawk programmer can write a smart contracts in an intuitive manner without having to implement cryptography.

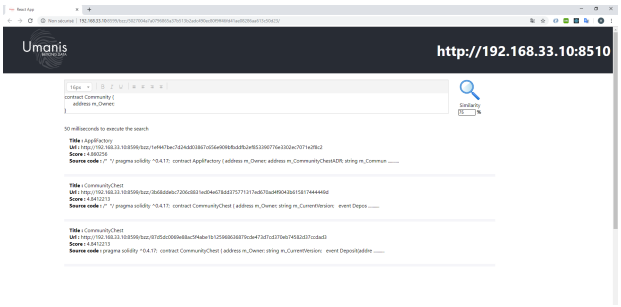

Figure 2: Interface of our search engine system

```
pragma solidity ^0.4.17;

contract CommunityChest {
 address m_Owner;
 string m_CurrentVersion;

 /* function which allow to depose the EThers */

   function deposit(uint256 amount) payable public {
       require(amount > 0,"amount parameter must > 0");
  require(address(m_Owner) != address(0));
  emit Deposit(msg.sender,amount,address(this).balance,1,now);
  emit AppLogMsg(msg.sender,amount,address(this).balance,4);
   }

}
```

Figure 3: Example of smart contract depositing money ethereum

Positioning

The novelty of our work compared to the existing studies consist to propose a new search engine of smart contracts in a blockchain platform. The goal of our search engine is to help a new user to define its new smart contracts by comparing with others similar smart contracts. Our search engine is implemented in Ethereum platform and it is based on Elasticsearch scoring approach [5].

4 SMART CONTRACTS SEARCH ENGINE

The goal of our search engine consists in helping the users to find a set of smart-contracts corresponding to the criteria requested among all smart-contracts stored in the database. In our context, each smart contract is considered as a textual document. However, it is difficult to perform an exact search by producing as a criterion the complete character string of the document. The idea is to interpret the request which represents the new user smart contract as a need, then identify the documents closest to the need. Our proposed search engine system is developed in the Ethereum blockchain platform. Figure 2 shows the interface of our search engine system. To give an example of a smart contract, we present in Figure 3 how we depose the crypto currency Ethereum (ETH) using a smart contract.

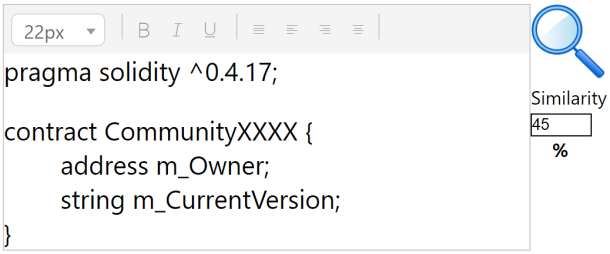

Figure 4: Example of a simple new user smart contract

In the literature, there are several advanced algorithms and methods which are proposed to do a text search. In particular, the use of natural language processing and the inverted indexing system allows us to efficiently retrieve and sort smart-contracts.

In our context, the proposed search engine is based on the Elasticsearch [5] scoring approach [1]. This approach is used to score documents, it allows firstly to reduce the candidate documents to check if the document match with the user query. Then, each document receives a personalized score according to it similarity with the user query.

As presented in [4], Elasticsearch uses Lucene's Practical Scoring Function. The principle is based on Term Frequency (TF) and Inverse Document Frequency (IDF) that also uses the Vector Space Model (VSM) for multi-term queries. The Lucene's Practical Scoring Function represents a relevance score of each document. This approach assigns the high score to the document which has a high similarity with the user query. The TF is a square root of the number of times the term appears in the field of a document as presented in the formula 1. TF assumes that each time a term has a high appearance in a document, its relevancy should be also high. IDF is a natural log of the documents in the index divided by the number of documents that contain the term as presented in formula 2.

$$\boxed{TF = \sqrt{termFreq}} \quad (1)$$

$$\boxed{IDF = 1 + ln(maxDocs/(docFreq + 1))} \quad (2)$$

5 DEMONSTRATION OF A USE CASE

In following, we present an example of how our smart contracts search engine works.

Figure 5 show the result of searching all smart contracts with 45% of similarity with a new simple user smart contract presented in Figure 4.

[1] https://www.compose.com/articles/how-scoring-works-in-elasticsearch/

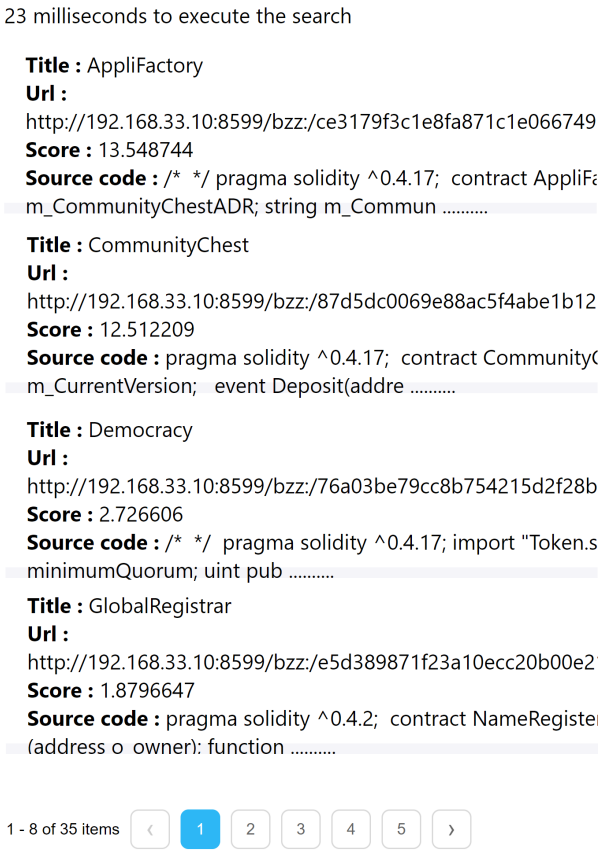

Figure 5: Result returned by our search engine with 45% of similarity with the new simple user smart contract

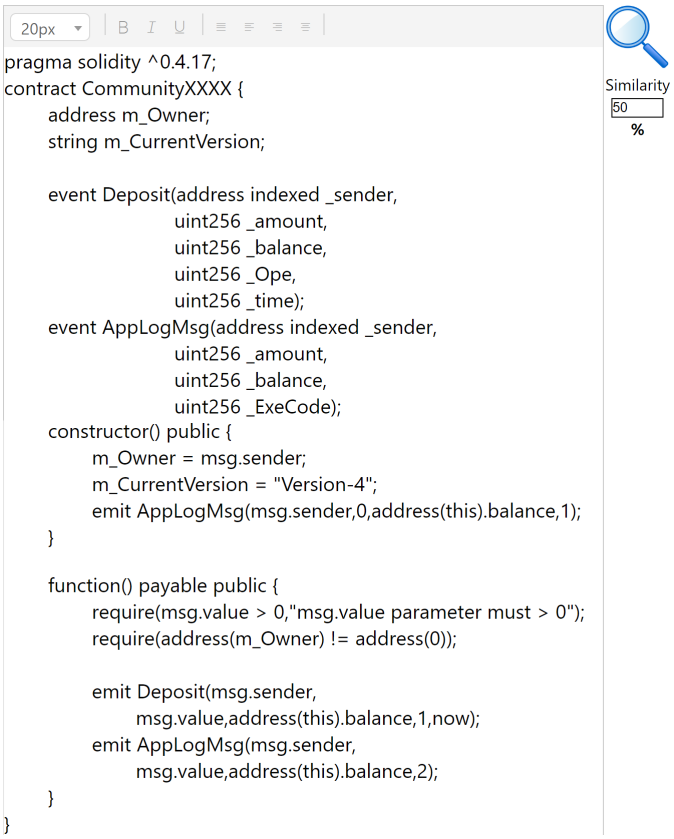

Figure 6: Example of a complex new user smart contract

Figure 7 show the result of searching all smart contracts with 50% of similarity with a new complex user smart contract presented in Figure 6.

Figure 8 show the result of searching all smart contracts with 75% of similarity with a new complex user smart contract presented 6.

Search for smart contracts	Computing time (ms)
45% of similarity with the new simple user smart contract	19
50% of similarity with the new complex user smart contract	52
75% of similarity with the new complex user smart contract	28

Table 1: Computing time required to search similar smart contracts

Table 1 show the computing time required to search similar smart contracts for a new simple and complex user smart contracts with different percentage of similarity. We note from Table 1 that our search engine return similar smart contracts quickly in less than one second. However, the computing time varied according to the type of the new user smart contract (simple/complex) and according to the requested percentage of similarity.

6 CONCLUSION

In this paper we present a new search engine of smart contracts. The benefit of our search engine is that it allows to help new user to improve the creation of a new smart contract by referring to the different smart contracts created previously in a blockchain platform. Our search engine is implemented in Ethereum blockchain platform. To search the set of similar smart contracts, our engine is based on the Elasticsearch scoring approach. As a perspective, we propose to improve the search approach by using a new algorithm in order to score the set of smart contract saved in the blockchain platform.

Our approach doesn't track the semantic dimensions of functional programming which can carry similarity characteristics as well, for example when different programmers

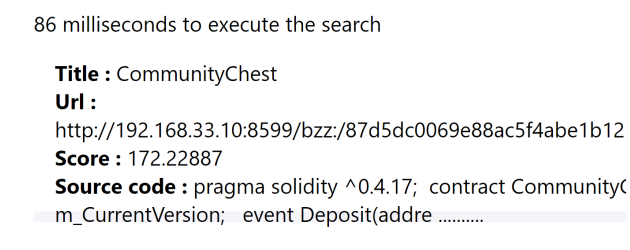

Figure 7: Result returned by our search engine with 50% of similarity with the new complex user smart contract

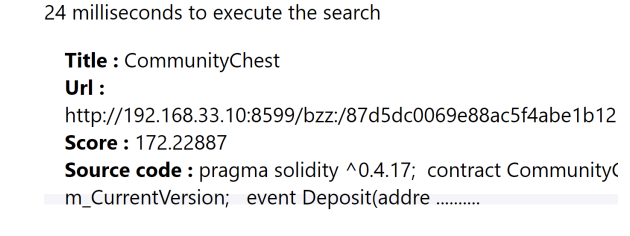

Figure 8: Result returned by our search engine with 75% of similarity with the new complex user smart contract

use different names for the same function in two different contracts. We propose as a second perspective to use our search algorithm, to tag a machine learning model. From there, we can train a model to infer the semantic dimension in order to complete the similarity score.

As another perspective, we want to use our search engine in an industrial context to help users in creating new smart contracts.

REFERENCES

[1] T. Abdellatif and K. Brousmiche. Formal verification of smart contracts based on users and blockchain behaviors models. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Feb 2018.

[2] M. Benchoufi and P. Ravaud. Blockchain technology for improving clinical research quality. *Trials*, 18(1):335, Jul 2017.

[3] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *Ieee Access*, 4:2292–2303, 2016.

[4] Elasticsearch: https://www.compose.com/articles/how-scoring-works-in-elasticsearch/, visited 01-04-2019.

[5] Elasticsearch: https://www.elastic.co., visited 01-04-2019.

[6] Blockchain ethereum: https://www.ethereum.org/, visited 01-04-2019.

[7] Blockchain ethereum: https://www.hyperledger.org/, visited 01-04-2019.

[8] F. Knirsch, A. Unterweger, G. Eibl, and D. Engel. *Privacy-Preserving Smart Grid Tariff Decisions with Blockchain-Based Smart Contracts*, pages 85–116. Springer International Publishing, Cham, 2018.

[9] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 839–858, May 2016.

[10] G. W. Peters and E. Panayi. *Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money*, pages 239–278. Springer International Publishing, Cham, 2016.

[11] N. Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.

[12] A. Unterweger, F. Knirsch, C. Leixnering, and D. Engel. Lessons learned from implementing a privacy-preserving smart contract in ethereum. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Feb 2018.