PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Visualizando o Ecossistema Blockchain Utilizando Machine Learning e Grafos de Conhecimento

Rodrigo Londres Agapito da Veiga

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, novembro de 2021

Rodrigo Londres Agapito da Veiga

Visualizando o Ecossistema Blockchain Utilizando Machine Learning e Grafos de Conhecimento

Relatório de Projeto Final, apresentado ao curso de
Ciência da Computação da PUC-Rio como
requisito parcial para a obtenção do título de
Bacharel em Ciência da Computação.

Orientador: Markus Endler

Coorientadora: Valeria de Paiva

Rio de Janeiro, novembro de 2020

"I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third-party." - Satoshi Nakamoto

Agradecimentos

A meus pais, por batalharem e investirem na minha educação desde pequeno. Aos meus amigos, de dentro e fora da faculdade, pelo companheirismo nas horas mais necessárias. À minha namorada, pelo apoio emocional e incentivo nos momentos de dúvida. Aos meus orientadores, Markus Endler e Valeria de Paiva, pois este trabalho não seria possível sem sua ajuda.

Resumo

Veiga, Rodrigo. Endler, Markus. De Paiva, Valeria. Visualizando o Ecossistema Blockchain Utilizando Modelos de Machine Learning e Grafos de Conhecimento. Rio de Janeiro, 2021. 30p. Relatório Final de Graduação – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Conforme um domínio acadêmico vai amadurecendo e sendo sujeito a cada vez mais pesquisa e desenvolvimento, a análise da informação torna-se mais complexa e trabalhosa devido ao volume acelerado de artigos publicados e tecnologias desenvolvidas, tornando as fontes de informação cada vez mais dispersas. Com o avanço da internet, a realização da pesquisa se tornou mais acessível, mas ao mesmo tempo resultou em mais informação para ser avaliada. Combinando as inúmeras fontes de novas informações com o avassalador ritmo com que tecnologias *open-source* são desenvolvidas, vemos uma dificuldade ainda maior em acompanhar tudo o que acontece em um domínio específico. A utilização de ferramentas de processamento de linguagem natural (NLP) possibilitam a construção de grafos de conhecimento de um domínio de conhecimento, contendo suas entidades e as ligando por meio de relações. Desta maneira, as informações ficam organizadas e interligadas, de maneira em que a pesquisa e o desenvolvimento científico são facilitados. Este trabalho busca utilizar ferramentas existentes que realizam as tarefas de extração de entidades e relações em um corpus de textos relacionados à área de blockchain, uma tecnologia relativamente nova e inovadora, onde novos projetos e tecnologias são introduzidas constantemente. Foi realizada uma avaliação dos resultados obtidos, a fim de identificar se são suficientemente bons para a construção de um grafo de conhecimento de blockchain. Também foi construída uma interface web para visualização do grafo de conhecimento. As ferramentas desenvolvidas neste trabalho serão disponibilizadas em repositório público, e o autor incentiva a utilização delas junto da aplicação dos mesmos modelos de predição utilizados para construir grafos de conhecimento de outras áreas relacionadas à computação, não apenas blockchain.

Palavras-chave

Blockchain; criptomoedas; grafos de conhecimento; machine learning; desenvolvimento web; processamento de linguagem natural (NLP); AllenNLP; DyGIE++

Abstract

Veiga, Rodrigo. Endler, Markus. De Paiva, Valeria. Visualizing the Blockchain Ecosystem Using Machine Learning Models and Knowledge Graphs. Rio de Janeiro, 2021. 30p. Relatório Final de Graduação – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

As an academic domain matures, the analysis of information becomes increasingly more complex and arduous due to the accelerated volume of published articles and new technologies that are developed, making the sources of information available more scattered. With the advance of the internet, the realisation of research has become more accessible, while

at the same time resulting in more information to be analysed. Combining the numerous sources of new information with the staggering rhythm with which open-source technologies are developed, there is an increased difficulty in keeping up with all of the new advancements in a specific domain. The utilisation of natural language processing (NLP) tools enables the construction of knowledge graphs for a domain of information, containing its entities and connecting them through their relations. This way the information becomes available in a more organised and interconnected manner, making it easier for research and development to be accomplished. This project aims to utilize existing tools to extract entities and relations from a corpus of text files related to the blockchain field, a relatively new and innovative technology, where new projects and concepts are constantly being introduced. An analysis of the results obtained was undertaken, to identify if they were well suited for the construction of a blockchain knowledge graph. A website was developed in order to visualize the knowledge graph. The tools that were developed for this project will be made available in a public repository, and the author encourages their use along with the prediction models used during this project in order to build knowledge graphs for any domain related to computer science, and not only blockchain.

Keywords

Blockchain; cryptocurrencies; knowledge graphs; machine learning; web development; natural language processing (NLP); AllenNLP; DyGIE++

Sumário

1. Introdução

1.1 Blockchain, Bitcoin e Industry 4.0

O conceito de blockchain foi introduzido pela primeira vez em um artigo científico de Haber, S. *et al*, com o objetivo de implementar um sistema onde informações sobre a criação de um documento fossem imutáveis [1]. Mas foram necessárias quase duas décadas, com a invenção do Bitcoin, para que este conceito fizesse parte de uma aplicação de verdade [2].

A blockchain é uma abordagem de Distributed Ledger Technology (DLT), que é um registro de transações baseado em cadeias de blocos que contém informações sobre diversas transações. Estes blocos mantêm sua integridade através de uma rede colaborativa de computadores que trabalham para ler e validá-los. Este trabalho computacional realizado conta com métodos matemáticos para manter a sua rede segura e confiável em relação a seus dados. Qualquer computador pode conectar-se à rede como um novo nó participante e executar tarefas para validar blocos de transações. Ao ser validado por um dos nós, um bloco é então adicionado à cadeia de blocos e os nós restantes da rede passam a ver esta nova cadeia estendida como a fonte de verdade. Desta maneira, os nós participantes da rede se tornam uma peça central no funcionamento de uma blockchain [3].

Esta breve descrição da blockchain é tomada do artigo que introduz sua primeira aplicação bem sucedida em escala, o *whitepaper* original do Bitcoin. Este arquivo foi divulgado em 4 de Janeiro de 2009 através de listas de e-mail para entusiastas de criptografia por uma entidade desconhecida utilizando o pseudônimo Satoshi Nakamoto, e com seu lançamento o Bitcoin se tornou a primeira tentativa bem-sucedida de se criar um sistema financeiro com moeda eletrônica totalmente digital e descentralizado. O contexto da época ajuda a entender parte da motivação por trás desta invenção, já que o mundo ainda se recuperava de uma das maiores crises financeiras da história em 2008. Um grande fator que contribuiu para este colapso leva em conta a irresponsabilidade de alguns bancos e grandes instituições financeiras, que durante décadas se consolidaram como pilares da economia moderna atuando como fonte de confiança no mercado financeiro e agora se encontravam na posição oposta, como sendo prejudiciais à vida financeira de seus clientes e motivo de grande desconfiança [4]. Parecia mesmo ser a hora de trazer

dinheiro para a internet, e assim se fez. Após 12 anos, a rede Bitcoin continua a cumprir o seu principal objetivo: utilizar de um mecanismo de consenso para garantir a veracidade e integridade das transações realizadas em sua rede. A maior ameaça para as blockchains são os ataques de 51%, que ocorre quando um ator mau intencionado controla mais de 50% de todos os computadores ligados como nós na rede [5]. Estes tipos de ataques não são financeiramente viáveis ou desejáveis, e até hoje nenhum ataque deste tipo foi realizado.

Embora muitos vejam esta tecnologia com um grau de desconfiança, como é o caso com qualquer tecnologia inovadora, blockchain e Bitcoin claramente ganharam muito em popularidade e adoção nos últimos anos, especialmente entre as gerações abaixo dos 40 anos, e agora também os nativos da internet [6]. Isso não deveria nos surpreender, devido ao fato de que este público é mais dependente e acostumado com computadores e a internet. Para muitas destas pessoas, uma forma de dinheiro nativo da internet parece completamente natural, e alguns inclusive se surpreendem que nenhuma foi completamente adotada ainda.

Serviços financeiros estão cada vez mais acessíveis a qualquer um com um dispositivo móvel, com quase 50% da população global já utilizando seus telefones celulares para ao menos realizar consultas de saldo e extrato [7]. Com cada vez mais dispositivos com acesso à internet e conectados uns aos outros, vemos no setor financeiro grandes promessas de inovação, e uma delas é a blockchain. Com cada vez mais dispositivos com acesso à internet e conectados uns aos outros, novas inovações se tornariam possíveis se eles possuíssem a capacidade de realizar transações financeiras entre si. Estas transações não se limitam apenas a telefones celulares, mas a qualquer dispositivo eletrônico que possa atuar como participante de uma economia. Um exemplo clássico de uma aplicação blockchain em dispositivos eletrônicos é o de uma máquina de vendas programada com um contrato inteligente para receber pagamentos e liberar os produtos assim que forem pagos. Estes contratos inteligentes, ou *smart contracts*, são uma das principais inovações na área de blockchain, e são definidos como uma espécie de conta bancária não controlada por um usuário, sendo um programa implementado em uma rede blockchain que funciona de acordo com as regras estabelecidas no código, ou no contrato. Estes contratos possuem seu próprio saldo e conseguem receber e realizar transações, possibilitando a interação do usuário com eles. [8]

Tendo em vista este cenário de cada vez mais dispositivos conectados e interagindo tanto com humanos quanto com outras máquinas, a invenção da blockchain pode ser o evento catalisador para que o setor financeiro se adapte por completo ao meio digital, tornando-se assim mais uma tecnologia-chave na engrenagem da “Industry 4.0”, nome dado à nova revolução industrial que envolve meios de produção e distribuição inteligentes e interligados, altamente dependentes em informação obtida em tempo real. [9]

1.2 Crescimento exponencial com open-source

O código fonte do Bitcoin foi imediatamente disponibilizado para qualquer um que desejasse analisar ou melhorá-lo. Sendo um software de código aberto, nos anos seguintes a comunidade Bitcoin e blockchain tratou de melhorar a tecnologia de maneira semelhante à como a comunidade Linux transformou o sistema operacional em um dos mais conhecidos e bem sucedidos do mundo [10]. A informação e o desenvolvimento são compartilhados por milhares de mentes criativas, e isso levou à criação de diversos projetos e tecnologias que podem trabalhar juntas a fim de alcançar um objetivo final inovador, criando novas soluções para problemas anteriores.

Em relação ao Bitcoin, ainda existem problemas técnicos importantes a serem resolvidos, com o maior deles provavelmente sendo o das taxas caras e velocidade de transferência lenta quando a rede se encontra congestionada, o que costuma estar. Por este motivo, é fácil perceber que o Bitcoin não serve como unidade de dinheiro eletrônico a ponto de substituir o Real e o Dólar para uso diário. Este tipo de problema técnico é complexo e difícil de resolver, mas provavelmente realizável. A probabilidade deste problema encontrar uma solução se torna ainda maior devido ao fato de que existem milhares de pessoas buscando soluções diariamente - afinal, tudo é aberto para qualquer um participar. O fato que qualquer um pode criar uma cópia do software do Bitcoin permitiu a criação de grandes inovações na área de blockchain, com o exemplo mais notável sendo o do Ethereum, uma criptomoeda proposta em 2013 e lançada em 2015 pelo então programador Russo-Canadense de 21 anos Vitalik Buterin [11]. Ao contrário do Bitcoin, o Ethereum possui uma blockchain Turing-completa, dando a esta rede mais flexibilidade e permitindo a criação de *smart contracts*, que podem ser programados

de maneira semelhante à linguagens de programação modernas. A introdução de *smart contract* foi um verdadeiro marco na área de blockchain - com ela, transferir dinheiro deixou de ser a única aplicação possível. Um conceito que tem ganhado força recentemente é o de finanças descentralizadas (DeFi), que faz uso de *smart contracts* para replicar serviços financeiros tradicionais (como empréstimos, seguros e derivativos) de maneira aberta, transparente e interoperável [12].

Atualmente, existem mais de nove mil diferentes criptomoedas [13], e embora a vasta maioria delas sejam inúteis, muitas possuem equipes e comunidades incrivelmente ativas focadas no desenvolvimento de projetos inovadores. Na verdade, novas tecnologias são introduzidas de maneira tão rápida que pode ser um tanto quanto trabalhoso, ou até mesmo impossível, de se manter atualizado com tudo o que está acontecendo. Para realmente entender o valor de muitos destes projetos, é necessário possuir não apenas capacidades técnicas como também a força de vontade para pesquisar a fundo. Mesmo atendendo a estes pré-requisitos, é comum precisar voltar a um tópico anterior para entender um novo. Se existe qualquer ponto negativo sobre desenvolvimento colaborativo e *open-source* nesta escala, deve ser esta constante necessidade de continuar aprendendo para não ficar para trás. Na maioria dos casos, é difícil saber inclusive por onde começar. Embora muitos sites façam um bom trabalho listando os principais projetos, eles não nos dão uma boa visão sobre o que estes projetos fazem e como se relacionam uns com os outros. Esta ideia de relacionar projetos pode não fazer muito sentido no mundo empresarial tradicional, mas é muito importante na área de blockchain e pode ser crucial para entender este campo, já que estes projetos estão sempre em constante colaboração.

1.3 Motivação e objetivos do trabalho

Com a grande aceleração com que este espaço é desenvolvido, cresce também a demanda para quem busca aprender sobre blockchain. Sendo um assunto de certa forma bastante técnico, pode chegar a parecer um tanto quanto intimidador para começar a aprender, e muitas pessoas nem sequer sabem por onde começar. Por este motivo, o principal objetivo deste projeto foi aplicar modelos de machine learning para realizar a extração de informação de *whitepapers* (como o do Bitcoin) e artigos acadêmicos voltados para a área de blockchain. Estes modelos de predição foram treinados com o dataset SciERC, criado por Yi Luan *et al* [14],

que foi construído a partir de informações tiradas de artigos científicos de computação. Buscamos obter as entidades e relações tiradas de arquivos de blockchain, a fim de gerar um grafo de conhecimento que pode ser utilizado para pesquisa e aprendizado. A realização deste trabalho pode ser vista em quatro partes:

1. **Coleta de dados de qualidade relacionados a projetos e tecnologias relevantes no ecossistema blockchain, separados em dois corpus distintos.**
    1.1. Não foi encontrado na internet um dataset relacionado a blockchain que atendesse aos propósitos deste trabalho, apenas sites com links para *whitepapers* e artigos acadêmicos. Foi necessário extrair dados destes documentos e agrupá-los da seguinte maneira:
        1.1.1. Corpus formado por frases de *whitepapers* publicados por projetos relevantes na área de blockchain. *Whitepapers* se assemelham a artigos científicos, pois são bastante técnicos e descrevem bem uma tecnologia para tratar de um problema, embora não sejam avaliados e julgados por outros pesquisadores como acontece com os artigos publicados em conferências científicas e revistas científicas.
        1.1.2. Corpus formado por frases de artigos acadêmicos relacionados a blockchain.
    1.2. A criação do corpus foi um de dois produtos gerados neste trabalho, e será exposto ao público que deseja utilizá-lo em pesquisas futuras.
2. **Execução do framework DyGIE++ para treinar o modelo de predição e aplicá-lo nos dados coletados**
    2.1. Utilização do dataset SciERC para treinar o modelo de predição. O SciERC é uma coleção de 500 abstratos científicos, de linguagem semelhante à dos dados coletados.
    2.2. Análise dos resultados

2.2.1. Comparação das entidades e relações obtidas pelo modelo com o resultado das entidades e relações obtidas manualmente

3. **Desenvolvimento do dygiepp-reader para formatar resultados e alimentar a interface web**
   3.1. Criação do programa em Javascript dygiepp-reader [14] para disponibilizar os resultados para uma interface web
4. **Desenvolvimento de uma interface web para visualização dos resultados**
   4.1. Criação de uma interface web para visualização e interação com os resultados
   4.2. Corresponde ao segundo produto gerado neste projeto, será disponibilizado na internet em domínio público para quem desejar utilizar da ferramenta para fins de pesquisa ou aprendizado.

Como projetos e tecnologias em blockchain estão em constante colaboração e aperfeiçoamento, é muito importante entender como estes se relacionam para entender exatamente o que está acontecendo nesta área. Como atualmente não existem ferramentas que permitam este tipo de visualização, é de se imaginar que existam oportunidades tecnológicas importantes que ainda estamos para descobrir com uma visão aumentada sobre o ecossistema blockchain. Utilizaremos o conjunto de dados SciERC para treinar um modelo de predição, e assim seremos capazes de gerar um grafo semântico com entidades (conceitos, problemas, métodos, etc.), e as relações que existem entre elas (usado-para, funcionalidade-de, etc.) [15]. Este modelo de predição será treinado, avaliado e executado através do DyGIE++, um framework em Python desenvolvido por coautores do paper que introduz o SciERC.

Tendo em mente que não foram encontrados recursos online que servissem como dataset para geração de um grafo de conhecimento e a crescente demanda por informações relacionadas à tecnologias e projetos em blockchain, este projeto busca atacar estes problemas com o desenvolvimento de dois produtos finais. O primeiro é um corpus de arquivos textos sobre tecnologias, conceitos e projetos relacionados a blockchain que seja suficientemente grande para a construção de um grafo de conhecimento. O segundo é uma interface web disponibilizada em domínio

público que permita a visualização e interação com este grafo, ligando conceitos, ou entidades, uns aos outros por meio de relações. Esperamos não apenas facilitar a vida de quem busca aprender sobre blockchain e criptomoedas, mas também auxiliar futuras pesquisas com os dados coletados, que ficarão públicos e serão postados em fóruns de blockchain para feedback e uso livre.

2. Ferramentas utilizadas

2.1 AllenNLP para construção de grafos de conhecimento

Com o avanço da pesquisa científica em qualquer área acadêmica, a análise da informação se torna mais complexa devido ao alto volume de artigos publicados, tornando as fontes de informação cada vez mais dispersas [16]. Felizmente, o rápido desenvolvimento de ferramentas de processamento de linguagem natural (NLP) permitiu que fossem construídos grafos de conhecimento de um domínio específico, contendo suas entidades e as ligando por meio de relações [17]. Desta maneira, as informações ficam organizadas e interligadas, de maneira que facilita a pesquisa e o desenvolvimento científico.

Uma destas ferramentas é o AllenNLP, uma biblioteca que aplica modelos de *deep learning* a tarefas de NLP, fornecendo comandos via linha de comando que são acessíveis para usuários que não estão muito familiarizados com aplicações de NLP, que geralmente exigiriam o desenvolvimento de código especializado. Este conhecimento avançado está acima do escopo da graduação, e como este projeto envolve a criação de grafos de conhecimento relacionados a blockchain, não seria possível realizá-lo se não fosse por esta grande ferramenta que diminui as barreiras de entrada para pesquisas de NLP, tanto para novatos nesta área quanto para pesquisadores mais conceituados mas que vivem em ambientes de maior carência tecnológica [18].

O AllenNLP contribui para o aceleramento na quantidade de pesquisas de pesquisas e experimentos de NLP, e ainda tem a vantagem de ser um projeto *open-source*, tendo seu repositório mantido por engenheiros e pesquisadores conceituados. O seu repositório possui no momento mais de 10 mil estrelas e 2 mil forks, com sua comunidade crescendo em ritmo acelerado e se tornando uma das principais ferramentas de NLP disponíveis. [18]

2.2 Anotações sobre entidades e relações

É possível fazer sentido de textos e artigos acadêmicos com o uso de modelos de predição em IA, que detectam entidades e relações em um conjunto de sentenças. Estes modelos são treinados com conjuntos de dados anotados com entidades e relações, idealmente feitas por profissionais e pessoas conceituadas na área sobre a qual um texto trata, e a qualidade das anotações são de importância vital para o sucesso na predição dos dados. Neste projeto utilizamos as anotações do conjunto de dados SciERC, uma coleção de 500 abstratos cientificos [14, 23]. Estas anotações possuem as entidades e relações dos textos coletados, e utilizaremos o framework DyGIE++ para treinar um modelo que faz uso do dataset SciERC para prever entidades e relações em nossa coleção de textos sobre blockchain. Com estes resultados, podemos criar um grafo de conhecimento sobre blockchain e utilizá-lo como ferramenta para pesquisar e analisar esta rede de conhecimento.

Este trabalho foi inspirado no artigo de Yi Luan *et al* [14]. Este trabalho possui dois principais produtos: um framework chamado SciIE (cuja sigla representa extração de informação de arquivos científicos) para identificação de entidades, relações e co-referências em artigos científicos; um dataset chamado SciERC composto por artigos científicos com anotações sobre entidades, relacionamentos e co-referencias. O modelo de predição do SciIE utiliza dados do SciERC que foram anotados por especialistas, contendo suas verdadeiras entidades e relações. Estas anotações são importantes para treinar o modelo a fim de prever as entidades e relações com certa acurácia. O SciIE tem como resultado final um grafo de conhecimento contendo relações entre as entidades detectadas no corpus de dados. Os grafos são primeiro construídos a nível de documento, isto é, tendo seu escopo limitado às frases de apenas um documento do corpus de artigos científicos. Com estes sub-grafos construídos, as entidades e relações são juntadas em um único grafo. Neste grafo, os nós correspondem à entidades que podem ser qualquer uma das seguintes:

- Task (tarefa)
    - Aplicações, problemas a serem resolvidos
- Method (métodos)
    - Métodos, modelos, ferramentas utilizadas, etc.
- Evaluation Metric (métrica de avaliação)

    - Métricas, medidas que podem ser utilizadas para avaliar um método ou sistema
- Material
    - Dados, recursos
- Other Scientific Terms (outros termos científicos)
    - Termos científicos que não se enquadram em nenhuma outra categoria
- Generic (termos genéricos)
    - General terms or pronouns that may refer to an entity but are not themselves informative, often used as connection words.

As arestas representam as relações entre os nós ou entidades, e são as arestas uma das seguintes:

- Used-for (usado-para)
- Feature-of (pertence-a)
- Hyponym-of (hipônimo-de)
- Part-of (parte-de)
- Compare (comparação)
    - Entidades sendo comparadas
- Conjunction (conjunção)
    - Entidades sendo usadas juntas para reforçar o mesmo ponto

[15]

Grafos de conhecimento são redes semânticas, representando ideias e conceitos e ilustrando a relação entre elas [19]. No caso do SciIE, ele foi utilizado para gerar grafos de conhecimento relacionados à área de inteligência artificial, mas pode ser aplicado a qualquer área de conhecimento. Com o grafo de conhecimento, ideias abstratas podem ser literalmente visualizadas ligando os conceitos que juntos trabalham para formular a ideia. Em uma área nova e de tão rápido desenvolvimento como blockchain, a construção de um grafo semântico ligando seus principais conceitos e tecnologias pode ser útil para fins educativos e de pesquisa.

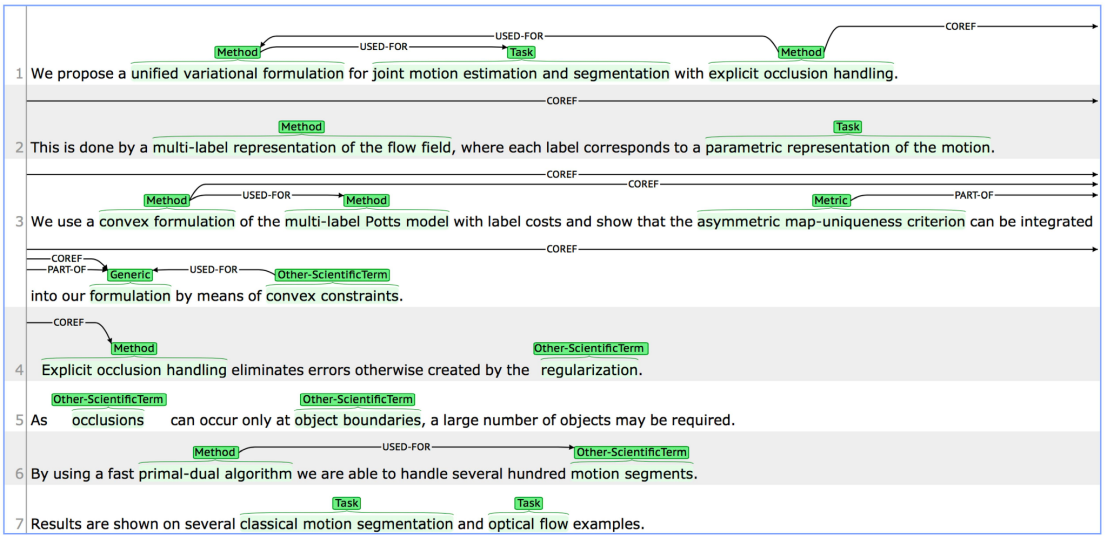


Figura 1: exemplo das anotações utilizadas pelo dataset SciERC [14]

2.2 SciERC

O SciERC é uma coleção de mais de 500 resumos científicos anotados com suas entidades, relações e co-referências. Estes dados foram anotados por especialistas na área que é abordada pelo artigo do abstrato em questão. O SciERC é uma extensão dos datasets SemEval 2017 e SemEval 2018, e inclui mais tipos de entidades e relacionamentos e ligações entre frases utilizando links de co-referência [14]. O ponto chave do SciERC para os interesses deste projeto é que seus artigos são voltados para a área de computação, mais especificamente a de inteligência artificial. Como não existe um dataset anotado com artigos sobre blockchain, foi necessário buscar dados anotados que se assemelham ao tipo de linguagem encontrada em *whitepapers* e artigos acadêmicos sobre blockchain, e o SciERC foi o que melhor se encaixa neste quesito.

2.3 DyGIE++

Utilizamos neste trabalho o DyGIE++, um framework em Python desenvolvido por Wadden, D. *et al [22]* que faz uso do AllenNLP e sua interface via linha de comando para treinar, avaliar e aplicar modelos de predição para extração de entidades e relações em um conjunto de dados. Este programa permite que o modelo seja treinado com 4 diferentes datasets, incluindo o SciERC, nosso dataset de preferência. O DyGIE++ disponibiliza scripts para treinar e avaliar um modelo, e

também realizar predições sobre um conjunto de dados. Embora nosso projeto tenha sido inspirado no artigo de Yi Luan *et al* [14], o framework DyGIE++ provou ser uma porta de entrada rápida para os propósitos deste trabalho, em linha com a acessibilidade que o AllenNLP promete, o que foi visto na seção 2.1.

O DyGIE++ possui os modelos scierc e scierc_lightweight - ambos utilizam o dataset SciERC, porém apenas o primeiro faz uso de coreferências. Quando o autor se referir ao scierc e scierc_lightweight (em letras minúsculas), trata-se dos modelos de predição utilizados. Os modelos scierc e scierc_lightweight foram avaliados, com o primeiro tendo 0,64% de precisão para extração de entidades e o segundo com 0,66% de precisão. Já para a extração de relações, o scierc apresentou 0,44% de precisão contra os 0,55% do scierc_lightweight. Embora o modelo scierc_lightweight inicialmente tenha apresentado uma avaliação melhor, ambos os modelos serão analisados mais adiante. O modelo scierc difere do scierc_lightweight pois faz uso de coreferências, isto é, consegue associar palavras em frases diferentes, enquanto o scierc_lightweight fica limitado ao escopo da frase.

Para realizar predições em um dataset novo, foi necessário formatá-lo da maneira que o DyGIE++ espera. Felizmente, o DyGIE++ fornece um script format_new_dataset para formatar os dados. Isto provou não ser o suficiente, e os resultados se tornaram adequados apenas quando removemos as quebras de linhas dos dados por meio de um script criado para este trabalho. Com os dados formatados podemos então realizar a predição com o modelo treinado, e assim gerar os arquivos de resultado que representam os grafos de conhecimento, separados a nível de documento.

2.3 Node.js

Para desenvolver uma interface web para visualizar o grafo de conhecimento final, foi necessário desenvolver um programa que pudesse manipular os dados dos arquivos de resultado do DyGIE++ e fornecê-los de maneira mais adequada a uma página web. O DyGIE++ armazena os resultados em um arquivo jsonl, contendo uma lista de objetos no formato JSON. Escolhemos o Node.js para realizar tal tarefa devido ao fato de Javascript ser uma linguagem especificamente desenvolvida para a web, e o Node.js possui uma grande variedade de bibliotecas que permitem a construção de uma APIs. Foi escolhida a biblioteca Express para o rápido

desenvolvimento de uma API que disponibiliza com uma rota para consultar os dados obtidos e exibir a rede de conhecimento. O resultado foi um programa que criei chamei de dygiepp-reader, e ele será abordado em mais detalhe em seções mais adiante.

2.4 React

Para desenvolver a interface web foi utilizado o React, um programa com o qual o autor deste trabalho já possui experiência e logo foi visto como a melhor solução para implementação de um site. Para realizar chamadas para a API do dygiepp-reader, foi utilizada a biblioteca Axios, com a qual também possuo familiaridade.

3. Desenvolvimento do trabalho

3.1 Revisão do plano de ação e cronograma

Foi desenvolvido um plano de ação e um cronograma durante a disciplina de Projeto Final I, que envolveu a preparação e estudo para o projeto que seria desenvolvido. O plano de ação inicial, atualizado pela última vez em junho de 2021 e sem nenhuma alteração, foi o seguinte:

1) Estudo e discussões sobre o projeto
2) Coletar textos de abstracts de centenas de *whitepapers* e artigos acadêmicos relacionados a blockchain - já realizado
3) Treinar um modelo de predição com o DyGIE++ utilizado o conjunto de dados do SciERC - já realizado; a Proposta de Projeto Final consta que o SciIE seria utilizado, mas optei pelo DyGIE++ por possuir melhor documentação e facilidade para configurar.
4) Melhorar o modelo utilizando anotações sobre os dados de blockchain coletados na primeira etapa:
    a) Definir os critérios de anotação de dados
        i) Definir os tipos de entidades adequadas
        ii) Definir os tipos de relações adequadas
    b) Anotar os dados de teste

5) Treinar novo modelo com o DyGIE++ com os dados de blockchain anotados
6) Gerar e analisar um grafo com base no resultado do modelo de predição
7) Publicar os resultados em fóruns de blockchain
8) Propor e desenvolver aplicações sobre este grafo que auxiliem no aprendizado sobre blockchain
9) Criação de um documento detalhado de todo o projeto

Tendo em vista este plano de ação, podemos fazer algumas observações. Em relação ao ítem número 2, até o presente momento foi construído um corpus com

1. 105 arquivos de texto retirados de *whitepapers* de projetos de blockchain
2. 57 arquivos de texto retirados de artigos acadêmicos relacionados à blockchain
3. 32 arquivos de texto retirados de páginas da Wikipedia relacionadas à blockchain

para um total de 194 arquivos. Sobre estes números, vale destacar que esperava-se que houvessem mais artigos acadêmicos que detalhassem projetos de blockchain, o principal foco deste projeto, porém o que foi observado foi que a maioria dos artigos buscava analisar tecnologias e conceitos existentes, sem focar em um projeto específico. Como era esperado, os *whitepapers* provaram ser fontes de informação mais adequadas para os propósitos deste projeto.

Como foi observado no ítem de número 3, optei por utilizar o DyGIE++ por possuir uma barreira de entrada mais baixa, por motivos que foram abordados na seção 2. Os ítens 4 e 5 provaram não foram realizados - o modelo de predição treinado utilizando somente os dados do SciERC produziu resultados suficientemente bons para a realização deste projeto. Seria interessante, porém, anotar mais dados relacionados a blockchain para melhorar o modelo de predição . Foram anotados alguns dados do corpus de blockchain para realizar a análise a qual o ítem número 6 se refere. O ítem 7 será realizado após o fim do projeto, e logo ainda não foi realizado. Do ítem número 8 saiu a ideia de desenvolver uma interface web para visualização do grafo resultado - a interface web ainda está em desenvolvimento. Em relação ao ítem número 9, durante todo o desenvolvimento

deste trabalho foram feitas anotações em um documento que eventualmente evoluiu para a redação deste relatório de Projeto Final II.

O cronograma desenvolvido para a disciplina de Projeto Final I, sem nenhuma alteração, foi o seguinte:

Etapas	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Etapa 1									
Etapa 2									
Etapa 3									
Etapa 4									
Etapa 5									
Etapa 6									
Etapa 7									
Etapa 8									

Tabela 1: cronograma inicial para a realização do projeto

Podemos atualizar este cronograma para ilustrar o que de fato foi desenvolvido e durante quais meses do ano. O resultado é o cronograma seguinte:

	Planejado e realizado
	Não planejado e realizado
	Planejado e não realizado

Etapas	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Etapa 1									

Etapa 2									
Etapa 3									
Etapa 4									
Etapa 5									
Etapa 6									
Etapa 7									
Etapa 8									
Etapa 9									
Etapa 10*									

Tabela 2: cronograma final do projeto

*Etapa 10: desenvolvimento do programa dygiepp-reader para formatação dos arquivos de resultado do DyGIE++

3.2 Fontes de dados para construção do dataset de blockchain

Para a realização deste trabalho, foi necessário coletar uma quantidade significativa de dados sobre blockchain. Para os objetivos deste projeto, como desejamos no final criar uma interface web que auxilie quem deseja aprender e pesquisar sobre conceitos e projetos de blockchain, procuramos textos que fossem bastante explicativos contendo termos e conceitos em comum. Como o dataset SciERC, utilizado no treinamento do modelo de predição, foi criado a partir de artigos científicos, foi necessário coletar dados sobre blockchain que possuíssem linguagem semelhante. Felizmente, existem centenas de *whitepapers* disponíveis que descrevem projetos de blockchain e os problemas e soluções que buscam atacar, com uma linguagem técnica e uma maneira de escrita que se assemelha aos artigos científicos.

Embora existam centenas e até milhares de *whitepapers* disponíveis, buscamos por arquivos de projetos mais consolidados na área de blockchain, com

equipes atuando no seu desenvolvimento constantemente. Desta maneira, podemos selecionar os textos que mais agregam na construção do grafo de conhecimento, com entidades e relacionamentos de qualidade. Para cada projeto, criamos um arquivo de texto com o nome **whitepaper_<nome do projeto>**, e incluímos nele trechos descritivos sobre o projeto.

Não foram encontrados datasets prontos relacionados a projetos de blockchain, o que significa que foi necessário construir este corpus do zero. Isto abre uma boa oportunidade de contribuição no trabalho, já que este dataset será disponibilizado para uso público. Foram coletados dados de 99 projetos diferentes, com 846 sentenças no total. Os seguintes sites foram utilizados para a coleta de dados:

1. **DeFi Pulse**

   Disponibiliza uma lista de projetos em blockchain, mais especificamente relacionados a DeFi (finanças descentralizadas), um conceito novo e que vem crescendo a todo vapor, tanto em interesse quanto em desenvolvimento. Esta lista está separada em subcategorias de DeFi, como Lending (Empréstimos), Payments (Pagamentos) e outras. A lista fornece uma breve descrição do projeto, com um link para o seu site. A partir do site do projeto, conseguimos acessar seu *whitepaper* e extrair as partes que desejamos incluir no seu arquivo texto. Este site também foi bastante útil para a construção de um glossário de termos que desejamos incluir no nosso grafo de conhecimento, o que será explicado em detalhe mais adiante. [20]

2. **CoinMarketCap**

   O CoinMarketCap é o site mais popular para conferir preços de mais de 14.000 criptomoedas. A grande vantagem do CoinMarketCap é que podemos obter a lista destes projetos ordenados por capitalização de mercado, o que nos dá um norte sobre quais projetos são mais interessantes para adicionar ao nosso corpus de dados. Ao clicar em um projeto, em muitos casos haverá um botão com link direto para seu *whitepaper*, o que facilita a pesquisa. No pior dos casos, quando um projeto não possui um *whitepaper* o CoinMarketCap fornece uma descrição por vezes bastante explicativa sobre ele, que podemos incluir nos nossos dados.

3.3 Utilizando o DyGIE++ para obter entidades e relações no dataset

Com os dados coletados, são necessários alguns passos para gerar o arquivo resultado. Primeiro é necessário treinar o modelo de predição sobre o dataset desejado. Neste caso queremos treinar o modelo com o SciERC, e podemos fazê-lo com os seguintes passos:

1. **Baixar o dataset**
   1.1. O DyGIE++ fornece um script para download de todos os datasets que disponibiliza, incluindo o SciERC
2. **Treinar o modelo**
   2.1. Utilizando um script providenciado pelo DyGIE++ que aceita como argumento o nome do dataset utilizado, neste caso scierc

Estes passos foram compilados em um script de execução (start.sh), e são os seguintes:

1. **Remover a pasta contendo os arquivos texto dentro do projeto DyGIE++**
2. **Criar uma nova pasta contendo os arquivos texto dentro do projeto DyGIE++**
3. **Rodar o script para remover as quebras de linha nos arquivos texto**
   remove_breaks.sh

```
for f in corpus/whitepapers/*.txt
do
  tr '\n' ' ' < "$f" > dygiepp/data/blockchain_whitepapers/raw_data/$(basename "$f" .txt).txt
done

for f in corpus/academic/*.txt
do
  tr '\n' ' ' < "$f" > dygiepp/data/blockchain_academia/raw_data/$(basename "$f" .txt).txt
done

for f in corpus/wiki/*.txt
do
  tr '\n' ' ' < "$f" > dygiepp/data/blockchain_wiki/raw_data/$(basename "$f" .txt).txt
done
```

Figura 2: script remove_breaks.sh

4. **Formatar os arquivos com o script do DyGIE++ e copiá-los para a pasta dentro do projeto DyGIE++**

5. **Rodar o comando predict do allennlp nos dados formatados, especificando um arquivo para guardar os resultados**

```
# Predict
allennlp predict \
  models/scierc_lightweight/model.tar.gz \
  data/blockchain_whitepapers/formatted.jsonl \
  --predictor dygie \
  --include-package dygie \
  --use-dataset-reader \
  --output-file data/blockchain_whitepapers/result.jsonl \
  --cuda-device -1 \
  --silent

allennlp predict \
  models/scierc_lightweight/model.tar.gz \
  data/blockchain_academia/formatted.jsonl \
  --predictor dygie \
  --include-package dygie \
  --use-dataset-reader \
  --output-file data/blockchain_academia/result.jsonl \
  --cuda-device -1 \
  --silent

allennlp predict \
  models/scierc_lightweight/model.tar.gz \
  data/blockchain_wiki/formatted.jsonl \
  --predictor dygie \
  --include-package dygie \
  --use-dataset-reader \
  --output-file data/blockchain_wiki/result.jsonl \
  --cuda-device -1 \
  --silent
```

Figura 3: comandos do AllenNLP no script start.sh

Com a execução do comando predict, o resultado é armazenado no arquivo result.jsonl. Arquivos do tipo jsonl (JavaScript Object Notation Lines) são arquivos de texto que contém um JSON por linha. Cada objeto JSON possui os seguintes valores:

- doc_key: nome do documento
- dataset: nome do dataset com que o modelo foi treinado
- sentences: lista de listas, onde cada lista representa uma frase do arquivo e cada elemento de lista representa uma frase ou pontuação
- predicted_ner: contém uma lista para cada entidade encontrada, seguindo o formato:

    - [índice do token de início, índice do token final, tipo de entidade]
- predicted_relations: contém uma lista para cada relação encontrada, seguindo o formato:
    - [índice do token de início da primeira entidade, índice do token final da primeira entidade, índice do token de início da segunda entidade, índice do token final da segunda entidade, tipo de relação]

3.4 Construção do dygiepp-reader

Com o uso do Node.js, foi desenvolvido o programa dygiepp-reader, responsável por estruturar os resultados do DyGIE++ de maneira mais inteligível. Com o uso deste programa, conseguimos realizar as seguintes tarefas:

1. **Obter os termos que foram detectados como entidades com mais frequência**

Esta listagem foi utilizada para se obter uma rápida ideia dos termos que mais foram detectados como entidades, servindo como um reflexo dos dados que foram coletados. Também foi utilizada para destacar as entidades mais relevantes na interface web criada para visualizar os resultados.

2. **Criar um dicionário de entidades, onde podemos facilmente armazenar informações sobre cada entidade, como:**
    a. Quantas vezes ela foi detectada como um tipo de entidade
    b. As frases nas quais ela foi detectada como entidade
    c. As relações das quais ela faz parte

Um exemplo de entrada no dicionário de entidades, neste caso para a entidade "erc20":

Figura 4: estrutura de uma entrada no dicionário de entidades

O dicionário de entidades facilita a visualização dos resultados durante o desenvolvimento, e mais importante, facilita o acesso dos dados para a criação de um grafo de conhecimento visual. Este tipo de estruturação é muito mais agradável para o olho humano, pois permite que vejamos exatamente quais termos se relacionam com os outros, diferentemente de como o DyGIE++ gera os resultados no arquivo jsonl, utilizando inteiros para representar a posição inicial e final dos termos em uma sentença, como mostra a figura a seguir:

```
"predicted_relations": [[], [[16, 16, 19, 20, "USED-FOR", 4.6466, 0.9905],
```

Figura 5: trecho do arquivo resultado jsonl gerado pelo DyGIE++, representando relações para um documento

3. **Servir os dados para um cliente web através de uma API**

Foi utilizado o Node.js com a biblioteca Express para o rápido desenvolvimento de uma API para que clientes web possam consumir os dados obtidos pelo dygiepp-reader. Ao realizar a chamada, a API responde com o dicionário de entidades e a lista das entidades mais frequentes no grafo.

Este programa possui uma classe chamada Visualizer, que é inicializada recebendo um caminho para o arquivo resultado jsonl. Na contrução da classe, é inicializada uma variável contendo os JSONs em uma lista. A classe Visualizer possui os seguintes métodos públicos:

- initializeEntityDict:

  Para cada linha do arquivo .jsonl, ou seja, para cada JSON que representa um documento, esta função cria o dicionário de entidades, que será importante na construção da interface web para visualização do grafo resultado. Este dicionário permite acesso direto aos dados de uma entidade, contendo

  1. As relações das quais a entidade faz parte, seja do lado esquerdo ou direito da relação
  2. As frases onde a entidade foi detectada como tal
  3. Uma contagem para a quantidade de vezes em que o termo foi detectado como cada tipo de entidade

     Esta função recebe como parâmetro um valor true ou false que especifica se o uso de *aliases* deve ser utilizado. Caso nenhum valor seja passado, a função utiliza os *aliases*. Uma explicação sobre *aliases* e porque são importantes será vista numa seção mais adiante.

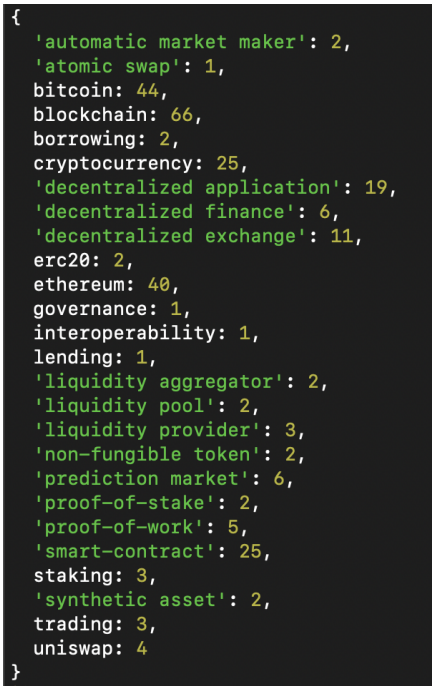
- getFrequencyList:

  Retorna uma lista com todos os termos que foram considerados entidades, ordenada pela quantidade de vezes em que o termo foi detectado como tal. Esta lista leva em conta todos os documentos do dataset, e cada elemento da lista é uma lista no formato [entidade, # de vezes detectada como entidade]. Esta lista serve dois principais propósitos:

  1. Avaliar se os resultados gerados estão em linha com o conteúdo dos documentos utilizados na geração do grafo resultado
  2. Obter um destaque maior para entidades mais frequentes na visualização do grafo resultado

- getGlossaryEntities:

  Retorna um dicionário contendo todos os termos do glossário que foram detectados como entidades e quantas vezes foram detectados



```
{
  'automatic market maker': 2,
  'atomic swap': 1,
  bitcoin: 44,
  blockchain: 66,
  borrowing: 2,
  cryptocurrency: 25,
  'decentralized application': 19,
  'decentralized finance': 6,
  'decentralized exchange': 11,
  erc20: 2,
  ethereum: 40,
  governance: 1,
  interoperability: 1,
  lending: 1,
  'liquidity aggregator': 2,
  'liquidity pool': 2,
  'liquidity provider': 3,
  'non-fungible token': 2,
  'prediction market': 6,
  'proof-of-stake': 2,
  'proof-of-work': 5,
  'smart-contract': 25,
  staking: 3,
  'synthetic asset': 2,
  trading: 3,
  uniswap: 4
}
```

Figura 6: exemplo de retorno da função getGlossaryEntities

- getNumberOfSentences:

  Retorna quantas frases o dataset possui no total. Esta informação foi utilizada para medir o tamanho do dataset

- getDocumentInfo:

  Dado o identificador de um documento, exibe na linha de comando as entidades e relações de cada frase do documento. Esta função é útil para avaliar os resultados obtidos no escopo de um documento específico.

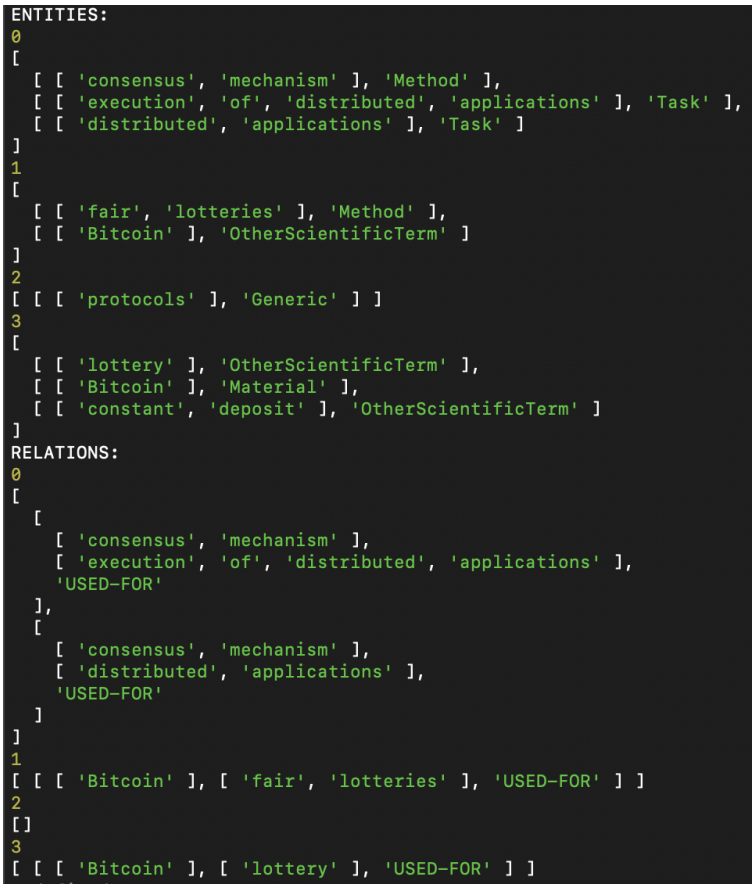


Figura 7: retorno da função getDocumentInfo para o documento com identificador "academic_6"

O dygiepp-reader possui um arquivo index.js, onde criamos instâncias da classe Visualizer para cada arquivo resultado jsonl e chamamos a função initializeEntityDict para cada uma delas. Com isso, podemos então realizar as operações desejadas.

```
import Visualizer from "./lib/visualizer.js";

const args = process.argv;
const model = args[2];

if(!model) {
    console.error('Please provide a model (yarn start <model name>)');
    process.exit();
}

const whitepaperVisualizer = new Visualizer(`./data/${model}/whitepaper.jsonl`);
const academicVisualizer = new Visualizer(`./data/${model}/academic.jsonl`);
const wikiVisualizer = new Visualizer(`./data/${model}/wiki.jsonl`);

whitepaperVisualizer.initializeEntityDict();
academicVisualizer.initializeEntityDict();
wikiVisualizer.initializeEntityDict();

console.log(academicVisualizer.getDocumentInfo('academic_6'))
console.log(whitepaperVisualizer.entityDict['bitcoin'])
```

Figura 8: arquivo index.js, que recebe o nome do modelo de predição desejado

```
Rodrigo@infras-MacBook-Pro dygiepp-reader % yarn start scierc
```

Figura 9: utilizando o gerenciador de pacotes yarn para rodar o index.js com o modelo scierc

O programa contém um diretório chamado lib, onde temos os arquivos aliases.js, glossary.js e visualizer.js. Estes dois primeiros são importantes para a avaliação e melhoria dos resultados obtidos, que veremos em mais detalhe na seção seguinte sobre avaliação dos resultados obtidos, enquanto o visualizer.js contém a classe Visualizer descrita anteriormente. Possui uma pasta contendo os scripts que rodam as análises dos resultados (analyze.js) e transformam uma linha do arquivo jsonl em um JSON dado um identificador de um documento (get-json.js). Os arquivos resultado jsonl encontram-se no diretório data, dentro da pasta com o nome do modelo de predição utilizado na geração do arquivo resultado.

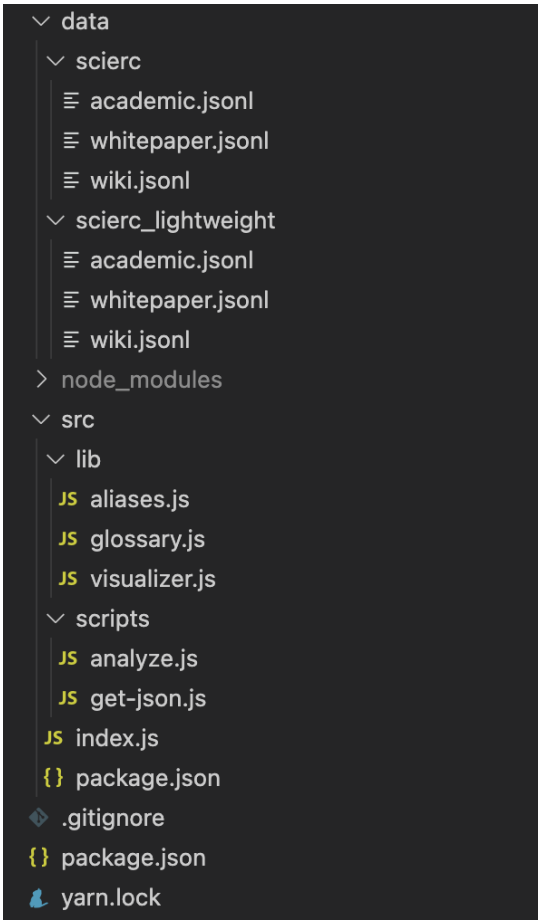


Figura 10: estrutura do programa dygiepp-reader

3.5 Avaliação dos resultados obtidos

Com os resultados obtidos a partir do modelo de predição, foi necessário fazer uma análise deles a fim de saber se são suficientemente bons para servir como base para um grafo de conhecimento que busca ser um mapa do ecossistema blockchain. Para isto, desejamos que o modelo de predição corresponda às seguintes expectativas:

1. **Gostariamos que ele seja capaz de detectar entidades e relações com uma porcentagem de acerto que se aproxime dos 64% e 55%, que foi descrito na seção 2.3**

A fim de medir a qualidade dos resultados obtidos, foram selecionados 5 frases do dataset de *whitepapers* e 5 frases do dataset de artigos acadêmicos para terem suas entidades e relações anotadas manualmente. Com estes dados

anotados, foi feita uma comparação com os resultados obtidos pelo modelo de predição do DyGIE++. Os resultados obtidos constam na tabelas seguintes.

doc_key	# entidades	# entidades corretas	% acertos	# relações	# relações corretas	% acertos
academic_2	3	3	100.00%	1	1	100.00%
academic_3	3	2	66.67%	2	2	100.00%
academic_7	3	2	66.67%	2	2	100.00%
academic_8	3	3	100.00%	1	0	0.00%
academic_14	5	4	80.00%	5	3	60.00%
whitepaper_celer-network	5	3	60.00%	1	0	0.00%
whitepaper_chainlink	2	2	100.00%	0	0	100.00%
whitepaper_polkadot	2	2	100.00%	0	0	100.00%
whitepaper_uniswap	9	7	77.78%	0	0	100.00%
whitepaper_loopring	7	3	42.86%	4	0	0.00%

% entidades corretas	% relações corretas
73.81%	50%

Tabela 3: avaliação dos resultados feita por anotações manuais de entidades e relações

Vemos então que para as entidades obtivemos uma porcentagem de acerto maior do que os 64% dados na avaliação do modelo. A porcentagem de acertos para as relações ficou um pouco abaixo do esperado, porém não o suficiente para que o trabalho fosse comprometido.

Foi feita também uma análise utilizando o Wikifier [21], um site que possibilita que o usuário envie um arquivo de texto e são retornados seus verbos, substantivos, adjetivos e advérbios destacados em cores diferentes. Para esta análise, foram selecionados resultados do modelo de predição em 10 sentenças do corpus de

whitepapers para que fossem comparados com os resultados do Wikifer. O objetivo desta análise é focado mais nas entidades - desejamos que os substantivos detectados pelo Wikifier também sejam detectados como entidades pelo DyGIE++. Vale lembrar que nem todos os substantivos do Wikifier devem ser detectados como entidades, já que as entidades se limitam aos tipos descritos na seção 2.1. Os resultados constam na Tabela 4, localizada no Anexo I ao final deste documento. Na coluna de acertos, entidades em verde foram detectadas corretamente, enquanto entidades em vermelho foram detectadas indevidamente ou deixaram de ser detectadas quando deveriam. Com esta avaliação, chegamos a uma porcentagem de acerto de 68,65% para as entidades, um valor próximo ao de 64% que foi obtido na avaliação do modelo de predição. Com estas duas análises realizadas, gostaríamos de concluir que o modelo de predição do DyGIE++ utilizando os dados do SciERC não perde sua precisão quando aplicado em outro conjunto de dados, neste caso nosso corpus de blockchain.

2. **Queremos que ele seja capaz de detectar as entidades relevantes para o propósito do projeto**

Embora o modelo de predição tenha detectado entidades e relações com um grau de precisão aceitável, precisamos também saber se as entidades detectadas são as que nos interessam, isto é, as que desejamos ver no nosso grafo de conhecimento. Para isso, foi necessário criar um glossário de termos de interesse. No programa dygiepp-reader, foi criado um arquivo glossary.js contendo termos relevantes para os interesses deste projeto. Naturalmente, foram incluídos termos relacionados à blockchain, com ênfase em conceitos de DeFi. Para a construção deste glossário, o uso do DeFi Pulse foi importante pois, como foi visto na seção 3.1, ele disponibiliza uma extensa lista de projetos de DeFi, separados por categorias, com uma breve descrição sobre os projetos. Foi feita uma leitura desta lista, e os termos mais frequentes foram adicionados ao glossário.

Foi utilizada também a função getFrequencyList da classe Visualizer do dygiepp-reader, que retorna uma lista ordenada das entidades que mais foram detectadas. Esta lista serviu não apenas para encontrar termos importantes que faltavam no glossário, mas também para encontrar entidades relacionadas a algum termo do glossário, que poderiam ser consideradas sinônimos ou variações do termo. Para os propósitos deste trabalho, nos referimos a estas entidades como

aliases. Por este motivo, foi criado também um arquivo aliases.js, que contém um dicionário onde cada chave é um termo do glossário e seu valor é uma lista com termos que devem ser considerados um *alias* desta palavra do glossário.

```
"smart-contract": ["smart contract", "smart contracts"],
```

Figura 11: Exemplo de um termo do glossário e seus *aliases*

Os *aliases* são importantes, pois permitem que possamos armazenar as relações e frases de múltiplas entidades com o mesmo significado em um lugar só. Um exemplo claro é o de entidades no plural. Como vemos na Figura 11, adicionamos "smart contract" (sem hífen) e "smart contracts" para a lista de aliases de "smart-contract", um termo do glossário. Sempre que alguma entidade detectada for um *alias*, adicionamos ela para a entrada no dicionário de entidades que corresponde à palavra do glossário a qual se refere.

Com o glossário construído, podemos então saber quantos de seus termos foram detectados como entidades. Faremos também uma análise da porcentagem de termos detectados com e sem o uso de aliases. Para isso, foi criado o script analyze.js, que calcula a porcentagem de termos do glossário que foram detectados como entidades para cada corpus, e também conta o número de relações detectadas para termos do glossário. Este script aceita como parâmetro um valor booleano indicando se os *aliases* devem ser usados na inicialização do dicionário de entidades. Foi feita a análise com e sem o uso de aliases, para os resultados do modelo scierc e scierc_lightweight. Os resultados podem ser vistos na tabela a seguir:

Corpus	Termos do glossário detectados como entidades (%)	Número de relações para termos do glossário
Whitepaper	16 de 47 termos detectados (34%)	119 relações
Academic	12 de 47 termos detectados (26%)	69 relações
Wiki	10 de 47 termos detectados (21%)	69 relações

Tabela 5: análise dos resultados sem o uso dos *aliases* para o modelo scierc

Corpus	Termos do glossário detectados como	Número de relações para termos do

	entidades (%)	glossário
Whitepaper	25 de 47 termos detectados (53%)	361 relações
Academic	17 de 47 termos detectados (36%)	155 relações
Wiki	16 de 47 termos detectados (34%)	151 relações

Tabela 6: análise dos resultados com o uso dos *aliases* para o modelo scierc

Corpus	Termos do glossário detectados como entidades (%)	Número de relações para termos do glossário
Whitepaper	18 de 47 termos detectados (38%)	118 relações
Academic	13 de 47 termos detectados (28%)	75 relações
Wiki	11 de 47 termos detectados (23%)	53 relações

Tabela 7: análise dos resultados sem o uso dos *aliases* para o modelo scierc_lightweight

Corpus	Termos do glossário detectados como entidades (%)	Número de relações para termos do glossário
Whitepaper	27 de 47 termos detectados (57%)	251 relações
Academic	18 de 47 termos detectados (38%)	145 relações
Wiki	17 de 47 termos detectados (36%)	88 relações

Tabela 8: análise dos resultados com o uso dos *aliases* para o modelo scierc_lightweight

Como esperado, o uso dos *aliases* permitiu que mais palavras do glossário constassem no dicionário de entidades, aumentando a presença deles em todos os corpus, para ambos os modelos. Também é de se destacar o fato de que, embora o modelo scierc_lightweight tenha detectado mais termos do glossário como entidades, o modelo scierc detectou significativamente mais relações com o uso de *aliases*. Combinando os três corpus de dados analisados, o modelo scierc_lightweight detectou 484 relações contendo termos do glossário, enquanto o modeo scierc detectou 667 relações, um aumento de 37%.

3.6 Desenvolvimento de interface web para visualização dos resultados

Ao longo do trabalho, o dygiepp-reader evoluiu de um programa voltado somente para uso via linha de comando para uma API simples possui uma rota para que clientes web consumam seus dados, como o dicionário de entidades e as entidades mais frequentes. Utilizei o React, uma biblioteca javascript que facilita a construção de páginas web, para desenvolver um site que realiza chamadas para a API do dygiepp-reader e exibe seus resultados. O site ainda está em desenvolvimento, e devo continuar trabalhando nele mesmo após a apresentação do projeto. Até o momento, o site possui três seções: a seção à esquerda exibe botões para as 100 entidades mais frequentes; a seção do meio mostra as relações para este tipo de entidade (no momento apenas as relações hipônimo-de, parte-de, funcionalidade-de e usado-para); a seção à direita mostra a frase onde uma relação clicada foi detectada. As três seções interagem entre si - ao clicar em uma entidade da seção esquerda as suas relações são exibidas, e estas também podem ser clicadas para atualizar a seção da direita.

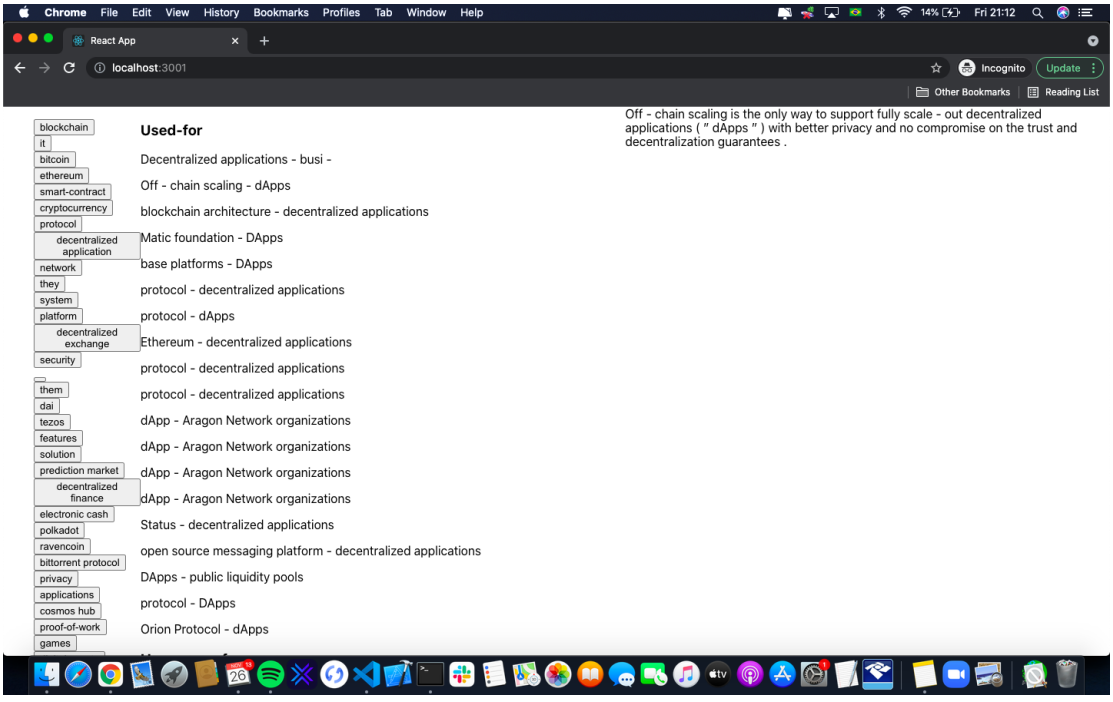


Figura 12: site desenvolvido rodando localmente, com a entidade "decentralized application" selecionada e a relação "off-chain scaling - dApps" (Used-for) selecionada

O site ainda é um rascunho do que ele idealmente poderia ser, mas com ele já podemos ver alguma utilidade para quem busca aprender sobre assuntos relacionados às entidades listadas. Na Figura 12 vemos o site em execução, após receber os dados da API do dygiepp-reader. Neste caso, a entidade "decentralized application" foi selecionada a esquerda, e logo foram listadas todas as relações das quais a palavra ou algum de seus aliases fazem parte. O uso de aliases pode ser visto devido ao fato de possuímos relaçõs para "dApps" e "decentralized app", ambos *aliases* de "decentralized application". Algumas melhorias podem ser feitas, como especificar para a API de qual dataset gostaríamos de obter informações em uma chamada. No momento, a API serve apenas os dados extraídos do dataset de *whitepapers*, mas poderia ser atualizada para servir tanto o corpus de artigos acadêmicos quanto o de textos da Wikipedia. Seria interessante também combinar os resultados, juntando os dados obtidos dos três datasets. Ao clicar em uma relação, a frase de onde foi retirada é exibida, mas gostaríamos de saber também de qual documento e em qual número de frase ela foi retirada. Existem também alguns problemas solucionáveis como a presença de relações repetidas.

4. Conclusão e futuras pesquisas

O desenvolvimento deste projeto teve inspiração no trabalho de Yi Luan *et al* [14], onde foi desenvolvido o framework SciIE para extração de entidades e relações de um conjunto de dados, produzindo um um grafo de conhecimento que pode ser utilizado para organizar informações de um domínio específico, tornando-se uma ferramenta importante para fins de pesquisa. Surgiu então a hipótese de que este modelo de construção de grafos de conhecimento poderia ser aplicado não apenas à área de inteligência artificial como foi no caso do SciIE, que utiliza o dataset relacionado a IA, o SciERC, mas também à área de blockchain.

Se o artigo que introduz o SciIE ajudou a formular a ideia inicial deste projeto, o trabalho de David Wadden *et al* [22], que introduz o framework DyGIE++, foi essencial para a realização dele. O DyGIE++ foi desenvolvido por alguns dos mesmos autores do SciIE, e foi descrito em mais detalhe na seção 2.3. Este programa abaixou a barreira de entrada para a aplicação de modelos de predição para realizar a extração de entidades e relações, tarefa fundamental na construção dos grafos de conhecimento que este projeto buscou construir. Vale ressaltar

novamente a utilização do AllenNLP, que forneceu comandos via linha de comando que foram rápidos e intuitivos, permitindo que um projeto de graduação pudesse fazer uso de conceitos que geralmente ficariam limitados para trabalhos mais avançados, como de mestrado e doutorado.

Como foi visto na análise dos resultados obtidos na seção 3.3 e com a visualização do grafo de conhecimento completo, a hipótese de que o SciERC poderia ser utilizado com o DyGIE++ para extração de entidades e relações em dados relacionados à blockchain parece correta. Com isto em mente, sugerimos que este trabalho seja aplicado em outras áreas no domínio de computação, sejam elas áreas mais estabelecidas ou novatas como é o caso da blockchain. A aplicação desta técnica de construção de grafos de conhecimento a outras áreas de computação deve trazer importantes desenvolvimentos tanto nas áreas em questão quanto para a técnica em si.

Grande parte da motivação deste projeto se deve ao fato da área de blockchain ser tão nova, com poucas ferramentas úteis para auxiliar a pesquisa sobre projetos e tecnologias emergentes neste domínio. A maioria das ferramentas disponíveis são voltadas para a análise de dados relacionados a estatísticas da blockchain (número de transações, valor de taxas, etc.), movimentações de preços de criptomoedas e análise técnica de gráficos de preço [24]. Esta falta de uma ferramenta que permitisse a visualização do ecossistema blockchain que não se limitasse a estatísticas e dados de mercado foi vista como uma oportunidade para desenvolver um projeto que fosse relevante neste domínio novo e inovador.

Embora o desenvolvimento deste trabalho não exigisse a construção de código que realizasse as tarefas de extração de entidades e relações, foram necessários conhecimentos de estruturas de dados, Javascript e desenvolvimento web para entregar os produtos deste trabalho. Foi desenvolvido o dygiepp-reader, um programa em Javascript para manipulação e estruturação dos resultados do DyGIE++, e também uma interface web para visualização dos resultados obtidos. Como API, o dygiepp-reader ainda pode evoluir para possuir rotas que aceitem parâmetros, como o corpus de dados para o qual o cliente deseja obter os resultados. Também é do interesse do autor que esta API seja capaz de combinar os resultados dos diferentes corpus de informação a fim de criar um grafo de conhecimento unificado. A interface web também pode evoluir para se tornar mais interativa, permitindo que o usuário saiba exatamente de qual documento e de qual

frase uma entidade ou relação foi retirada. Como este trabalho possuiu um foco especial em coletar dados relacionados a projetos de blockchain, seria interessante permitir que o usuário busque as entidades e relações de um projeto específico.

Ambos os produtos, tanto o programa dygiepp-reader quanto a interface web serão disponibilizados em domínio público. O primeiro será enviado à equipe que criou do DyGIE++, pois pode ser relevante para quem busca utilizar seu framework e deseja obter os resultados em uma interface web por meio de APIs. Já o segundo será postado em fóruns online de blockchain, para que entusiastas de criptomoedas como o próprio autor deste projeto possam fazer uso deste site para auxiliar na pesquisa de projetos e suas tecnologias.

5. Referências bibliográficas

[1] Haber, S., Stornetta, W. S. How to Time-Stamp a Digital Document https://www.anf.es/pdf/Haber_Stornetta.pdf acesso em novembro/2021

[2] Conway, Luke Blockchain Explained Blog post, acesso em outubro/2021

[3] Nakamoto, Satoshi *Bitcoin: A Peer-to-Peer Electronic Cash System*, Blog post, acesso em abril/2021

[4] Amadeo, Kimberly What Caused 2008 Global Financial Crisis, Blog post, acesso em abril/2021

[5] MIT Media Lab 51% Attacks https://dci.mit.edu/51-attacks, acesso em novembro/2021

[6] Bogart, Spencer Bitcoin is (Still) a Demographic Mega-trend: Data Update, Blog post, acesso em abril/2021

[7] The Nielsen Company Nielsen Global Money Report October 2016, Blog post, acesso em outubro/2021

[8] Ethereum Foundation Introduction to Smart Contracts Blog post, acesso em novembro/2021

[9] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Shahbaz Khan, Rajiv Suman Blockchain technology applications for Industry 4.0: A literature-based review, Blockchain: Research and Applications, Elsevier, acesso em novembro/2021.

[10] Dineley, Doug The greatest open source software of all time acesso em abril/2021

[11] Finley, Klint Out in the Open: Teenage Hacker Transforms Web Into One Giant Bitcoin Network acesso em abril/2021

[12] Schär, F. Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets https://files.stlouisfed.org/files/htdocs/publications/review/2021/04/15/decentralized-finance-on-blockchain-and-smart-contract-based-financial-markets.pdf acesso em novembro/2021

[13] CoinMarketCap: Cryptocurrency Prices, Charts And Market Capitalizations

[14] Yi Luan, Luheng He, Mari Ostendorf, Hannaneh Hajishirzi "Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction", arXiv preprint, https://arxiv.org/pdf/1909.03546v2.pdf acesso em março/2021

[15] Luan, Yi Supplementary Material http://nlp.cs.washington.edu/sciIE/annotation_guideline.pdf acesso em março/2021

[16] Dessì, Danilo, et al. "Generating knowledge graphs by employing Natural Language Processing and Machine Learning techniques within the scholarly domain." *Future Generation Computer Systems* 116 (2021): 253-264 .https://arxiv.org/pdf/2011.01103.pdf acesso em novembro/2021

[17] Guilei Wang et al 2020 "Research on Key Technologies of Knowledge Graph Construction Based on Natural Language Processing", J. Phys.: Conf. Ser. 1601 032057, https://www.researchgate.net/publication/343719031_Research_on_Key_Technologies_of_Knowledge_Graph_Construction_Based_on_Natural_Language_Processing acesso em novembro/2021

[18] Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., & Zettlemoyer, L. AllenNLP: A Deep Semantic Natural Language Processing Platform (Version 2.8.0) [Computer software]. https://doi.org/10.18653/v1/W18-2501 acesso em novembro/2021

[19] IBM Cloud Education Knowledge Graph https://www.ibm.com/cloud/learn/knowledge-graph acesso em novembro/2021

[20] DeFi Pulse The DeFi List https://defipulse.com/defi-list/ acesso em Outubro/2021

[21] Wikifier http://wikifier.org/ acesso em Setembro/2021

[22] David Wadden, Ulme Wennberg, Yi Luan, Hannaneh Hajishirzi "Entity, Relation, and Event Extraction with Contextualized Span Representations" https://aclanthology.org/D19-1585.pdf acesso em Setembro/2021

[23] Luan, Y. *et al* SciERC dataset https://paperswithcode.com/dataset/scierc acesso em novembro/2021

[24] Kibet, L. 10 Best Crypto Research Tools You Must Have as an Investor, Blog post, https://www.gobankingrates.com/investing/crypto/best-crypto-research-tools/ acesso em novembro/2021

Anexo I - Tabela 4: Comparação das entidades extraídas com os resultados do Wikifier

Documento (# sentença)	Entidades DyGIE++	Resultado Wikifier	Resultados	% de acertos
whitepaper_aave (1)	[['Aave'] , 'Method']	Aave is a decentralized finance protocol that allows people to lend and borrow crypto.	1. Aave 2. decentralized finance protocol 3. lend 4. borrow 5. crypto	20%
whitepaper_akropolis (15)	[['Akropolis', 'protocol'], 'Method'] [['digital', 'financial', 'landscape'], 'OtherScientificTerm'], [['digital', 'financial', 'organizations'], 'OtherScientificTerm']	The Akropolis protocol aims to create this new digital financial landscape by providing a unified program interface for the cooperation and exchange of value of digital financial organizations.	1. Akropolis protocol 2. digital financial landscape 3. digital financial organizations 4. unified program interface	75%
whitepaper_avalanche (6)	[['Snow', 'protocol', 'family'], 'Method'], [['it'], 'Generic'], [['internet', '-', 'scale', 'electronic', 'payment', 'system'], 'Method'], [['-', 'scale', 'electronic', 'payment', 'system'], 'Method'], [['Avalanche'], 'Method'], [['large', 'scale', 'deployment'], 'Task']	The paper describes the Snow protocol family, analyzes its guarantees, and describes how it can be used to construct the core of an internet-scale electronic payment system called Avalanche, which is evaluated in a large scale deployment.	1. Snow protocol family 2. It 3. internet-scale electronic payment system 4. Avalanche 5. large scale deployment	100%
whitepaper_bancor (1)	[['Bancor', 'Protocol'], 'Method'], [['automatic', 'price', 'determination'], 'Task'], [['autonomous',	The Bancor Protocol enables automatic price determination and an autonomous liquidity mechanism for tokens on smart	1. Bancor Protocol 2. automatic price determination	80%

	'liquidity', 'mechanism'], 'Method'], [['smart', 'contract', 'blockchains'], 'OtherScientificTerm']	contract blockchains.	3. autonomous liquidity mechanism 4. smart contract blockchains 5. tokens	
whitepaper_celer-network (14)	[['Celer', 'Network'], 'Method'], [['layered', 'architecture'], 'Method'], [['clean', 'abstractions'], 'OtherScientificTerm'], [['rapid', 'evolution'], 'OtherScientificTerm'], [['generalized', 'state', 'channel'], 'OtherScientificTerm'], [['sidechain', 'suite'], 'OtherScientificTerm'], [['provably', 'optimal', 'value', 'transfer', 'routing', 'mechanism'], 'Method'], [['optimal', 'value', 'transfer', 'routing', 'mechanism'], 'Method'], [['throughput'], 'Metric'], [['development', 'framework'], 'Method'], [['runtime'], 'OtherScientificTerm'], [['chain', 'applications'], 'Task'], [['cryptoeconomic', 'model'], 'Method'], [['network', 'effect'], 'OtherScientificTerm'], [['stable', 'liquidity'], 'OtherScientificTerm'], [['off', '-', 'chain', 'ecosystem'], 'Method'], [['-', 'chain', 'ecosystem'], 'Method']	Celer Network embraces a layered architecture with clean abstractions that enable rapid evolution of each individual component, including a generalized state channel and sidechain suite that supports fast and generic off-chain state transitions; a provably optimal value transfer routing mechanism that achieves an order of magnitude higher throughput compared to state-of-the-art solutions; a powerful development framework and runtime for off-chain applications; and a new cryptoeconomic model that provides network effect, stable liquidity, and high availability for the off-chain ecosystem.	1. Celer Network 2. layered architecture 3. clean abstractions 4. rapid evolution 5. generalized state channel 6. sidechain suite 7. provably optimal value transfer routing mechanism 8. throughput 9. development framework 10. runtime 11. cryptoeconomic model 12. network effect 13. stable liquidity 14. off-chain ecosystem 15. individual component 16. off-chain transitions 17. off-chain applicati	0,77%

			18. ons high availability	
whitepaper_celsius (4)	[['hedge', 'funds'], 'Material'], [['crypto', 'funds'], 'Material'], [['cryptocurrencies'], 'OtherScientificTerm']	Celsius will generate income for coin holders by allowing hedge funds and crypto funds, to create short positions on cryptocurrencies using actual coins borrowed from the Celsius pool of lenders.	1. hedge funds 2. crypto funds 3. cryptocurrencies 4. Celsius 5. generate income 6. coin holders 7. create short positions 8. Celsius pool of lenders	0,375%
whitepaper_chainlink (8)	[['on', '-', 'chain', 'components'], 'Method'], [['-', 'chain', 'components'], 'Method'], [['chain', 'components'], 'Method'], [['ChainLink'], 'Method'], [['external', 'connectivity'], 'OtherScientificTerm'], [['software'], 'Generic'], [['network'], 'Generic']	We describe the on-chain components that ChainLink provides for contracts to gain external connectivity, and the software powering the nodes of the network.	1. on-chain components 2. Chainlink 3. external connectivity 4. Software 5. network 6. contracts 7. nodes	0,71%
whitepaper_compound (13)	[['decentralized', 'system'], 'Method'], [['frictionless', 'borrowing', 'of', 'Ethereum', 'tokens'], 'OtherScientificTerm'], [['money', 'markets'], 'OtherScientificTerm'], [['safe', 'positive', '-', 'yield', 'approach'], 'Method'],	In this paper, we introduce a decentralized system for the frictionless borrowing of Ethereum tokens without the flaws of existing approaches, enabling proper money markets to function, and creating a safe positive-yield approach to storing assets.	1. decentralized system 2. frictionless borrowing of Ethereum tokens 3. money markets 4. safe positive-yield approach	80%

	[['-', 'yield', 'approach'], 'Method']		5. storing assets	
whitepaper_cosmos (15)	[['Cosmos', 'Hub'], 'Material'], [['governance', 'mechanism'], 'Method'], [['network'], 'Generic']	The Cosmos Hub is a multi-asset proof-of-stake cryptocurrency with a simple governance mechanism which enables the network to adapt and upgrade.	1. Cosmos Hub 2. governance mechanism 3. network 4. multi-asset proof-of-stake cryptocurrency	0,75%
whitepaper_elrond (3)	[['Elrond'], 'Method'], [['architecture'], 'Generic'], [['genuine', 'state', 'sharding', 'scheme'], 'Method'], [['energy', 'and', 'computational', 'waste'], 'OtherScientificTerm'], [['distributed', 'fairness'], 'OtherScientificTerm']	This paper proposes Elrond, a novel architecture which goes beyond state of the art by introducing a genuine state sharding scheme for practical scalability, eliminating energy and computational waste while ensuring distributed fairness through a Secure Proof of Stake (SPoS) consensus.	1. Elrond 2. architecture 3. genuine state sharding scheme 4. energy and computational waste 5. distributed fairness 6. practical scalability 7. Secure Proof of Stake (SPoS) consensus	0,71%

% entidades corretas
68,65%