SocialAI

Stay social, stay connected

ABSTRACT

We developed SocialAI, which can readily provide you with all social media links for a given token address. We leverage Natural Language Processing (NLP) concepts to do this accurately and efficiently. To summarize, we bring to the crypto space:

- fast and reliable access to social media links (Twitter, Telegram, Medium, etc.) (✓)
- a **telegram bot** that can provide this information in any chat or group (✓)
- a novel **webshop-like DAPP** to obtain social credits (✓)
- hold incentives via our unique DAPP (✓)
- **dev dashboard** to upload social links
- **live feed** of social updates

Bullet points with the checkmark (✓) have already been completed. We will occasionally update this whitepaper to keep this and the information regarding the algorithm up to date.

1. INTRODUCTION

In crypto, we have dozens, and sometimes hundreds of launches a day. You see someone share a token address. The price went up 300% already. Your veins dilate. Adrenaline enters your bloodstream. Your hands start to sweat. Is this it? Is this the token you have been waiting for? Knowing crypto you are cautious. You paste the token address in a scam checker and want to find the social media channels. How is the community? Does the website look good? But you can't find it and none of your friends seem to know it. No longer. **SocialAI is that friend you need**. It scans the contract source code, Twitter, and webpages to find links for **Telegram, Twitter, Medium, and the token website**. SocialAI is equipped with state-of-the-art Natural Language Processing (NLP) allowing it to quickly dissect large corpora of text. With its own telegram bot, you can conveniently obtain token information.

2. BACKGROUND

When reading this you are probably familiar with crypto so this will mainly focus on familiarizing you with some Natural Language Processing (NLP) concepts. NLP is one of the branches of AI. in NLP we predominantly focus on understanding text or spoken words. Since unfortunately we do not often have spoken or video material in crypto our main aim is understanding text. In the last decade, NLP has been extremely useful to extract simple information from large quantities of text. Aided by technological advances NLP could search words and understand simple concepts at speeds unparalleled by humans. Yet NLP could not understand the text as we humans can. This revolutionary changed with the development of models like BERT, GTP3, T5, and PALM. These models filled the gap between simple text-parsing and human-like interpretation. Going from recognizing simple words to understanding context, establishing long-range word dependencies and being able to resolve complex text ambiguities.

We are now capable of reliably "tagging" words in large texts. With "tagging" we mean labeling words as things like "X is a company name", "X is an adjective of Y", etc. In jargon, this is referred to as **entity recognition** and **dependency extraction**. An example of this, using Stanford's NLP model:

Figure 1; example of dependency. *We see an nsubj dependency linking "website" to "https://...". It "understands" that the subject of this sentence is "https:///..." being referred to by "website". See the Stanford description for more detail**

For us humans it's directly evident that the word "website" refers to the link "*https://www.token.com*". However, this is much less obvious for computers. For example, if we write "*The website of token X is nice, but https://www.google.com is so useful*" we again have the word "website" and a valid url, but these are totally unrelated. Interestingly, the relatively simple coreNLP understands the absence of this connection:

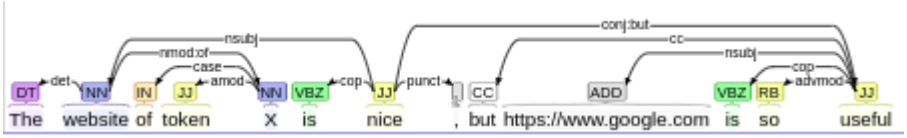

Figure 2; example of the absence of a dependency. *While, just like in Figure 1, we have both a mention of the word "website" and a valid url CoreNLP does not create a dependency between the url and the word "website"*[1]

For fun, scroll through these ambiguous newspaper headlines. Take "*Dead Expected To Rise*". This sentence has two different interpretations, and we can probably agree on the correct one, but this is far from obvious for a computer. To dive into this, let's understand the difference between **syntax** and **semantics**. With syntax, we refer to grammar, whereas semantics is the meaning being conveyed. We can write a semantically correct sentence, but that does not mean that it makes sense, or in fact that we even understand it ourselves. Take the famous sentence from Chomsky: "*Colorless green ideas sleep furiously*". Grammar-wise this is fine, however, it does not carry any meaning, and it does not make sense semantically. So if a computer can recognize verbs and nouns we only reach the level of syntax not of semantics. We as humans unconsciously process the syntax to understand the message it conveys. Computers can't do this. This is arguably the toughest part of linguistics. Put simply, a kind of syntactical question would be "What is the website url?" but a semantic question would be "What would be the token website url" - this involves interpretation and context incorporation.

We take advantage of the progress NLP has booked and apply this to crypto text corpora. We initially use relatively simple models (see METHODS below) and then expand to deep-learning guided models (see THE FUTURE).

3. METHODS

We will briefly describe how the alpha version of our code works and how we will improve this with raised funds. We can roughly divide our *alpha* algorithm into three blocks: (1) data acquisition, (2) link extraction, and (3) data storage. After this, we will explain how the DAPP works and the measures we take to ensure integrity, and finally, future methodology where we touch on developments we have planned.

---

[1] *We just produced these images as a simple example. In reality, such connections are obviously much more complex. For our approach, read "Methods"*

3.1 SOCIAL AI ALGORITHM

Data acquisition

To provide you with the latest social media links we have to scan the relevant resources. Quite often developers provide the social media links in the contract source. This is arguably the most reliable place to find a project's social media links. Therefore we consider this the first-line reliability level (indicated by 🟢).

Now let's say we don't find any social media links, but we do find the website link. Since this link was in the contract source code we are pretty confident that it is related to the project. Our algorithm will navigate to the website and employ some text scraping to extract the relevant links. Although pretty confident, but not direct from the source, we mark this result in orange (🟠).

What if everything is empty up until now? In that case, it's time to explore the world wide web. We use advanced Twitter queries to retrieve information. Without going into too much detail this for example looks for tags, chart urls, swap links, etc. Moreover, we take into account whether tweets are retweets, direct tweets, or quotes. Basically, anything that might help us find the social media links. This is somewhat unreliable as Twitter is just a free text space. In our tests, querying Twitter was highly effective but we can't guarantee correctness. For this reason, we mark this red (🔴). Our telegram bot directly provides the links so it will just be a matter of seconds to validate this yourself. It is possible that we still did not obtain any social links. We hit the end of the trail here but let's say we did find a website link on Twitter. If so, we navigate to the webpage and try to extract all relevant links. Again this is somewhat unreliably and hence marked 🔴.

We did consider other sources, such as Reddit. However, we noticed that a lot of recently launched meme tokens did not create posts here. We probably passed the "moonshot post" era, nevertheless, we can easily integrate this when deemed beneficial. Our approach is modular and uses a voting mechanism to select links from multiple resources. Hence we can simply plug in more resources, such as Facebook, DexTools, etc., and automatically let these be used in link extraction.

Link extraction

Currently, we use an NLP model, similar to that of CoreNLP to extract relationships as discussed in the previous section. We also have a tiny model that we trained on 10K observations to differentiate actual meaningful links from those of "influencers". Especially relevant when we had the "Twitter" hype where developers often shared links to tweets of influencers. These links referred to tweets that the token was based on instead of their own.

Moreover, when inspecting the weights of this model we also noticed that we quite often found links for charts, swaps, or coin-listing websites. This simple model will deal with interpreting that. We also perform some other sanity checks. For example, as we all know, scammers often just link a website that is not even live. We check the status codes when requesting such websites to avoid listing them. To keep track of the performance of our link extraction we monitor the retrieved links for all submitted tokens. If we notice that links are missing, or the wrong ones are extracted, we will just update the extraction model.

Data storage

We observed that some recently launched AI tokens experienced difficulty processing the high demand of requests. Mostly also because they relied on OpenAIs APIs that occasionally go offline when traffic is high. We also have to rely on external APIs, such as those for Etherscan to obtain the source code or the Twitter API, but we have some checks and databases in place to mitigate problems.

For example, as soon as social information is requested we check our local database. If this token was searched <1 minute ago we immediately return the results. We assume that we could not acquire enough new information making it worth querying again. Of course, in some cases, this might be the case, but this will be of negligible impact. This way of buffering drastically reduces the interactions we have with external services and also smoothens user experience as replies from our SocialAI friend will almost be instantaneous (when recently requested before by someone else).

3.2 WEB-ARCHITECTURE AND DAPP

Aside from core functionality, arguably one of the most important aspects of utility tokens is the incentive to hold. In our case tokens give you access to so-called “social credits”. Social credit can be spent searching for social links. A bit more on this later, first we want to emphasize that this is not taken care of by a conventional smart contract.

We develop a novel smart contract that combines a webshop with staking. Let us clarify this. Most utility tokens have a tier system where you have to hold an X amount of tokens to benefit from certain functionalities. An early buyer, holding a large portion of the total supply, could easily gain access to all functionality, whereas a genuine believer might not possess sufficient funds to purchase the required amount of tokens. For this reason, we developed a smart contract where not only the number of tokens matters but also your belief in SocialAI. We reflect the belief in a staking period. The longer you stake the more social credits a single token gives you. To illustrate, let's say you spent $1 to obtain 10 tokens. You can choose to stake 3 hours with a ratio of 1 token = 1 social, or stake 1 week with a ratio of 1 token = 10 socials (this is just hypothetical and will change for our actual DAPP). We understand that not everyone wants to lock their tokens so we give two credits a day to every holder. This can then be topped up with whatever staking period you feel comfortable with. We also allow upgrades, let's say you have all your tokens in one of our packages but then decide you actually want more social credits. You can simply “upgrade” this in our DAPP. When upgraded you simply transfer your lock days to the new packages such that you don’t have to start all over again. When your lock time expires you can withdraw all your tokens again.

Getting back to how these “social credits” work. As soon as a package is “purchased” (i.e. tokens are staked) an input field will appear. You can provide the telegram handle you want to use to perform social scans. Your daily requests and handle are tracked by our telegram bot that, for each search, decrements your social credits by one. Every 24 hours we refill your social credits. Ideally, we would track the number of credits left on the blockchain. We acknowledge this would be more transparent, however extremely expensive (~4$ currently) for every simple decrement transaction. We think this would be prohibitively high, so we decided to track the packages and staking in a smart contract and store the social credits in a database. This database runs on a Firebase server with over 99.99% uptime. Moreover, we have automated backups in place and our smart contract is equipped with an “emergency unlock” that, when activated, allows everyone to withdraw their tokens regardless of the remaining lock period. With these measures in place, we don’t foresee any problems. Moreover, we want to highlight that all transactions regarding tokens are happening on-chain.

We also want to touch on the security of the data we collect. We understand you might be wary to provide a telegram handle linked to a transaction. By the way, it’s completely fine to use a “burner” telegram for this which you only use to perform scans. Aside from this, we irreversibly hash the sender's address and store this together with the telegram handle. So if regardless of our security precautions the database would be breached there is no way to link the actual wallet address to the telegram handle. We want to accentuate that a breach is highly unlikely, if not impossible. Regarding the DAPP contract, we, as developers can only add packages and update this (again with restrictions). There is no way we can lock tokens indefinitely, obtain them ourselves, or prevent returning them. We

encourage you to just read through the contract yourself and ask us questions when something is unclear.

3.3 FUTURE METHODOLOGY

Everything mentioned before is just the beginning of SocialAI. We have much more planned for the future. We will continuously keep improving our algorithm. With funds raised from the tax, we will start incorporating more complex models, and evoke deep learning to elevate SocialAI's human-like interpretation skills. As we expect usage to increase sharply we will also spin up several more servers to keep up with the demand and, when necessary, elevate our API plans.

Aside from the improvement of that what is already there, we will start building our social dev dashboard. In our opinion, Dextools charges a ridiculous amount just to show social media links on their website. This is a matter of updating a single entry in a database which is practically free. We want to make this much more accessible and also make the provided socials easily findable. We don't want to open Dextools to find a link, we want to directly see that in our telegram - something SocialAI will soon provide.

In addition to the developer uploading such social media links, we will build a DAPP-like upload and voting system so users can upload social links. We recently had these "larp" tokens where developers and holders communicated via transactions and made their own community telegrams. While we could technically extract links encoded in such transactions on EtherScan there is no way to estimate its accuracy. In fact, this would open up space for malicious intent to purposely distribute scam Telegram links to be captured by SocialAI. For such reasons, and just for rating credibility in general, we think a DAPP-like voting system displaying votes in Telegram is a useful addition.

Lastly, we will create SocialTools. A new platform where developers and investors come together. This will include a set of tools aiming to optimize the currently overwhelming social media realm related to crypto. These tools will provide investors with more reliable and easy-to-access information, and provide developers the opportunity to more easily reach an audience with important notifications

4. ROADMAP

We currently have the plan as shown below. This is subject to change depending on development and market sentiment. When we spot gaps regarding socials in crypto that we can fill we will prioritize this. Another scenario would, for example, be the release of public models, like GPT3, that can drastically improve performance. Similarly, we of course will try to maximize exposure regardless of whether that is being mentioned here or not.

Stage 1

- Birth of our friend, SocialAI
- Fully functional webshop DAPP
- *Alpha* version of SocialAI scanner
- Initial marketing
- Whitepaper

Stage 2

- Video demo of the DAPP and SocialAI
- Apply CMC and CG
- Development of dev social dashboard
- advertisement slot on the bot

Stage 3

- CMC and CG listed
- Contract audit
- Paid ad campaigns
- Yahoo Finance and Business insider articles
- Development of live feed

Stage 4

- Expanding to other chains (in case of high demand this will move up)
- New DAPP features like comments and ratings
- exposing API endpoints for seamless integration with other platforms
- Release of SocialTools

Stage 5

- Global expansion, at this point, SocialAI is not just a friend anymore, it's family

5. TOKENOMICS & TAX

We will create a fair launch on Pinksale to which we will deposit 60% of the tokens. The remaining 40% will be divided into 30% for liquidity, 5% for listing on central exchanges (cex), and 5% for marketing. The liquidity will be initially locked for three months and then, when everything is in place, extended. Listing on central exchanges will not be possible directly after launch so we will lock these for two weeks. We don't require all the marketing tokens unlocked, in fact, this is probably discouraging for buyers as well so we will use a vested lock here to gradually unlock the tokens. Marketing is also included in our tax so if enough volume is generated we won't sell off the 5% marketing tokens but burn them instead.

Token distribution

We will have a total supply of 1 billion tokens. Of this:

- *90% liquidity*
- *5%* cex
- *5%* marketing

Transaction tax

This will be our initial tax distribution. May we need to improve liquidity we will increase the percentage going to liquidity and decrease dev/marketing tax. Our tax will never exceed 10%, which is hardcoded in the contract. Also, wallet-to-wallet transfers are not subject to any tax.

Total buy tax: 6%

- 2% liquidity
- 2% marketing
- 2% dev

Total sell tax, 8%

- 3% liquidity
- 3% marketing

- 2% dev

6. CONCLUSION

SocialAI uses state-of-the-art natural language processing to quickly search the realm of crypto-related texts. With high credibility, it dissects texts and identifies social media links to platforms such as Twitter and Telegram. Armed with a unique DAPP and a clear plan for the future we are confident that SocialAI will soon be an indispensable asset. Stay social, stay connected.