BITCEO/Zeniius – Activate the Power of Business Network

(BitCEO is the currency on Zeniius - a dynamic platform for CEOs networking and doing business)

The BitCEO - Zeniius Team

Jun 23th 2018

ABSTRACT: In today's world, all businesses need to connect with other business partners to learn, collaborate and stay competitive. Networking physically requires time, efforts and skills which not all CEOs possess. Our solution is to facilitate an online social network platform, known as **Zeniius,** for CEOs to interact, network and utilize their existing abilities and connections to do business globally. Ultimately, CEOs are able to assist other members of the ecosystem to quickly and efficiently resolve business demands. This paper describes the application **Zeniius** in terms of its architecture. It also depicts the implementation of Blockchain in **Zeniius** as the optimal solution which ensure transparency and honesty in business transactions between CEO users. Additionally, it describes the Artificial Intelligence technology employed by **Zeniius**. Overall, A.I. enhances the rapid delivery of business demand to the ideal solver, as well as optimizes the rating system of CEOs in Zeniius.

KEYWORDS: CEO Network, Zeniius, Blockchain, Artificial Intelligence

1. Introduction

An estimate of 115 million businesses around the world and their directing CEOs are challenged with a myriad of business request daily. To stay relevant and compete successfully in an ever increasingly demanding consumer market, it becomes crucial for CEOs to quickly resolve those commercial needs.

As a result, the ability to connect with the right CEO in the shortest time possible and minimal stress would greatly help businessmen stay on top of their operations. At the same time, CEOs constantly need to connect with other business partners to expand their network and ready themselves for new opportunities.

Zeniius is developed as a holistic solution to satisfy business demands without geographical restriction. With Zeniius, CEOs around the world can expand their network and do business together. Members of Zeniius have a lot in common, which allows them to easily get together for sharing knowledge, scheduling meetings and on-site visits to each other's company to learn about different business models, creating forums for various investment channels (including real estate, cryptocurrency, or other social activities such as charity events, etc.) As a result of these interactions, the network has been growing and creating values. For nearly a year, this network has been active in real estate projects; financial investments were generated with a total transaction value amounting up to 25 millions USD. This significant sum has been reinvested in the network and has continued to create values. These early milestones motivate us to strive towards perfecting the platform. With that said, Zeniius acknowledges the limitations we currently experience and use them positively as catalyst for the team to enhance our platform, creating a better connective environment for CEOs around the world, working towards a better world together.

In this paper, we explain the reasons behind Zeniius in Section 2. The entire Zeniius system will be illustrated in details in Section 3 & 4. Section 5 & 6 will illustrate how Blockchain & AI will be integrated in Zeniius. Lastly, we classify future work in Section 7.

2. Related Work

The figure below illustrates the interactions between CEOs, based on real life synergy, depicted in a four-leveled model termed as "4J Model".

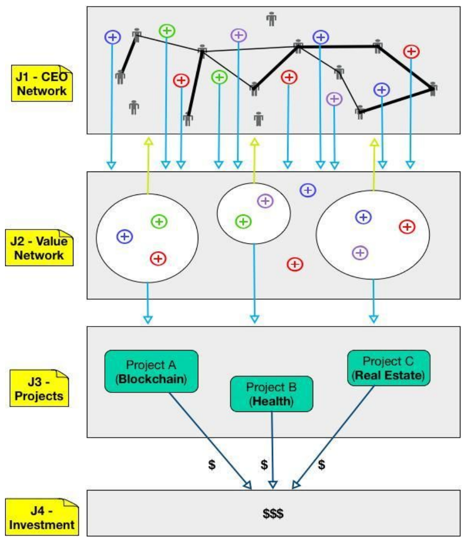

Fig. 1: CEO network model (J4 Model)

- <u>(J1) CEO Network</u>: The interaction (circle) of the CEO in real life through direct or indirect activities such as CEOs meetings, gatherings, sports meetups, formal meetings, etc. Some of these interactions will definitely create value. Such values include relationships, contracts, needs, capabilities, etc. of a certain CEO. (Circle with plus sign: Value to be defined).

- <u>(J2) Value Network:</u> Create a customizable environment from valuable network interactions from (J1) and form a Value Network. This network allows for the integration of valuable interactions.

E.g.: Combining CEO A's need to purchase a car with that to sell one of CEO B's Company into a Sales Contract. This network also allows the amplification of these values by creating complementary activities to increase interactions in (J1).

- (J3) Projects: From the Value Network (J2), new projects or larger ones can be created, requiring a combination of values in the Value Network. E.g.:
  CEO A's product distribution network, CEO B's cake factory and a marketing system solidified by CEO C can be combined into a branded product which can be sold to the market.
  A construction industry CEO is implementing a large project. This CEO will likely need to source support from all the CEOs in related fields such as real estate, architecture, planning, interior design, etc. to complete the construction work as well as in the process of adding finishing touches for the project.

- (J4) Investment: Projects generated from (J4) can generate large and collectible cash flows that can be used to reinvest in the network as well as in the community.

The core of this model (J4-Model) is the values that can be created based on the relationships and interactions of CEOs. This model only works well if there is a Value Network (J2) environment, which allows members of the interactive network to amplify these values and, at the same time, creating the premise for generating communities and projects in (J3).

Upon referencing current social network platforms, we observe their incapability to satisfy real-life demands for the CEO network. CEOs connecting and doing business together require an environment that has optimal transparency and stringent regulations. Moreover, the system has to be sufficiently capable technology wise to satisfy CEOs' needs and increase the value of the CEO network.

Features	**Linkln**	**Opport-unity**	**Twitter**	**Facebook**	**Steemit**
Mobile, Tablet, Desktop	Yes	Yes	Yes	Yes	No
Group Chat	Unlimited	Unlimited	Unlimited	Upto 150	No
Business Filter	Yes	Yes	No	No	No
Blockchain Enabled	No	No	No	No	Yes
Digital Wallet	No	No	No	No	Yes
Digital Currency	No	No	No	No	Yes
AI Enabled	Yes	No	Yes	Yes	No
CEO User	No	No	No	No	No

Table.1: Compare between other platforms

As such, Zeniius was created as a holistic solution. What distinguishes Zeniius from other social platforms is the focus on CEOs and business owners. This group normally possesses professional qualifications, financial potential, extensive human resources as well as capable facilities. Therefore, they have a real potential to resolve business demands posted by others in the Zeniius community. More than that, referral from from other platforms, Zeniius is integrated blockchain technology, artificial intelligence, as well as other outstanding features to optimize and more efficiently.

4. Zeniius System

In BitCEO's Whitepaper, we mentioned that there are three basic factors that address the how-to-be-motivated problem of CEOs addressing the needs of business demand:

- ZeniScore (Score of Reputation): The CEO receives ZeniScore from the system whenever the he / she contributes in the system. In addition, ZeniScore also serves as a parameter for the user's influence on system policies in the future.
- Ability Card: Represents abilities of the CEOs in the real world. CEOs can use such Ability Cards to reward each other as well as trade.
- BitCEO (BCEO): The transactional token of the system. Like Ability Cards, CEOs can use BitCEO to reward each other. In addition, users can utilise the token as the payment platform for the Ability Card and other features in the system.

Zeniius includes 4 main part of systems:

- Zeniius Main System: Social network platform. This system handles all social media features, including unlocking features with BitCEO, creating conversations, vieweing of newsfeed and user profiles.
- Zeniius Reward System: Reward processing system for the entire application. This system is responsible for the following functionality: organize the transfer of BitCEO from the Poster to the Solver, coordinate the change of hands of Ability Cards from the Poster to the Solver, award ZeniScore to each user in the system, process the accumulation of ZeniScore for each user which leads to leveling up and BitCEO rewards.
- Zeniius AI System: AI system responsible for matching & solving demands.This system will employ data learning and parsing business demanding into data techniques to carry out matching Ability, represented by Ability Cards, and Business Demand by Poster.
- Zeniius DAO System: Voting system for the entire application. This system will use ZeniScore as the basis for voting.

4.1. Post Activity Process

The following flow-diagram depicts the entire activity process when a user posts a demand on Zeniius:

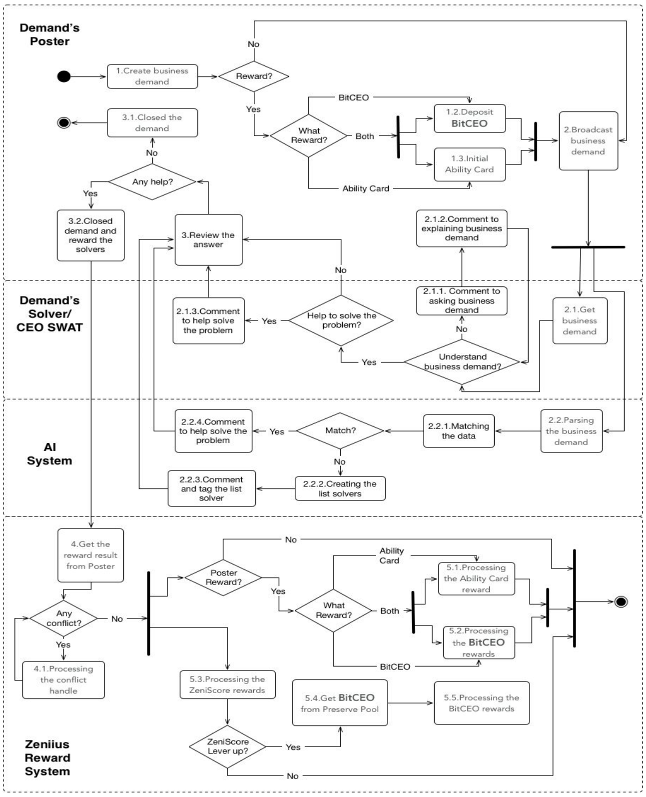

Fig. 3: Activity flow diagram in Zeniius

After the creation of the Business Demand, the Poster has a choice whether to reward the Solver with (i) BitCEO (ii) Ability Card or (iii) Both. To do so:

- (i) BitCEO will be directly deposited into the Business Demand.
- (ii) Ability Cards previously created by the Poster can be added into the Business Demand as rewards.
- (iii) A combination of (i) & (ii).

Upon the creation of Business Demand & attachment of Rewards in the form of BitCEO and/or Ability Cards (if any), the Poster may then publish this information onto Zeniius platform for it would be broadcasted to the application's entire user base. The following two groups would potentially approach the information:

- Solver / CEO SWAT: These are existing users in the system. Upon receiving information on Business Demands, these users could discuss directly with the Poster on the post in order to gain deeper understanding of the Poster's demand. They may then offer solutions which may help the Poster solve the issue. They may also recommend other Users, both internal & external to the system, whom they think may be able to provide support regarding the Poster's demand.
- Zeniius A.I.: An Artificial Intelligent system powered by Zeniius to assist users with solving their business demands. This AI system will first parse information in the Poster's demand into keywords. These keywords, coupled with data previously captured by the system, will help the AI with mapping these demands with any Ability which has been previously added into the system by other users. If the mapping process yields absolute matches, the AI will recommend these users to the Poster by directly commenting on his or her post. If there is no exact match between the keywords generated from the Poster's demand and currently existing abilities in the system, AI would still recommend the Poster with next-to-best matches by leaving a comment on the post.

The two systems of Solver / CEO SWAT and AI would complement each other to assist the Poster with resolving his or her demand. Two scenarios ensue. (1) If the Business Demand remains unsolved after this process, the Poster may close the post. Any BitCEO deposited or Ability Cards attached would be released. (2) If the Business Demand was successfully resolved, the Poster will thank the Solver with bitCEO or Ability Cards. Should there be multiple contributors, the Poster will decide the user who would be rewarded with Ability Card(s) (the main Solver) and those who would be rewarded with BitCEO (the Contributors).

In the second scenario, Zeniius Reward System (ZRW) will coordinate the distribution of the rewards based partly on the Demand Poster's review. In the case of dispute regarding Poster's solution review, the system will process the case as a Pending Dispute to be resolved (discussed later), otherwise the activity flow continues normally. If the Demand Poster have chosen to reward the Solver with Ability Card, ZRW will transfer the specific Ability Card to the Solver. If BitCEO was chosen as Rewards, 20% of the total token amount will be rewarded to the system for hosting the problem-solving environment. Of the remaining 80%, ZRS will calculate the rewards to each Solver / Contributor based on % voted by the Poster.

The Zeniius Reward System will also award ZeniScore for Solvers / Contributors based on the Poster's vote. The Poster would receive ZeniScore from other users as well. This self-configuring system will increase user's ZeniScore. They would then receive BitCEO from the Preserve when their user hits a certain tier.

4.2. Conflict Activity Flow

4.2.1. Business Demand Conflict Activity Flow

In section 4.1, we discuss the issue of dispute resolution when the Demand Poster review solutions. In fact, this is a rare case for CEOs who are highly conscious users. However, in order to minimize errors in future operations, this is a problem we also need to address. Currently, a specialised team known as CEO SWAT will help us to solve issues relating to inter-user disputes. The CEO SWAT team will receive conflict cases, collect evidence, rely on consensus and resolve the disagreements. If the CEO team concludes that the conflict was caused due to dishonest behaviours from the Demand Poster, this CEO will be receive penalty to Zeni Score. Figure 4 shows the activity flow of the dispute resolution process.

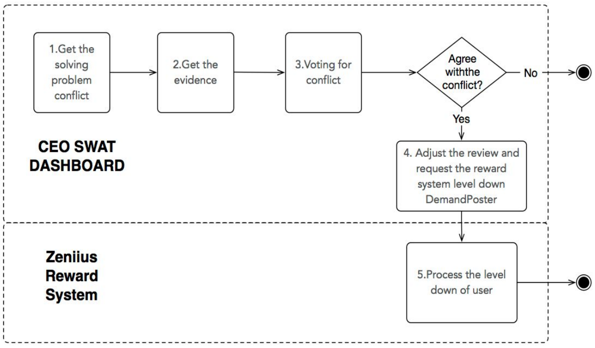

Fig. 4: Conflict Resolution Flowchart

4.2.1. Ability Card Conflict Activity Flow

In Zeniius, in addition to handling disputes regarding the review of results by Demand Posters, there is another dispute type which the system needs to handle. This refers to the potential conflict between the owner of the Ability Card and its Creator, when the time comes for the owner to exercise his or her rights of owning the Ability Card. This dispute resolution process also relies on the collection of evidence and the consent of the CEO SWAT team to process. The CEO violating the rules will be punished with Zeni Score deduction. In the scenrarior where the Creator is the perpetrator and he/she has a BitCEO stake in the Ability Card, this amount of BitCEO will be passed to the Owner as a form of compensation. Figure 5 illustrates the activity flow of this dispute resolution process.

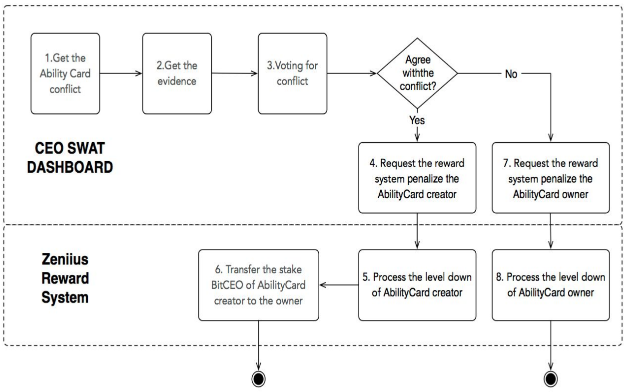

Fig. 5: Ability Card Conflict Resolution Process

4.3. DAO Voting Activity Flow

In the future, Zeniius will share some of the power of system development to users who are great contributions to the Zeniius system. This is an indispensable trend to ensure that the system not only develops based on existing resources but also develops based on the user community. The CEO SWAT team is the starting point of the system and will be developed as a future DAO. Figure 6 depicts the two main activity flows of the DAO formation and voting process in DAO.

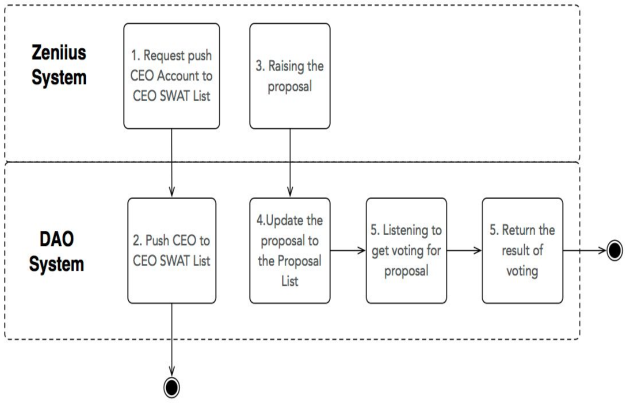

Fig. 6: DAO Activity Flow

5. The Blockchain Implementation

In Zeniius, to ensure system transparency as well as business engagement with CEOs, we need to apply Blockchain, and Smart Contract as a solution. However, to balance transparency and security of user data, data is selected based on absolute necessity to be decentralized and some necessary processes are used to deploy Smart Contracts.

In implementation, a Contract Layer will be built to implement the Smart Contracts in the system. In addition, to increase ease of use for BitCEO by users in the system, we built a Wallet Layer. Figure 7 depicts these two layers applied to the Zeniius system as well as the architecture of the Zeniius system.

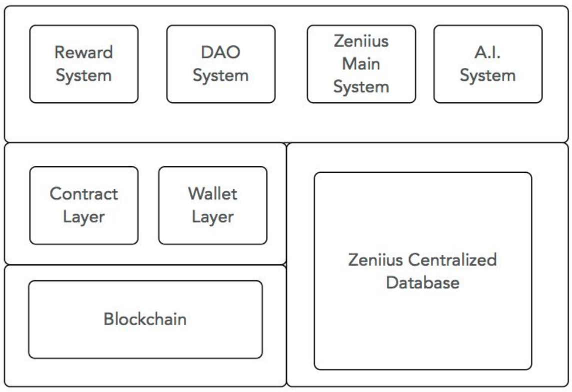

Fig. 7: Blockchain in Zeniius

The Contract Layer will contain all of the system's Smart Contracts. These Smart Contracts are based on the operating flows of the system in Section 4.

- In Section 4.1 - Post Activity Process, Flow 1.2 (Deposit BitCEO) depicts the BitCEO CashFlow Smart Contract, which was built to be responsible for controlling the flow of BitCEO in the system.
- In section 4.1 - Post Activity Process, Flow 1.3 (Initial Ability Card) depicts 4 Smart Contracts as follow:
    - "Logic For Promise" Smart Contract: This is a smart contract that processes the execution of the Ability Card.

- ○ "Verification of Ability Card" Smart Contract: This is a Smart Contract that deals with Ability Card validation - whether the Ability Card is accurately representative of the user in Zeniius.
- ○ "Aggregate of all Ability Cards" Smart Contract: This is a Smart Contract that contains all information about the Ability Cards that have been verified.
- ○ "Account User" Smart Contract: Contains the user information that owns the ability card, as well as the amount of Zeni Score that the user owns.

- In Flow 1 of Section 4.2.1, Figure 4: Business Flow Conflict Activity Flow, we built the CEO SWAT Smart Contract to file dispute information to be pushed to the CEO SWAT team to handle. CEO SWAT Smart Contract is also used in Section 4.2.1. - Ability Card Conflict Activity Flow - to handle conflicts when a user in the system fails to commit to promised endoresed within the Ability Card.
- In section 4.3. DAO Voting Activity Flow, during Flow 2, we build DAO Smart Contract to save Governing Account information as well as Voting Logic.
- In section 4.3. DAO Voting Activity Flow, at Flow 4, we created a Proposal Smart Contract that records the user's proposal information for the system.
- In addition to being able to easily update the Smart Contracts as well as managing Ability Cards' execution flow, we built the Master Smart Contract. All Smart Contracts will be subjected to Master Smart Contract navigation.

Figure 7 shows all Smart Contracts in the Smart Contract Layer

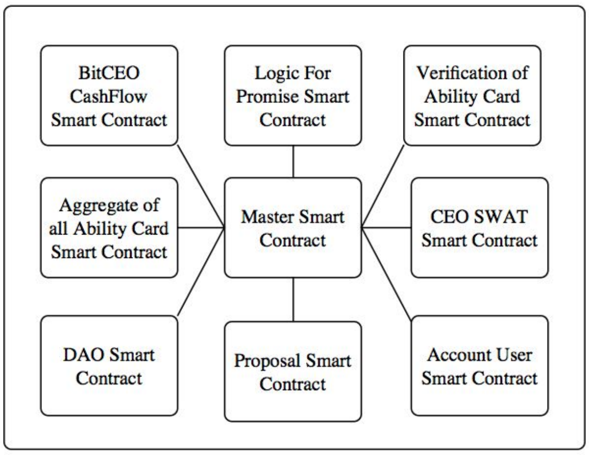

Fig.7: Smart Contracts in Zeniius

We are currently planning to run all Smart Contracts and BitCEO on Ethereum blockchain as it is one of the most popular, flexible, and suitable platforms for adopting Smart Contracts. Following this, depending on future blockchain trends, we will be running tests on other platforms such as Zilliqa, NEO, etc to ensure speed, integration, scalability, etc.

We will be using a lightweight node implementation to be run on mobile devices, web and embedded into Zeniius. A lightweight node also known as Simple Payment Verification (SPV) node does not download the entire blockchain locally to perform transaction validation. It downloads only the headers of the blocks. This avoids having to download the entire blockchain which would take too much time and consume too much space on a mobile device.

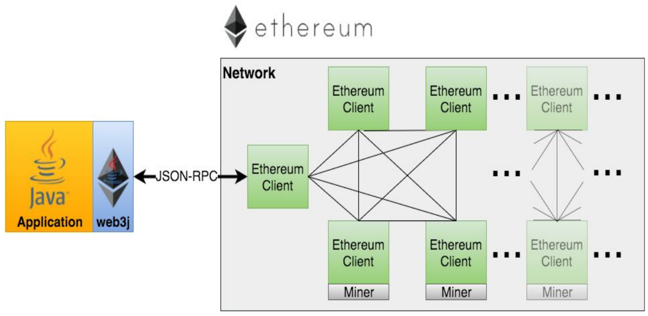

Fig. 9: Web3j Module

We also use Web3j - is a highly modular, reactive, type safe Java and Android library for working with Smart Contracts and integrating with clients (nodes) on the Ethereum network.

We are currently in the formation phase of the project. More information about Blockchain integration & actualization will be updated shortly.

6. The A.I Implementation

6.1. Demand-Ability Matching by A.I

Demand-Ability matching is an interesting problem that can be automated with the use of AI. This will bring about tremendous improvement in the usefulness of a social platform, especially one with that are filled with goal-oriented individuals such as ours. Here we implemented an algorithm that will match between buyers and sellers of real estate properties. We have noticed a high amount of posts in renting properties for setting up warehouses and shops for businesses, and we thinks that an algorithm that could automatically match sellers and buyers is very beneficial to our community. Our system will be able to recognize a post about selling, renting or buying property, analyze the post for relevant phrases that captures the characteristics of that property such as price, address, architecture and surrounding area, etc. that could be of interest to other people.

We defined this problem as sequence labelling, a term that encompasses all tasks where a sequence of input data is matched with that of discrete labels. Well-known examples include speech and handwriting recognition, part-of-speech tagging, named-entity recognition and event detection.

For instance, let us consider the following Vietnamese input: "Bán đất tiện xây phòng trọ, cho thuê, gần nhà máy sữa Vinamilk".

Consider this input at a sequence of Vietnamese, this sequence will be matched with a sequence of labels as follows.

('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('tiện', 'O'), ('xây phòng trọ', 'POTENTIAL'), ('cho thuê', 'POTENTIAL'), ('gần', 'O'), ('nhà máy sữa', 'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME')

where TRANSACTION_TYPE, REAL_ESTATE_TYPE, O, POTENTIAL,SURROUNDING_PLACE,SURROUNDING_NAME are the labels bearing the corresponding semantics.

Here what we aim to recognize is all the criteria of a property that people usually consider when purchasing a real estate. This is a challenging problem because: (i) these criteria is very diverse; (ii) they are expressed in many different way in natural language; and (iii) spelling mistakes is frequently made in posts on a social network. Previous works on linguistic sequence labelling employed statistical model such as Conditional Random Field [2], Hidden Markov Model [3] and Maximum Entropy Model [4] with carefully designed feature engineering that are domain-and-language-specific. Such features are costly to develop and thus these result cannot be easily adapted to new problem. Recent development in the field has yielded state-of-the-art result with neural network models, such a [5],[6],[7]. All of those works combined Bidirectional LSTM with Conditional Random Field, with some additions such as character-level word embedding generated either by a CNN or a Bi-LSTM. These models typically operated on word embedding trained through the use of unsupervised methods such as GloVe[8] or Word2Vec [9] , therefore requires only minor or no feature engineering at all.

In this whitepaper, we present the way that we improve upon the work in [7], adding components to make our model more suitable to the characteristics of the Vietnamese language. Similar to that paper, we first apply an simple 1D CNN-Max Pooling layer to generate character-level representation of a word. Then we combine the pre-trained word embedding with its character-level representation and feed them into another CNN, this time on word level, to capture the features of n-grams. These features are the input to a stack of BiGRU layer. The output layer is a Conditional Random Field to decode the sequence of labels with the highest score given the input sequence. We evaluate our model on our data set of real estate advertisements and demonstrate that the added components to the model certainly improved our model performance on the data set.

6.1.1. Background

a. Convolution and pooling

Convolution [10] is an operation that is widely used in many deep learning models for various research fields such as computer vision and natural language processing, the latter usually involves a 1-D convolution whose structure is depicted in Figure 10.

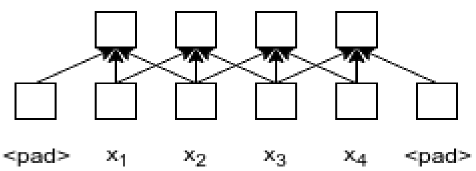

Fig.10: A 1d convolution layer

Let d be the size of each word vector and s be the length of sequence. Let C ∈ Rs x d be the sequence matrix. A convolution operation applies a kernel H ∈ Rd x k to a window of k words to capture the features of k-grams in the sequence. Specifically, a feature of an k-grams from a window of words C[i : i+k; ∗] is generated by:

$$ci = \sigma(\sum(C[i : i+k; *] \odot H) + b) \quad (1)$$

where b ∈ R is the bias term, σ is a non-linear activation function such as tanh or ReLU and ⊙ is the element-wise matrix multiplication. To retain the length of a sequence after convolution is applied, we zero-padded the sequence evenly on both side with a padding size of (k-1)/2. Under such padding scheme, a kernel is convolved with each possible k-grams in the sequence to produce a feature map c = [c1, c2, …, cs] with c ∈ Rs.

Pooling is an operation used to aggregate a group of generated features. A popular function for pooling is to find the maximum value among them. In term of sequence modelling, global pooling is often used, which returns the most distinctive features of a sequence when coupled with the max pooling function.

b. Gated Recurrent Unit

Gated Recurrent Unit (GRU) was introduced by [11]. It is a variant of Recurrent Neural Networks (RNNs) capable of learning long term dependencies while avoiding the problem of vanishing or exploding gradients faced by vanilla RNNs. GRU is proved to be comparable to LSTM [12] while maintaining a smaller set of parameters. See Figure 11 for the graphical illustration of GRU.

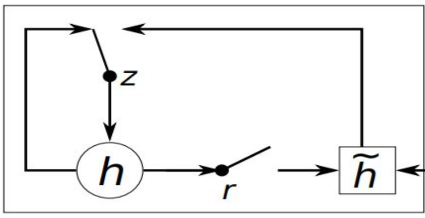

Fig.11: An illustration of a GRU cell [11]

The parameters are updated using the following equations:

$$zt = \sigma g(Wzxt + Uzht\text{-}1 + bz) \quad (2)$$
$$rt = \sigma g(Wrxt + Urht\text{-}1 + br) \quad (3)$$
$$ht = \sigma h(Whxt + Uh(rt \odot ht\text{-}1) + bh) \quad (4)$$
$$ht = (1\text{-}zt) \odot ht\text{-}1 + zt \odot ht \quad (5)$$

where σg and σh are activation functions and σg(x) ∈ [0, 1], which usually are a sigmoid function and hyperbolic tangent respectively, ⊙ denotes the element-wise matrix multiplication, rt is the reset gate, zt is the update gate, xt is the input vector and ht is the output vector.

c. Conditional Random Fields

Let x = [x1, x2, …, xT] be an input sequence and y = [y1, y2, …, yT] be the sequence of corresponding tags. Let D be the set of possible tags, that is yi ∈ D ∀i ∈ [1, T] and Y be the set of all possible label sequences.

A linear-chain conditional random fields (CRF) model [2] is defined as

$$s(y) = \sum_{t=1}^{T} A_{x_t, y_t} + \sum_{t=2}^{T} V_{y_{t-1}, y_t} \quad (6)$$

$$P(y|x) = \frac{e^{s(y)}}{\sum_{\tilde{y} \in Y} e^{s(\tilde{y})}} \quad (7)$$

where A_{x_t, y_t} denotes the score of t-th word having tag yt, V_{y_{t-1}, y_t} denotes the transition score between the previous tag yt-1 and the current tag yt and s(y) denotes the score of the label sequence y. In this model, prediction can be decoded using the O(D^2T) Viterbi algorithm to find a sequence of tags with maximum score given the input sequence.

CRF models have a property of incorporating interactions between consecutive tags into making predictions. This introduced two major advantages. The first one is the guarantee that certain constraints of a tagging scheme (e.g: IOB tagging) are always upheld. The second one is the added constraint means that the model requires less training data. However, it has worse computational complexity compare to independent prediction, that is making prediction at each position based only on the features of that position.

6.1.2. Model Architecture

Figure 12 gives an overview of our model which includes the following components

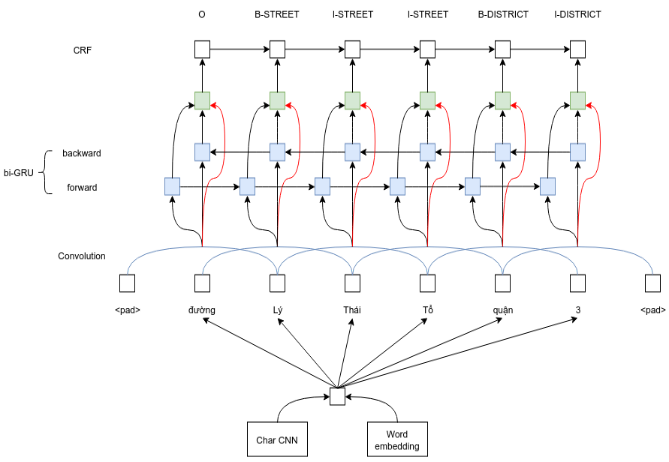

Fig.12: The model architecture where charNN is the CNN network for generating character-level representation (red lines indicate skip connection; blue lines indicate convolution; blue boxes indicate GRU and green boxes indicate CRF)

a. Bidirectional GRU with CRF, convolution and skip connection

Follow the current state-of-the-art model for sequence tagging [5], our model's main component is a bidirectional GRU (bi-GRU) network for feature extraction, followed by a linear-chain CRF as the output layer to determine the tag at each position while considering tags of previous positions. We chose to use GRU in place of LSTM because it has been showed that GRU is comparable to LSTM and even better in certain tasks ([13],[14]) while allowing faster training time and lower sample complexity due to having smaller number of parameters. To account for the fact that syllables in Vietnamese language are written separably, hence each element in a sequence is usually a syllable, we use a convolution layer after the word embedding to capture soft n-gram of syllables, using the output of this layer as input to the bi-GRU component. Using this method, our model is able to correctly capture long and polysyllabic words (e.g "trường đại học") more frequently. To alleviate the problems of vanishing gradients when training over long sequences and to allow our model to have more layers, we added skip connection at each bi-GRU layers. Specifically, each input and output of an bi-GRU layer is concatenated to form the final output of that layer. Given an input sequence $x = [x_1, x_2, \ldots, x_T]$, each bi-GRU layer can be described using following equations:

$$h_t^f = \overrightarrow{GRU}(x_1, x_2, \ldots, x_t) \quad (8)$$

$$h_t^b = \overleftarrow{GRU}(x_t, x_{t+1}, \ldots, x_T) \quad (9)$$

$$o_t = [x_t, h_t^f, h_t^b] \quad (10)$$

For training, the loss function maximizes the probability of getting the correct label sequence from the corresponding input sequence:

$$J = -log(P(y|x)) = -s(y) + \sum_{\tilde{y} \in Y} e^{s(\tilde{y})} \quad (11)$$

where $s(y)$ is defined by (6).

b. Character-level representation

Normally a model for natural language processing can only include a fixed size vocabulary, and replace any unseen words with an out-of-vocabulary (OOV) token. This strategy could reduce performance since the model neglects useful information unknown words could provide. One solution to this problem is to incorporate morphological information of each word into the model. By adding this information, important features such as word case is considered when making predictions. We decided to generate such representation of each word by using a simple convolutional neural network depicted in Figure 13.

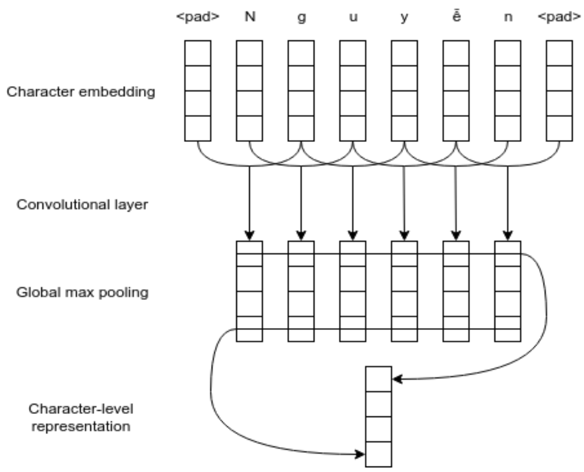

Fig.13: CNN for character-level embedding [7]

This representation is then combined with the corresponding word embedding by concat them together to create the final vector that contains all the information about the word. By using this method, we indirectly embedded information such as word case and group of characters into the model.

6.1.3. Experiment

a. Dataset

Our data comprise 9231 documents collected from mogi.vn, a website for trading real estate in Vietnam. Based on the information usually contained in a real estate advertisement, we came up with 19 different labels representing different properties of a real estate. Table 2 shows information about each label.

Label	**Description**	**Supports**
City	The city where the real estate is located.	1542
District	The district where the real estate is located.	5826
Street	The street where the real estate is located	7351
Ward	The ward where the real estate is located	3512
Area	The area of the real estate	13917
Floor	Number of floors	10956
Room	Number of rooms	10185
Legal	Legal information related to the real estate (e.g certificate of home ownership)	5263
Orientation	The direction which the real estate is facing	1424
Position	Whether the real estate is on a street or a lane.	6908
Potential	Potential usage of the real estate (e.g: to open a shop, a company office)	17706
Price	Price of the real estate	10320
Project	the project the real estate belongs to (e.g Sunrise city)	1876
Type	The type of the real estate (e.g land, apartment, villa, house)	20058
Surrounding places	Places near the real estate (e.g school, hospital, restaurant)	31155
Surrounding characteristics	The characteristics of surrounding area (e.g secured, quiet, populous)	11624
Surrounding names	Names of the places near the real estate (e.g Bách Khoa, Bình Dân)	21563
Transaction type	Type of the transaction (e.g buy, for rent, to rent, for sale)	8526
Normal	Any others information that does not belong to the labels above	194892

Table 2: The labels used in the dataset

For data preprocessing, we replaced all the digits with the character '0'. Any word with less than five occurrences in the dataset is replaced with the generic OOV token. We did not convert the text to a single case because each word case could give useful information (e.g word start with an uppercase letter usually indicate a street name). We first used fastText [15] to train the word embeddings in an unsupervised manner, then using this word embedding as inputs to the model. We decided to use word embeddings with 100 dimensions. We chose to use IOB tagging schemes in our data.

b. Hyperparameter

Word embedding size	100
Character embedding size	30
Character CNN filter size	3
Character CNN number of filters	50
Word CNN filter size	3
Word CNN number of filters	128
GRU hidden size	100
Number of BiGRU layers	2

Table 3: Hyperparameters of our model

c. Training method

To optimize our model, we use the Adam Optimizer [16] with learning rate of 0.001 as recommended. We applied dropout [17] of 0.5 after each layers to avoid overfitting problem. We chose a batch size of 100 and shuffle the data set on each epoch to avoid our model falling into a local minimum. We divided our data set into 3 part: 80% for training, 10% for validation and 10% for testing. We stopped training our model using early stopping, ending the process if the performance of the model on the validation had not improved for 10 consecutive epochs. The final model is the one with the highest performance on the validation set.

d. Result

Table 4 consists of results on three model: the original model introduced in [7] (BiGRU-CharCNN-CRF), the model with residual connection (+ res), the model with word level CNN, (+ WordCNN), and the model with both components.

	F1	Precision	Recall
BiGRU-CharCNN-CRF	85.6	85.3	85.8
BiGRU-CharCNN-CRF	85.8	85.7	85.8

(+ res)			
BiGRU-CharCNN-CRF (+ WordCNN)	86.1	85.4	86.9
BiGRU-CharCNN-CRF (+ WordCNN + res)	86.3	86.1	86.6

Table 4: Experimental results

From the result we can see that adding residual connection increase the precision of the model, probably due to the the fact that shortcut connections allow faster and easier convergence. The WordCNN increases recall significantly because its help the model to capture long phrases with more than 4 words. Figure 9 compares the result between model before and after adding the WordCNN layer. Finally, adding both the WordCNN and residual connection yield the best result.

('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('tiện', 'O'), ('xây phòng trọ', 'POTENTIAL'), ('cho thuê', 'POTENTIAL'), (', gần', 'O'), ('nhà máy sữa', 'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME'), (',', 'O'), ('kumho', 'SURROUNDING_NAME'), (',', 'O'), ('colgate', 'SURROUNDING_NAME'), ('với hơn 35000 công nhân đang làm việc', 'O'), ('ở', 'POTENTIAL'), ('đây , sát', 'O'), ('trường đại học', 'SURROUNDING_PLACE'), ('quốc tế miền đông \n', 'O'), ('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('xây nhà trọ', 'POTENTIAL'), ('Bình Dương', 'CITY'), ('vị trí rất đẹp , đường xá rộng lớn xe hơi đổ cửa , xung quanh dân cư sinh sống rất', 'O'), ('đông', 'SURROUNDING_CHARACTERISTIC'), (',', 'O'), ('buôn bán', 'POTENTIAL'), ('tấp nập', 'SURROUNDING_CHARACTERISTIC'), (', rất thích hợp', 'O'), ('kinh doanh', 'POTENTIAL'), ('buôn bán', 'POTENTIAL'), (',', 'O'), ('xây kiot', 'POTENTIAL'), (',', 'O'), ('quán ăn', 'POTENTIAL'), (', ... \n DT :', 'O'), ('24 mx 30 m = 720 m 2', 'AREA'), (',', 'O'), ('sổ đỏ riêng', 'LEGAL'), ('đã tách 4', 'O'), ('sổ riêng', 'LEGAL'), ('. \n Giá :', 'O'), ('450 triệu / sổ', 'PRICE'), ('. \n', 'O'), ('Đất', 'REAL_ESTATE_TYPE'), ('sổ đỏ', 'LEGAL'), ('- thổ cư 100 % , đường đã trải nhựa \n Vui lòng liên hệ chính chủ : 0903 995 824 - 0902 969 278', 'O')]
('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('tiện', 'O'), ('xây phòng trọ', 'POTENTIAL'), ('cho thuê', 'POTENTIAL'), (', gần', 'O'), ('nhà máy sữa', 'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME'), (',', 'O'), ('kumho', 'SURROUNDING_NAME'), (',', 'O'), ('colgate', 'SURROUNDING_NAME'), ('với hơn 35000 công nhân đang làm việc ở đây , sát', 'O'), ('trường đại học', 'SURROUNDING_PLACE'), ('quốc tế miền đông', 'SURROUNDING_NAME'), ('\n', 'O'), ('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('xây nhà trọ', 'POTENTIAL'), ('Bình Dương', 'CITY'), ('vị trí rất đẹp , đường xá rộng lớn xe hơi đổ cửa , xung quanh dân cư sinh sống rất', 'O'), ('đông', 'SURROUNDING_CHARACTERISTIC'), (',', 'O'), ('buôn bán', 'POTENTIAL'), ('tấp nập', 'SURROUNDING_CHARACTERISTIC'), (', rất thích hợp', 'O'), ('kinh doanh', 'POTENTIAL'), ('buôn bán', 'POTENTIAL'), (',', 'O'), ('xây kiot', 'POTENTIAL'), (',', 'O'), ('quán ăn', 'POTENTIAL'), (', ... \n DT :', 'O'), ('24 mx 30 m = 720 m 2', 'AREA'), (',', 'O'), ('sổ đỏ riêng', 'LEGAL'), ('đã tách 4 sổ riêng . \n Giá :', 'O'), ('450 triệu / sổ', 'PRICE'), ('. \n', 'O'), ('Đất', 'REAL_ESTATE_TYPE'), ('sổ đỏ', 'LEGAL'), ('- thổ cư 100 % , đường đã trải nhựa \n Vui lòng liên hệ chính chủ : 0903 995 824 - 0902 969 278', 'O')

Table. 5: Result on the model without WordCNN (above) and with WordCNN (below). Notice the long phrase in red is captured by the latter model, but is ignored by the former.

6.2. Use the A.I Algorithm in Scoring

Many technological companies have employed A.I. Algorithm, in scoring users in its ecosystem. We have studied in-depth researches about this topic by other innovative teams in similar projects and discovered valuable learnings. Specifically, DropDeck and its sophisticated design and innovation in A.I. Scoring have been great inspirations for us to developed our own. Despite the project being unsuccessful, the values in terms of concept in the area of rating users brought into light by Dropdeck and its team is undeniable. As a result, we have learned from the legacy of Dropdeck and incorporate them into our more advanced Scoring system. Beyond that, we received direct technical advices from some of Dropdeck's founding members to better develop our project and avoid similar pitfalls.

Our final goal is to score, rank, and recommends CEOs in terms of overall reliability and competence. In order to approach the true measure of a CEO (ZeniScore), our principle is to incorporate all possible sources of data, including evaluation data from people on Zeniius. A user's evaluation usually incorporates all the available private information about the evaluated CEO, which is otherwise inaccessible to any kind of data partners or crawlers. However, not everyone's evaluation can be trusted.

Existing platforms operate on a one-user, one-vote principle. This creates an environment where sybil attacks and the service providers such as Quora, Reddit, Facebook can manipulate rankings, etc. must proactively identify and block abusers. On the other hand, Steem operates on the basis of one-token, one-vote. Under these models, individuals who have contributed the most to the platform, as measured by their account balance, instead of individuals who have the most perceived credibility on the matter at hand, have the most influence.

We have to build a highly dynamic relational graph where each entity can be a company or a person, and each entity's change in score would have a ripple effect on all associated entities.

We hand-craft expert models with all feature weights determined by expert opinions collected via surveys, and build reinforcement learning models to be trained from continuous labeled data. Then we combine the scores from these models to obtain a final score for each user. The more data available over time for training, the more scores from the trained models will contribute to the final scores.

Ground truth data is generated from crowd evaluators (any user on the platform) and expert evaluators (individuals appointed by Zeniius) through a feature called "Trust" on our website, designed to collect labeled data. For generating this data, a "trustor" can assign a score that indicates his level of trust to any "trustee" on the platform. Surveys are also designed to actively collect labeled data, where evaluators are shown pairs of users who were known to them in their network, and are asked to identify the trustworthier one in each pair. Multiple users to reduce bias rank each pair of users. The models are trained for these user ratings and rankings in the ground truth training set, using reinforcement learning methods to generate a weight associated with each feature.

After months of research and validation since February 2017, we have selected the following components for calculating the Zeni score, list them in order of weight (importance) and will implement them one by one (not all at once).

A user's Zeni score is based on the following components:

- Crowd evaluation: how people rate or vote for the trustworthiness on Zeniius
- On-site behavior: signs of spam, flip-flop, etc. (negative) or moderation, consistency, etc. (positive)
- Crowd interaction: such as how strongly you spread and attract information
- Credentials: trustworthiness of co-investors/co-founders, potential of the investment portfolio/companies founded
- Associated entities: trustworthiness of the people, potential of the companies related to the user
- Ratings/rankings: the reputation on a set of reliable websites (Quora, MatterMark, CB Insights, etc.)
- Social activity: your behavior, interaction on your social networks (Linkedin, Facebook, Twitter, etc.)
- Linguistics: the use of the language and the people who interact with the user
- Digital footprints: the number of your identities on the web
- Identity consistency: consistency of your information across those identities

Since the ZeniScore takes into account all user behavior and interaction with other users, it represents a reputation system to protect Zeniius users, restrict spammers/criminals, and encourage healthy activity. This reputation system will be present in every aspect of the user experience, from initial sign-ups to evaluation to comments, allowing users to easily identify users who may have a history of misbehavior or spamming. Newly joined users will have to obviously improve their ZeniScores to build trust in the community. This method could reduce “Distrust" reports handled by oracles. Likewise, the reputation system will enhance the competitive experience by encouraging users to behave with integrity, be active and build their reputation. The platform’s oracle selection engine gives higher priority to users with higher ZeniScore. Each individual user will also have the ability to set the minimum ZeniScore threshold for the deck’s or other information's visibility. Having a higher ZeniScore thus allows users to see more private content on the platform. A global ranking system will be viewable on Zeniius to encourage competition. Ranks will be rewarded with Zeniius subscription for premium features or other promotions.

The challenge we faced is deriving algorithms for scoring that most people can consider to be a fair assessment of the subjective qualities being scored. Moreover, there's basis to judge whether a definite score is fair or not. Common questions might be “Why is this company 70? 70 to what? Which one is 100?” Therefore, the definite Trust and ZeniScores will be hidden and replaced with percentile scores. Now when you see a company with a score of 70, any one can know that company has more potential than 70% of the other companies on Zeniius. It requires further research and collaboration with reputable partners to establish scoring standards before definite Trust and ZeniScores make sense and are public.

6.3. ZeniScore with A.I

While trustworthiness is a broad and subjective concept, we can quantitatively describe it in terms of observable behaviors. In particular, we look for signals that exhibit signs of cheat, spam, doubt, uncertainty, hesitation, rhetoric, contrivance, impulsivity, thoughtlessness, contradiction, etc. which correspond to a low ZeniScore and, on the opposite, signs of decisiveness, conciseness, consistency, etc. which correspond to a high ZeniScore. In our definition, one’s ZeniScore is a predictive score that measures one's probability to make or back claims that would be true or accurate above a certain threshold.

Personalized recommendations for investors, entrepreneurs, and service agencies would be derived from users' ZeniScores generated using machine learning models trained with 1000+ features. These features also incorporate factors that provide a proxy for real world or offline trustworthiness such as the ability to attract information (and resources) from other people. For example, the most trustworthy venture capitalists usually get connected to the best deals by entrepreneurs and other investors who trust them.

So, for a person with high ZeniScore:

- Information propagated by that person has a high accuracy or probability of being true of accurate
- It’s more likely for people to share information with that person

A user A's ZeniScore (ZS) includes the following components:

- Absolute TS (AZS - indicates user's objective level of trustworthiness), incorporates:
    - Permanent TS (based on personal information, behaviors that indicate long lasting traits)
    - Dynamic TS (based on behaviors that indicate temporary traits within 90 days)
    - Scalable TS (based on variables that scale with the number of selected “Dow Jones decks”)
    - Attractive TS (based on user A's ability to attract information)
- Relative TS (RTS - indicates your subjective level of trust placed in user A) incorporates:
    - Conversational TS (based on conversations between you and user A)
    - Propagational TS (based on your acts of propagating user A's information or sending information to user A)

User A also has many Topical TSs (TTSs) each of which indicates user A's trustworthiness or expertise level on one of many predefined topics (e.g. industries, funding stages, locations).

6.3.1. Topical ZeniScores

To generate one's TTS on a topic, we must predict one's expertise level on that topic, based on one’s biography texts from Zeniius, Twitter, Quora, and tuple (Company name, Job title) from Linkedin and AngelList. We use multi-class classification, with each class being a bucket on the expertise spectrum. We use learning models such as logistic regression, random forests, and gradient boosted decision trees; features such as n-grams, list of named entities, cosine similarity between topic name and biography text. Training data sources include hand labeled data and label propagation. After generating a TTS for every topic for each user, we can further refine them with the following insight: since trust in expertise is transitive (A→B, B→C, then A→C), it propagates through the network. Therefore, we create a graph where each node is a user and its TTSs, each edge is a user-trusts-user relationship weighted with the value which the trustor has assigned to the trustee. Then we apply graph algorithms like PageRank to update all TTSs at each node.

6.3.2. Pipeline

When a user registers on Zeniius.com, he associates his identities on different social networks with his Zeniius profile. For Twitter, public data is collected via the Mention Stream, and data for other social networks is collected via REST APIs on the user's behalf, based on the granted permissions. All collected data is parsed and normalized to protocol buffers that encode user interactions, graph, and profile information. Data is continuously collected from interactions in a trailing window of 90 days using Django, Celery, Redis, etc. The collected data is written out to a distributed file system. The batch processing pipeline derives features for each user, normalized against the global population. Feature weights from the reinforcement models built using ground truth data are then applied to generate scores.

6.3.3. A.I. technologies

A.I. technologies are used in order to increase the degree of convenience, speed, and accuracy of our scoring engine and recommendation system, which will be used to optimize the decision making process in venture investment. Our technology of choice is deep reinforcement learning. We use ElasticSearch as a search engine for indexing, search, and quick analytics. Languages and frameworks include Python, Scikit-learn, TensorFlow, Theano, and Keras. Learning models include recurrent, convolutional neural networks for text classification, recurrent Neural Networks for time series analysis, and deep Boltzmann Machines for recommendation system.

6.3.4. Validation

In order to evaluate success, we have to validate that users with higher ZeniScores are able to predict good investments more accurately and attract more information in a network. Then, we compare the performance of the score against other scoring/ranking systems and also analyze the dynamic nature of the score. We examine different topical domains and find that highly-trustworthy users are correctly identified within these domains through their high scores. Below, we examine the Trust Score from 5 different aspects to illustrate its correctness and usefulness.

a. Accuracy of judgment

Each user can evaluate favorably ("Approve") or unfavorably ("Deny") a claim. For each deck, we divide users who have evaluated its claims into those with high ZeniScores (for example, above 80%) or "High group" and those with low ones or "Low group." We aggregate evaluations (weighted by each evaluator's ZeniScore) by High group to obtain a High score and those by Low group to obtain a Low score. In parallel, we allow entrepreneurs to report their verified achievements (investment, admission to incubators or accelerators, won prizes, etc.) and build a learning model to score them (in the same way as generating TTS) and generate a Success score for each deck.

After one year, we run an experiment: for each deck, we compare its High score (HS) and its Low score (LS) to its Success score (SS). A higher similarity between the HS and the SS than that between LS and SS would indicate a higher accuracy of judgment by users with high ZeniScores.

b. Attraction of Information

To validate the effectiveness of the ZeniScore, we will run an experiment to measure the attraction of information with respect to the user scores. Users with varied ZeniScores will be targeted with perks, which are premium data that can only be claimed under a condition - the users will be required to send the perks to at least one other user, so that both can claim the perks. A higher number of perks received indicates a greater attraction of information.

c. Comparisons with Other Systems

Real-World Rankings: We can compare the Topical ZeniScore with other rankings that indicate trustworthiness, such as Quora's topical rankings for top viewed writers on venture capital. To measure the ranking quality of TTS, we adopt the normalized Discounted Cumulative Gain (nDCG) metric, defined in Eq. 4. The Discounted Cumulative Gain upto position p (DCGp) is calculated as in Eq.12, and the ideal DCG for p is denoted by IDCGp.

nDCGp = DCGpIDCGp (12)
DCGp = i=1p(2reli− 1)log2(i+1) (13)

We calculate the IDCGp by using the Quora's topical rankings as the ideal ordering of users. We set the relevance rel of a person as p/rankideal, where the rankideal is his/her position in the ideal ranking. With this setting, a high nDCG, for example p = 10 (higher than 0.7), would demonstrate that the TTS is able to capture online reputation to a high degree for these examples.

d. Trustworthiness by topic

Since trust is typically contextual, we explore the effectiveness of the ZeniScore across different topical domains. Users in topical domains are ranked by their TTSs within their respective domains. After one year, we will take note of rankings in a few selected topics such as Fintech, Healthcare, and Robotics. For a topic such as robotics, we should see, for example, that Elon Musk (founder of Tesla) has a higher TTS than Evan Spiegel (founder of Snapchat) in order to validate that TTS can correctly identify the reputed figures in a variety of domains.

e. Accuracy of recommendation rankings

Every day, for each user, we recommend 5 ranked trustworthy people, and record which of them would be added in which order after 24 hours. In the order of the ranking, at every user we compute the precision (the number of added users up to and including that user, divided by the total number of users up to that user), and then take the average (AP score). Then we calculate the Mean Average Precision (MAP) across all AP scores for all queries. We check whether our MAP is higher than 0.5.

7. Conclusion

Currently, Zeniius is in its formation stages and nearing the completion of Phase 1: creating a social networking platform for CEO to network and do business. It will be time consuming, but necessary, to implement Phase 1. We will collect feedback from users to further refine our model. Our next phase will be the integration of A.I. and Blockchain technology in Zeniius which will make it one of the most useful, optimal tools for CEOs.

References

[1]. Divakar Goswami. 2016. "Deloitte - B2B payments market is a significant untapped opportunity". https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-b2b-payments-untapped-opportunity-infographic.pdf

[2]. J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," 2001.

[3]. J. Kupiec, "Robust part-of-speech tagging using a hidden Markov model," Comput. Speech Lang., vol. 6, no. 3, pp. 225–242, Jul. 1992.

[4]. A. Ratnaparkhi, "A Maximum Entropy Model for Part-Of-Speech Tagging," Dec. 2002.

[5]. Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv [cs.CL], 09-Aug-2015.

[6]. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition," arXiv [cs.CL], 04-Mar-2016.

[7]. X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," arXiv [cs.LG], 04-Mar-2016.

[8]. J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, vol. 14, pp. 1532–1543.

[9]. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," arXiv [cs.CL], 16-Oct-2013.

[10]. Y. L. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[11]. K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," arXiv [cs.CL], 03-Jun-2014.

[12]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[13]. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv [cs.NE], 11-Dec-2014.

[14]. W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," arXiv [cs.CL], 07-Feb-2017.

[15]. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," arXiv [cs.CL], 15-Jul-2016.

[16]. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv [cs.LG], 22-Dec-2014.

[17]. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv [cs.NE], 03-Jul-2012.