The NFO Investment Assistant Based on Aided Labeling and Orchestration of Multiple Statistical and Deep Learning Information Extraction Models

Ivan Perić*, Radana Perić**, Doma Ghale***, Miroslav Kondić*, Stefan Anđelić*

* Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia

** Faculty of Business Economics, University of East Sarajevo, Bijeljina, Bosnia and Herzegovina

*** University of New York, New York (NY), United States of America

ivanperic@uns.ac.rs*, radanaperic94@gmail.com**, domaghale.16@gmail.com***, kondicm@uns.ac.rs*, stefan.andjelic@uns.ac.rs*

Abstract: **A new fund offer (NFO) is the first subscription offering for any new fund offered by an investment company. A new fund offer occurs when a fund is launched, allowing the firm to raise capital for purchasing securities. Investors can research new launches of funds either by monitoring various investment companies' press releases or by checking NFO-related news aggregate sites. In this paper, we propose an automated digital twin model for NFO detection by using the NLP/NLU models combined with scoring metrics to optimize the model performance. Additionally, we propose an approach to use these models without any labeled data, while creating a model evolution pipeline through investor involvement and feedback, enabling the digital twin to evolve through time and improve NFO alerting quality.**

I. INTRODUCTION

A new fund offer (NFO) refers to the initial sale of fund shares issued by an investment company to investors. Similar to an IPO in the stock market, NFOs are intended to raise capital for the fund and attract investors. Even though NFOs are marketed, they are done less aggressively so than IPOs, and target certain select groups of investors. As a result, new fund issues may be less noticeable to individual investors than IPOs. Investors looking to research new fund launches can monitor the press releases of various investment companies as well as news outlets dedicated to aggregating the latest fund news. New fund offers often have the potential for significant gains after beginning to trade publicly.

Often, new fund offers are not widely publicized making them challenging to identify. Companies must register a new fund offering with the Securities and Exchange Commission (SEC) offering one method of tracking. Investors seeking information on new fund offers prior to their launch date may also receive alerts from their brokerage firm. But for attractive offers this moment can be too late, since a lot of other investors will monitor SEC offerings as well.

However, monitoring multiple news sources and investment company publications, before NFO is registered with SEC and publicized, can be very time consuming process, where the introduction of the digital twin would be ideal. Deep learning models focused on natural language processing in combination with automated data acquisition pipelines can automate this process and enable the investor to detect new fund launches in the initial stages to secure potential gains in the future.

From the technical standpoint, creation of a digital twin to represent an actual investor requires news articles acquisition from multiple data sources (news portals, investment companies websites, RSS feeds and similar), but the biggest challenge is the machine comprehension of those unstructured pieces of textural information, more formally in the Natural Language Understanding (NLU) component. This component needs to be powerful enough to understand news articles, detect whether they are related to NFOs or not, and then extract most important information about the fund itself (fund name, launch date, country, capital, etc).

Usually, systems like these require complex modeling to support good NLU capabilities, but that also requires a significant amount of labeled data to support the training of these models.

First goal of this paper is to propose an approach to modeling both knowledge and reasoning process of the physical investor and build its digital representation through Digital Twin concepts, in order to detect NFOs from various unstructured data sources and make an investment recommendation in later stages of the process. The system should be able to detect required fund information without any labeled data, through the usage of pre-trained models, with an addition of intelligent scoring techniques to maximize the performance and to introduce the necessary bias into the system.

The second goal of this paper is to propose an approach to the evolution of the investor's digital representation through time. While the first goal of this paper focuses on the efficient NFO detection, it would be good to enable the digital representation of the investor to evolve through extensive usage and external validation from physical investors. This feature will enable the Digital Twin to become more efficient, and closer in terms of efficiency to its physical counterpart.

II. RELATED WORK

From the domain perspective, the problem we are addressing in this paper demonstrates a novel approach and we didn't manage to find any relevant research similar to this topic. It appears NFO detection from unstructured data sources has been done only manually so far, with no introduction of Digital Twin concepts. Investors are doing this part of the process by hand, or by monitoring NFO events published on Securities and Exchange Commission (SEC) filings [1]. However, monitoring NFO events on data sources like SEC represents a much simpler problem than the one presented in this paper since those events are structured and they don't require any Natural Language Understanding capabilities.

From the technical perspective, approach presented in this paper uses a few well-known NLP/NLU methodologies, but it also introduces some novel components we didn't manage to find in other researches. A few papers with relevant concepts are listed below.

A. *Smart question answering using vectorization approach and statistical scoring*

This paper [2], published in 2021, presents an approach for information extraction based on normalized term weighting-TF-IDF with cosine similarity to detect text passages with potential answers and then they use BM25 ranking function to extract final answer. The system achieves an overall precision of 93.2% and recall of 84.3%, with F1 measure of 88.5% and accuracy of 80%.

While the system achieves decent performance measures, NFO detection would require better NLU capabilities where approaches based on TF-IDF won't be good enough because the answer needs to be precise. Paper uses sentence-as-an-answer approach, while our problem requires on a word-as-an-answer level approach. Moreover, words creating an answer could appear in various contexts, which increases the problem complexity. Word representation itself needs to be more semantically rich in order to achieve this. That is why the usage of Language models is required, instead of TF-IDF. Authors of this paper use cosine similarity as a measure on how relevant each text passage is, which we are going to use as well, but in a different setup.

B. *BERT and R-NET question answering models*

Since the introduction of transformer models in 2017 in a paper "Attention is all you need" [3], transformer models have taken over the field of NLP and improved majority of state-of-the-art results for multiple problems, including Question Answering on SQuAD dataset [4], Named Entity Recognition and Information Extraction. First deep contextualized language models emerged, including Google BERT, published by Google AI Language [5]. These language models can be fine-tuned, biased towards other domains and used for many different purposes, including question answering [6]. All SOTA results in this field of NLP are achieved through transformer models, with the F1 score of 77.96% that can be compared to the human performance on the text comprehension.

Microsoft published R-NET model [7] for question answering in 2017 and it achieved SOTA results before transformer models came into picture. They use pointer networks to locate positions of answers from the text passages.

Both BERT and R-NET models achieve high precision measure, which is important in information extraction problems, and we will leverage that in this paper. Our approach uses an ensemble of these models, including BERT and R-NET to achieve maximum accuracy of extractions, which is a novel approach we didn't find anywhere else at the moment of writing this paper.

III. METHODOLOGY

The main goal of this paper is to create a digital representation of the investor's reasoning process and knowledge in order to detect NFOs by using artificial intelligence and NLP concepts. To achieve this, multiple objectives need to be satisfied, including:

- Creating strong NLU capability for investor's digital counterpart - Digital Twin, to understand investment company publications and news articles in order to detect relevant pieces of information. This is challenging since this information is completely unstructured,
- Extract crucial information about the fund from the detected publication (fund name, publish date, investment company, fund type, currency, country),
- Incorporate the knowledge of the physical investor into the NLU model (include bias towards the domain).

The main challenge is to achieve this without any labeled information for NLU model training. Pre-trained deep learning models for question answering are used for this purpose. They can be used for information extraction purposes and biased towards a particular domain through different specifications of input questions.

Moreover, we are dividing the system into two main submodules – *module for recall maximization* and *module for precision maximization*. By doing this, we are aiming to get best possible extractions for all pieces of information required by the NFO.

A. Extraction ensemble for recall maximization

In order to model the digital counterpart of the investor and create its digital twin, text comprehension capabilities need to be introduced into the system. Investor (human) reads a set of publications and news articles, understands them, detects pieces of information that create an NFO event (fund name, publish date, investment company, fund type, currency, country) and then reacts on the event. In order to give the machine the similar ability, state-of-the-art language models need to be used in order to achieve human-like performance.

To address the problem of machine comprehension without any labeled data, we are using question answering models. Those models are trained on SQuAD dataset [4] and can generalize over different domains.

In order to enable these models to perform information extraction (IE), bias needs to be introduced in a form of configuration questions. Each piece of information that needs to be extracted can be represented with a set of questions, where the answer should be the piece of information targeted for extraction. The more questions created for the same piece of information increases the chance of that piece of information being extracted by some of the models, since input variations can affect output quality, as mentioned in [8]. The main hypothesis we are going to exploit in the next chapter (scoring metrics) is that all those varied input questions should result in similar outputs from all used models since they are targeting the same piece of information. This hypothesis can be used to increase the precision measure of the whole system.

Pre-trained models used in the information extraction ensemble are BERT [5], R-NET [7] and pre-trained NER model. We focus on precision and recall measures of each and one of these models in order to get the optimal performance of the whole system.

a. BERT model

Early models built for SQuAD 2.0 substantially underperformed human-level performance until the release of BERT by Devlin [5]. Now, as of June 2022,

majority of the top twenty submissions on the SQuAD 2.0 leaderboard leverage BERT in some capacity, and, by relying heavily on BERT, the top submissions have almost achieved human-level performance on SQuAD 2.0. Thus, BERT has become foundational to state-of-the-art machine reading comprehension systems. [8]

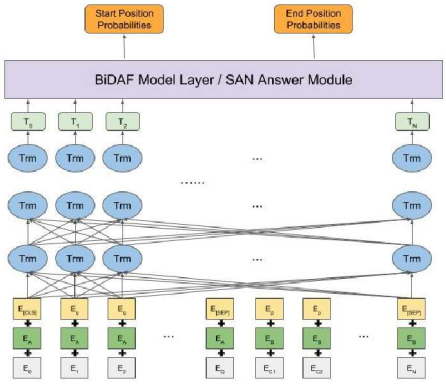

Figure 1 BERT Question Answering model on SQuAD dataset *[9]*

BERT model uses multiple embedding layers to model input sequences of words (position embedding, segment embedding and wordpiece embedding), multi-layer bidirectional attention blocks (transformer blocks) and the model specific layer for answer pointing that gives the start and the end position of the answer for specific input question.

While evaluating the performance of information extraction by using the BERT model, we have observed that when the model returned the answer for an input question, it was correct in almost all scenarios, which means that BERT model gave a high-precision performance. However, in some cases it didn't return any candidates for the answer even if they are present, which signals low recall. This high-precision and low-recall is presented in the next figure.

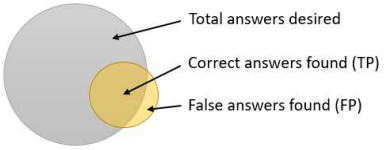

Figure 2 High-precision and low-recall performance of BERT model

Previous diagram shows that BERT models yields an output only in situations when it's mostly sure of its validity. This behavior will be exploited in the scoring module.

b. R-NET model

R-NET is an end-to-end neural networks model for reading comprehension style question answering which aims to answer questions from a given passage. The architecture matches the question and passage, then proceeds with gated attention-based recurrent networks to obtain a question-aware passage representation. Then a self-matching attention mechanism refines the representation by matching the passage against itself, which effectively encodes information from the whole passage. Finally, pointer networks are used to locate the position of answers from the passages. [7]

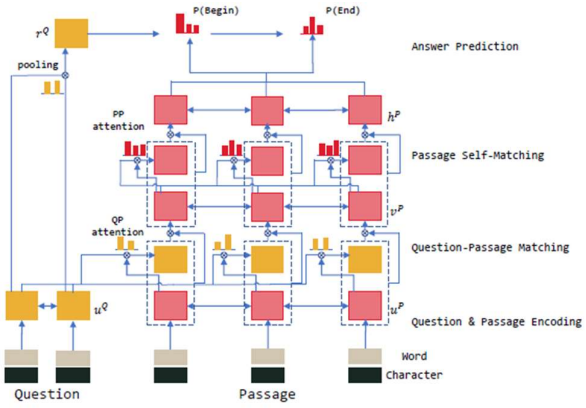

Figure 3 R-NET model architecture

The question and passage are processed by a bi-directional recurrent network (BiRNN) separately. Then, question and passage are matched with gated attention-based recurrent networks, obtaining question-aware representation for the passage. On top of that, self-matching attention is applied to aggregate evidence from the whole passage and refine the passage representation, which is then fed into the output layer to predict the interval which contains the answer for the question, similar to the BERT model.

While evaluating the performance of this model, we observed it achieves similar results as BERT model, with a slightly better recall. R-NET model managed to return an answer in a few situations when BERT didn't manage to do so, while keeping the precision measure very high.

c. NER model

Previously mentioned models returned results only in situations when they were sure about answers validity, while many scenarios were processed without any extractions for NFO information. While some pieces of information were obvious (fund launch date and investment company), those were in complex contexts and models weren't able to extract them. Since we know that the investment company launching the fund is a named entity, we introduced NER model for a few specific fields of NFO (company launching the fund, fund launch date, fund currency and the country the fund will be launched in). These fields can be generalized by pre-trained NER models.

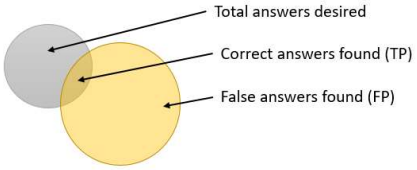

Figure 4 Low-precision and high-recall performance of NER model

Even though NER model gives a low-precision result, its high-recall feature can be exploited in order to maximize end-to-end system performance. NER models will return a large amount of false positive answers, but those answers can be ranked in the scoring module and discarded if needed. However, whenever question answering models fail to give high-precision results, NER output can be used to suggest the low probable answer to the user for review. Investor can get all possible candidates

for each and one of the fields, review and correct the output. That information will be used for model evolutions in the future.

B. Scoring metrics for precision maximization

Previous section introduced an ensemble of information extraction models based on pre-trained question answering models biased with a set of configuration questions for every piece of information targeted for extraction. Each of these models can return an answer for each of these questions, and since there are multiple questions for every NFO piece of information, there might be multiple candidates for each and one of them.

Each piece of information that needs to be extracted is represented with a set of questions, where the answer should be the piece of information targeted for extraction. The main hypothesis we are exploiting is that all those varied input questions should result in similar outputs from all used models since they are targeting the same piece of information. This means that it will be enough to find the most dominantly repeated answer from the extraction ensemble since that should be the answer with the highest likelihood of being a correct piece of information extracted. Multiple scoring metrics are used in this module, focusing on different similarity aspects.

a. Cosine similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis. [10]

$$\cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The traditional distance measures do not work well for sparse numeric data. E.g., two term-frequency vectors may have many 0 values in common, meaning that the corresponding documents do not share many words, but this does not make them similar. Cosine measure focuses on the words that the two documents do have in common, and the occurrence frequency of such words. In other words, it is a measure for numeric data that ignores zero-matches.

This is very convenient for text vectors and will give a good similarity indicator for multiple answer candidates returned from multiple models and varied input questions targeting the same piece of information for extraction.

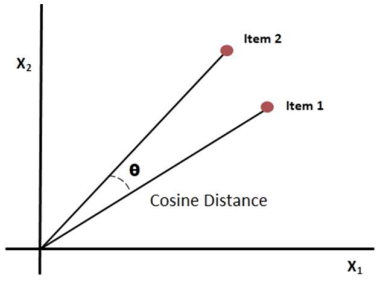

Figure 5 Cosine similarity/distance

b. Jaccard similarity

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used in understanding the similarities between sample sets. The measurement emphasizes similarity between finite sample sets, and is formally defined as the size of the intersection divided by the size of the union of the sample sets. [11]

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Since Jaccard similarity is calculated as an intersection over union, it will penalize extraction candidates that are outliers in terms of length. This is a very desirable behavior since some of the extraction models tend to give longer answers than others, with a few excess words before and/or after the ground truth extraction. By using the Jaccard index, longer answers get a penalized score, while shorter answers get a boosted score.

c. Combining similarity measures

The use of multiple similarity measures enables the scoring module to focus on different aspects of text similarity in order to maximize the likelihood of the correct extraction. As mentioned in the previous section, models from the extraction ensemble focus on different performance measures.

Model	Precision	Recall
BERT	high	low
R-NET	high	low
NER	low	high

Table 1 Summary of model performance measures

The main idea of the scoring module is to maximize the precision measure of the whole system. One additional feature added to the scoring module is the model importance coefficient. As shown in the Table 1, BERT and R-NET models already have a high-precision achieved, which means that if they return any answer, it's very likely that answer is correct. That's why these models have a high importance coefficients. NER model has a low precision measure initially, but it's used to get a set of candidates based only on named entities, in situations where the system wants to inform the investor about potential extractions in aided-labeling module. These candidates are not extremely likely to be correct extraction, but are still potential extractions if human investor reviews and confirms them. That's why NER model has a low importance coefficient in the scoring module.

In our approach, we set BERT and R-NET candidates an initial importance score of 1.0, while NER model was initialized with the importance score of 0.5, meaning these candidates are 50% less likely to be correct extractions since NER model in general gives low precision. Giving model importance coefficients by measuring precision and recall of all models and using both low-precision and high-precision models is a novel approach, at the moment of writing this paper.

After getting multiple candidates for each of the NFO pieces of information, e.g. Fund Investment Company,

scoring module ranks all candidates for investment company names by doing the following:

- boost candidates that are similar and returned multiple times by multiple models (higher likelihood of being a correct extraction)
- penalize candidates with excess information and further boost similar shorter candidates

$$score = modelWeight * \frac{cosine + Jaccard}{2}$$

Each extraction candidate is scored and the rank list is created for each piece of information targeted for extraction. TOP 1 candidate is automatically selected as an extraction result. However, human investor gets to review the NFO event and everything its digital twin managed to understand and extract. For example, if the system detected multiple potential investment companies as a fund launcher, one of them will have the top score, while the other investment companies will remain in the list of candidates. Investor can then change the final result, which will be counted as a feedback to its digital twin, from which its digital counterpart can learn and evolve though time, in order to give more precise extractions in the future.

C. *Digital Twin model evolution through time*

In order to give a digital twin an ability to evolve through time, real investor can give feedback to its digital counterpart and correct false extractions, as mentioned in the previous section.

Given feedback is stored in the knowledge base, which can be used as a dataset for a supervised model training, specialized in NFO event extraction. All previous models were unsupervised from the usage perspective, and this is the first dive-in into a supervised area. Supervised approach wasn't possible because the lack of labeled data. However, after the investor starts using the system and correcting its digital counterpart, it will start creating a dataset of correct extractions. Those reviewed NFO detections and corrections are then being used in a supervised "sequence to sequence" model for information extraction, as shown in the Figure 6.

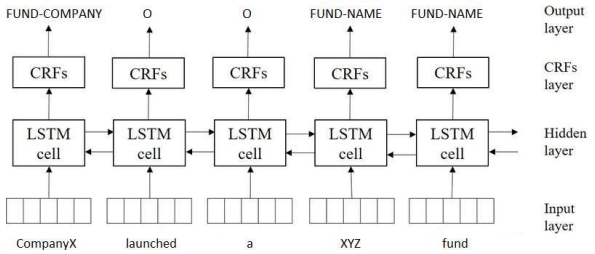

Figure 6 Supervised "sequence to sequence" model for information extraction based on reviewed and corrected extractions

This supervised model can later be added to the extraction ensemble, and also to the scoring module with the highest model importance coefficient since it will have the highest precision score and it will become the model specialized for this problem.

IV. PROPOSED ARCHITECTURE

In order to build a digital twin of the investor, all steps in the investors reasoning process need to be emulated and modeled through data acquisition and NLP in this scenario. Next figure presents an end-to-end modeling and flow diagram.

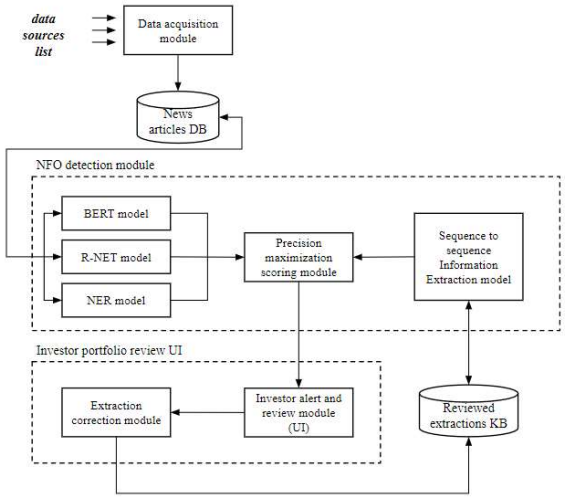

Figure 7 Investor digital twin solution architecture

The process starts with identifying all relevant data sources that investor usually opens and reads when looking for NFO announcement news. Those data sources (news portals, investment company websites, etc.) are an input to the digital twin, and the rest is modeled and automated.

Data acquisition module emulates a proactive nature of the investor. It periodically goes over the list of data sources, opens them and looks for new news articles. If anything relevant is detected, those articles are scraped and placed inside the systems knowledge base, and will be picked up by the rest of the system afterwards.

NFO detection module is triggered by data acquisition module if relevant news articles are detected. Its main purpose is to model the news article understanding process (natural language understanding) that investor does when starts reading a news article. This module uses an ensemble of deep learning and statistical models in order to extract information relevant to NFO from any news article. All extracted candidates for pieces of information of the NFO are scored by the scoring component of this module, in order to maximize the precision measure, as described in 3.B and 3.C sections of this paper. This module represents the major chunk of digital twin reasoning process.

As an addition, we propose the Investor's portfolio review UI, which would give valuable notifications to the real investor. Investor will be notified by its digital twin on new NFO events, so an investment decision can be made on time. On the other side, when notified, investor can review detected NFO events and correct any piece of information that wasn't extracted correctly. This feedback will be saved by digital twin and used for active learning. This information is used as a dataset for a supervised model for NFO detection. This model will evolve through time and will be very helpful when the volume of feedback

from real investors get larger, while the model itself is designed to achieve high precision measure, which places it alongside other deep learning NLU models used in the extraction ensemble.

V. Experiments and results

End-to-end digital twin performance is evaluated on random 100 publications related to NFOs that were fetched online, and then evaluated manually, since the system works with unlabeled data. Additional 20 publications were fetched for the evaluation purpose of the model evolution module and the feedback loop of the real investor to its digital counterpart.

Results are presented for each extracted piece of information separately, since not all fields are equally important to the investor, and they have a different level of bias through question configurations as well. The first column represents the percentage of articles where the correct extraction was ranked at the top by scoring module, while the second column represents the percentage of articles where the correct extraction was in the list of candidates for that field, but not ranked at the top. This means that extraction ensemble detected it as a possible correct extraction, but the scoring module didn't place it at the top. However, investor can correct those TOP 1 candidates and give the model a chance to evolve through time and correct its scoring.

NFO field	TOP 1	Candidate
Fund name	94%	98%
Investment company	89%	96%
Launch date	57%	71%
Fund type	61%	64%
Currency	80%	86%
Country	69%	72%

Table 2 Extraction results before investor feedback introduction

It's important to mention that some of these extractions were not even possible to be successful since they were not even present in the article. For example, country is sometimes not mentioned since it's derived from the article data source, which can be country-specific.

Some fields, like date of fund launch is not listed in the TOP1 since articles usually contain multiple dates, and the extraction model can get confused. That's where the model evolution pipeline through investor's feedback gives the best results. These results are extracted from 20 random publications that were not a part of feedback loop and were not added to the dataset.

NFO field	TOP 1	TOP1 after evolution
Fund name	95%	95%
Investment company	90%	100%
Launch date	55%	65%
Fund type	70%	75%
Currency	80%	85%
Country	65%	65%

Table 3 Extraction results after investor feedback introduction

Apparently, model performance is increased because the feedback loop model manages to catch more patterns. This behavior would be even more visible with more data added to the model training, which will happen with constant feedback from the investor to its digital twin.

VI. Next steps

This paper covered the creation of the digital representation of the investor (digital twin) for the NFO discovery and process, which represents the most time-consumable part of the process, according to investors.

Next steps for research in this area includes the rest of investors reasoning process in order to get a score on how well the fund might perform after it has been launched. By leveraging this score, the digital twin can cover the investment decision process as well, to automatically invest in potentially good funds without much human intervention. Next steps would include following research questions:

- Analysis of the investment company that launched an NFO, from historical economic information about that investment company
- Analysis of other funds launched by the same investment company, from historical information, and how they performed after some time. Investing in a new fund is risky, and this information is very valuable to the investor
- Analysis of other similar funds on the market and how did they perform after some time. This information also enabled the investor to make a decision that will result in maximum gains.

These research questions represent the next part of investors reasoning process and its digital twin, the investment itself.

VII. References

[1] K. H. Baker and G. Filbeck, Hedge Funds - Structure, strategies, and performance, Oxford University Press, 2017.

[2] T. Manjunath, D. Yogish, S. Mahalakshmi and H. Yogish, "Smart question answering system using vectorization approach and statistical scoring method," in *Materials Today Proceedings*, 2021.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," arXiv, 2017.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," ArXiv, 2016.

[5] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv, 2019.

[6] K. Pearce, T. Zhan, A. Komanduri and J. Zhan, "A Comparative Study of Transformer-Based Language Models on Extractive Question Answering," arXiv, 2021.

[7] M. R. A. Natural Language Computing Group, "R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS," Microsoft Corporation, 2017.

[8] S. Schwager and J. Solitario, "Question and Answering on SQuAD 2.0: BERT Is All You Need," Stanford University.

[9] H. Zhangning, "Question Answering on SQuAD with BERT," Stanford University, 2019.

[10] P. Dangeti, Statistics for Machine Learning, Packt Publishing, 2017.

[11] J. Leskovec, A. Rajaraman and J. Ullman, Mining of Massive Datasets, Cambridge. ISBN 9781108476348, 2020.

[12] Y. Zhang and Z. Xu, "BERT for Question Answering on SQuAD 2.0," Stanford University, 2019.