Oxford Review of Economic Policy, Volume 37, Number 3, 2021, pp. 564–584

Artificial intelligence research in finance: discussion and examples

Manuela Veloso,* Tucker Balch,** Daniel Borrajo,*** Prashant Reddy,**** and Sameena Shah*****

Abstract: Artificial intelligence (AI) is a science and engineering discipline that is highly relevant to financial services, given the significant amount and diversity of data generated (and consumed) as those services are delivered worldwide. Global banks process billions of international payments each day, while equity exchanges handle trillions of orders and billions of transactions. All of this activity is recorded as data, and driven by exogenous information sources such as news services and social media. To address these challenges, at J.P. Morgan, we established a new group dedicated to research at the intersection of AI and finance in mid-2018 to investigate how to develop and optimize the use of AI. In this article, we introduce and discuss the directions of focus of AI Research and present a few selective projects that illustrate potential novel applications to finance.

Keywords: artificial intelligence, finance, machine learning, natural language processing, simulation, synthetic data

JEL classification: C11, C15, C3, C53, C63, C88

* J.P. Morgan AI Research, USA; e-mail: manuela.veloso@jpmchase.com
** J.P. Morgan AI Research, USA; e-mail: tucker.balch@jpmchase.com
*** J.P. Morgan AI Research, USA, on leave from Universidad Carlos III de Madrid (consultant); e-mail: daniel.borrajo@jpmchase.com
**** J.P. Morgan AI Research, USA; e-mail: prashant.reddy@jpmchase.com
***** J.P. Morgan AI Research, USA; e-mail: sameena.shah@jpmchase.com

The authors gratefully acknowledge our colleagues who have conducted this research with us or collaborated on the writing of this paper: Selim Amrouni, Samuel Assefa, David Byrd, Naftali Cohen, Danial Dervovic, Sumitra Ganesh, Mahmoud Mahfouz, Alice McCourt, Armineh Nourbakhsh, Vineeth Ravi, Srijan Sood, Andrea Stefanucci, Robert Tillman, Svitlana Vyetrenko, and Zhen Zeng.

https://doi.org/10.1093/oxrep/grab019

I. Introduction

Banking and finance are among the oldest industries in human society, but they represent perhaps the last important industry to embrace computing, and in particular, artificial intelligence (AI). Though there has been an increasing number of successful applications of AI to a number of aspects of the financial industry, much work still remains to be done in order for us to take full advantage of the depth and importance of the available data for efficient and profitable operations.

At AI Research (AIR), we focus on seven long-term challenges or 'aspirational goals' for AI in finance. These are described in what follows, along with an overview of emerging successful techniques to address these challenges.

(i) Seven challenges for AI in finance

We have organized these challenges as 'aspirational goals' in three groups: foundational topics; stakeholder engagements; and a cross-cutting theme.

Foundational topics

Core challenges in finance and banking:

Challenge 1: Eradicate financial crime.

AI can be leveraged to detect and eliminate financial crime, including: (*a*) enhanced customer understanding; (*b*) the analysis of relationships between people and organizations based on relationships, locations, money flow, and newsworthy events involving customers and client organizations; (*c*) game-theoretic simulation and analysis; (*d*) processing public information sources so that the data can be more easily analysed by people and machines; (*e*) detecting unusual or anomalous financial activities; (*f*) behaviour mining, analysis, and simulation; and (*g*) cryptographic methods to impede fraud.

Challenge 2: Liberate data safely.

AI can support and enable secure information sharing in the financial industry while preventing leaking sensitive information: (*a*) help discover relevant dimensions and parameters of the data and label it so it's useful to researchers and machine learning (ML) algorithms; (*b*) access and leverage real data to develop effective interpretations, and meaning from hidden data through secure computable functions; and (*c*) develop and share 'synthetic' or artificially created data with research groups to more robustly test and evaluate prototype algorithms before exposing confidential data (see section II(ii)).

Challenge 3: Predict and affect economic systems.

AI can help us understand and model the many participants in complex economic systems: (*a*) enable simulation of economic systems using agent-based modelling methods; (*b*) develop learning agents capable of improving their performance with experience; (*c*) develop methods for calibrating the simulations and agents in reference to the current and future behaviour of the modelled systems; and (*d*) provide algorithms to predict potential future states of economic systems, including estimates of uncertainty.

Stakeholder engagements

AI can support and empower our key stakeholders:

Challenge 4: Perfect the client experience:

AI can enhance: (*a*) the ability to discover potential clients; (*b*) streamlining the process of 'onboarding' new clients; (*c*) connecting internal information regarding clients with external data in the public domain; and (*d*) anticipating the desires and goals of clients based on their observed behaviour.

Challenge 5: Empower the employee:

AI can empower operations and employees by: (*a*) providing methods that analyse specific business operations; (*b*) suggesting revisions to the workflow patterns employees use to complete data-centric tasks; (*c*) automating cognitive tasks; and (*d*) enhancing employee experience and job role fit.

Challenge 6: Enhance policy compliance:

AI can enable us to address a number of challenges related to our obligations to comply with laws, regulations, and policies: (*a*) discover how regulations can be translated into machine-readable form so they can be integrated to digital processes; (*b*) monitor regulatory and governmental publications for changes in law that affect our processes; and (*c*) create mechanisms that enable automatic updates to our monitoring systems.

Cross-cutting theme

Challenge 7: Establish ethically and socially good AI:

As AI practitioners we should enable ethical and socially good AI by: (*a*) improving existing ethically and socially good models; (*b*) supporting state-of-the-art establishment of standards, frameworks, and toolkits of AI and ML; (*c*) ensuring software is able to detect and mitigate improper bias in systems; and (*d*) deploying methods that ensure a system is ethically and socially good for clients, employees, and stakeholders.

(ii) Emerging AI techniques for finance

In this sub-section, we provide an overview of projects that we have explored. Our examples are derived from the work of AIR at J.P. Morgan on real problems facing our company. Key themes include:

- an **agent-based** view of problems in AI, including those in finance;
- **transformation of representation** to provide enhanced accessibility to data for machines and enhanced understanding of the data for people;
- **synthesizing data** to enable access to information for experimentation that might otherwise not be available and to augment existing research data sets;
- leveraging **graphical models** to understand and exploit relationships between data sets;
- **multi-agent simulations** to model complex systems such as electronic markets and explore counterfactuals;

- **analysis of fairness, ethics, and bias in AI systems** particularly as they are more widely used.

We discuss each of these themes in more detail below.

The agent-based view

While historical data are prevalent in finance research, this approach excludes the use of unseen or out-of-sample data. With our multi-agent simulation frameworks, we have developed infrastructure that provides an agent-based view for evaluation and experimentation, including the creation of realistic fictional scenarios. As part of section II(ii), we discuss this agent and simulation-based view that enables testing a variety of strategies for future decision-making, counterfactuals, and testing under completely novel unseen scenarios.

Transformation of representation

We noticed that a variety of disparate tasks which industry domain experts engage in may look varied from a task-specific point of view, but in fact, from a computing perspective, may be thought of as a change in how the information is represented. We identified numerous opportunities. For example, this could be from numerical data to text, numbers to picture, picture to action, text to knowledge graph, data to insight. In section II(i), we present a novel approach that considers changing the way a signal is represented by the computer from a time-series of numbers to image-based data for trading decisions based on stocks charts. In section II(iii) we present an interesting case of generating financial insights from data.

Synthesizing data

Accessing data for research, or development of new applications to serve clients can be more challenging than one might expect. Depending on specific use, the problem may concern no data, very little data, or noisy or biased data. In order to address the issue of little to no data availability, we have developed an area of research focused on generation and utilization of synthetic data. In section II(ii), we present one of our contributions in this area.

Graphical models

In many large companies, processes and data access can be 'siloed', which is to say that the data are only available to those working in a particular group. We have also observed that different groups may be using the same data for different purposes. Sometimes domain experts in a particular group may be enriching data (e.g., removing errors or adding annotations) which may be valuable for other groups too, but those types of connections and sharing across groups are not common. AI can help with the re-use and repurposing of data for multiple uses. This enables sharing such benefits more easily between groups, reducing duplication of work, as well as enhancing the quality of insights that may be derived from the data. We initiated a number of projects on this front, including effective graphical model-based approaches that have been reported in the literature (Tillman *et al.*, 2020*a*,*b*).

Outcomes, biases, and fairness

Based on numerous successful deployments and research experimentation, we find AI performing increasingly more complex tasks—even surpassing human accuracy at times. This brings ethical concerns around bias and fairness to the forefront. We have pursued research that involves reducing bias in ML models and make some practical recommendations.

II. Examples of applications of AI in finance

AIR has a large set of individual projects under way in which we have contributed novel AI approaches to finance problems. In this section, we provide summary reports of three of these projects. The work contributed by AIR is at different stages of deployment in the business.

(i) Contribution: computer vision for time series forecasting

The complexity of the data available in the finance domain, such as market data, leads to the use of many visualizations. Interestingly, such visualizations support many aspects of decision-making for problem-solving.

The problem

A common problem in many fields, including finance, concerns the prediction of future values in time series data. In general, we are interested not only in predicting the specific values, but also acting now on the predicted future values while anticipating the uncertainty or volatility of those values.

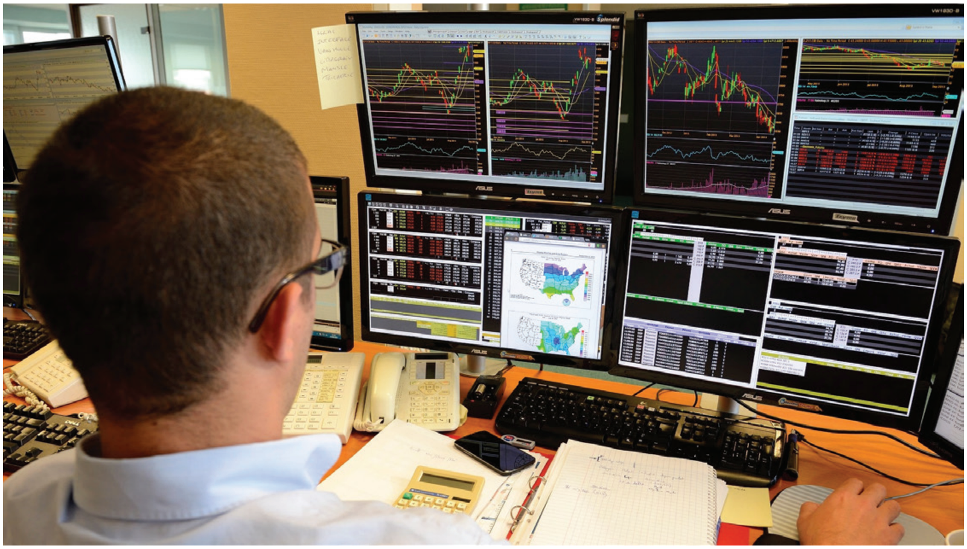
Approach

We observe that humans often analyse and act on numerical data presented as images—for instance, as charts of historical stock prices, or other sorts of visual presentations (see Figure 1). Our approach is to transform the data into images and train the system with example 'before' and 'after' images. Subsequently, the trained system can be presented with a new query (presumably the present situation) and generate a prediction in the form of a new image. Machine learning (ML) and in particular deep learning (DL) methods can be used for *classification* and *regression* problems. Classification problems centre on 'either/or' questions such as whether to buy a stock or not. Regression problems concern numerical prediction problems such as the future price or volatility of a stock.

Our work is motivated by a financial application. Figure 1 shows a financial trader executing trades while observing time series images on his desktop screens. Financial time series data is consumed in its numeric form but, as shown in Figure 1, decisions are often augmented by visual representations.

Time series forecasting has been extensively studied in statistical literature that uses historical information to predict future values. The underlying assumption is that past information contains signals that can be expected to continue. See Hyndman

Figure 1: A trader uses visual charts to make trading decisions

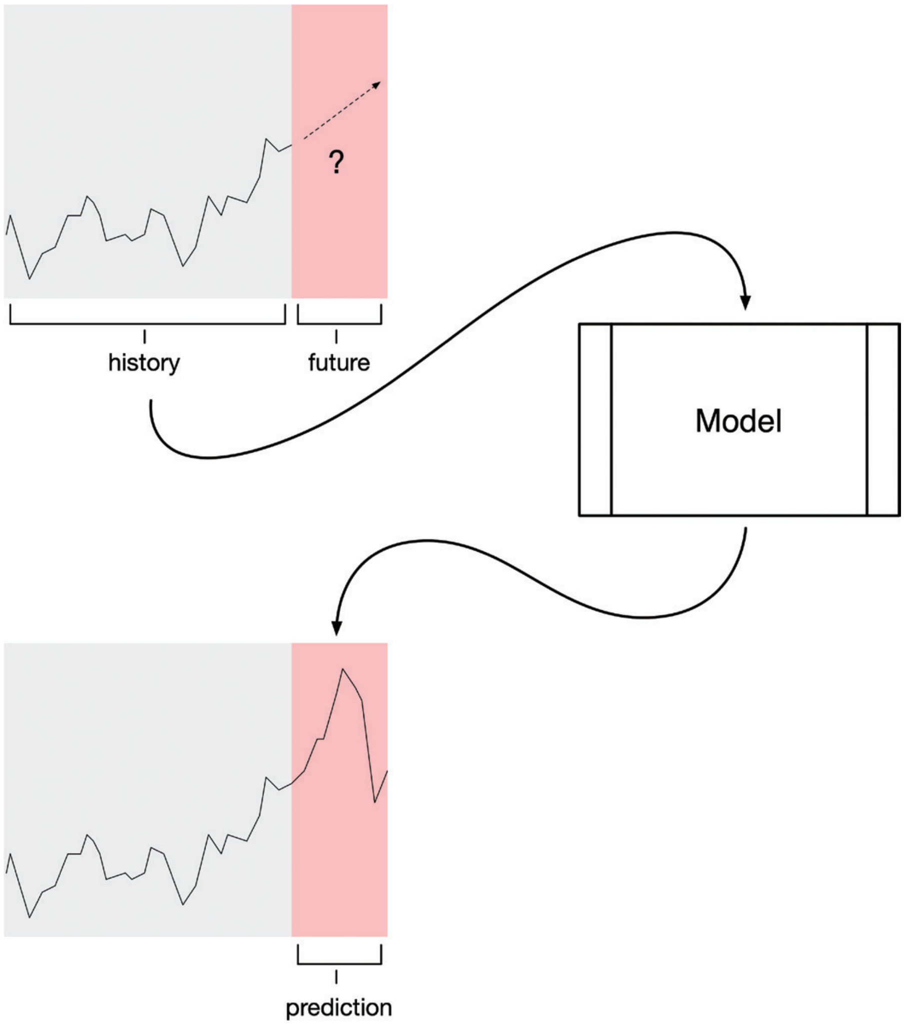


and Athanasopoulos (2018) for an overview. Over the years, many techniques have been developed for point-wise forecasting and confidence intervals about the prediction. The vast majority of successful forecasting algorithms use statistical methods (e.g. Wilks, 2011; Hyndman and Athanasopoulos, 2018; Krispin, 2019). Some are simplistic, based on rolling averages, while others involve more complex concepts such as seasonal decomposition, exponential smoothing, and auto regressive integrated moving average (ARIMA) models. Interestingly, ML methods have yet to significantly impact the time series forecasting field, even for nonlinear tasks (e.g. Makridakis *et al.*, 2018).

Recent work by our group (Cohen *et al.*, 2019; Zeng *et al.*, 2021) shows that computer-vision techniques and visual representations are effective in identifying and classifying trade patterns, as well as jointly predicting the price change direction of multiple assets. We follow these approaches and process time series data as images to produce corresponding forecast images of probabilistic future values. Our work is novel in that it involves a transformation from a numerical time series representation to an image-based representation. The work in this section was originally reported in Cohen (2019, 2019) and Zhen (2020).

In our approach, given a time series signal, we would like to produce a *visual* forecast of its future (see Figure 2). We first convert the numeric time series into an image as explained later and then produce a corresponding forecast image. Addressing this as a visual problem provides several advantages. By way of background, traditional approaches to prediction problems involve the creation of a model in which an equation or formula is created. Such formulas are called 'parametric' because one must find the parameters or coefficients of the formula. Visual time series forecasting is a data-driven non-parametric method, therefore not constrained to discovering a set of parameters. Thus, as shown when applied to the various datasets, the approach is flexible and adaptable to multiple data forms.

Figure 2: Approach: time series data are presented to a trained model as an image (top). The model outputs an image that includes a forecast region (red area, bottom)



In order to create the model, we generate a series of pairs of 'before' and 'after' example images from historical data that are presented to a DL algorithm for training (Figure 3). The information presented in the image pairs overlaps, as shown in Figure 3. For the study we report here, we train our system to reproduce the first 75 per cent of a time series image, and predict the next 25 per cent of the image. The predicted 25 per cent is then compared with the actual historical data to assess the quality of the prediction. When the process is used in 'live' prediction the 'future' 25 per cent represents the algorithm's estimate of the future values of the series.

To evaluate the approach, referred to as *VisualAE*, we examined four different time series datasets with various degrees of complexity (Figure 4): (i) harmonic data generated by combining two sine waves of varying phase, frequency, and trend (i.e. up or down), (ii) random time series data generated using a mean-reverting random walk

Figure 3: How historical data are sampled to create 'before' (top) and 'after' images to train the DL model

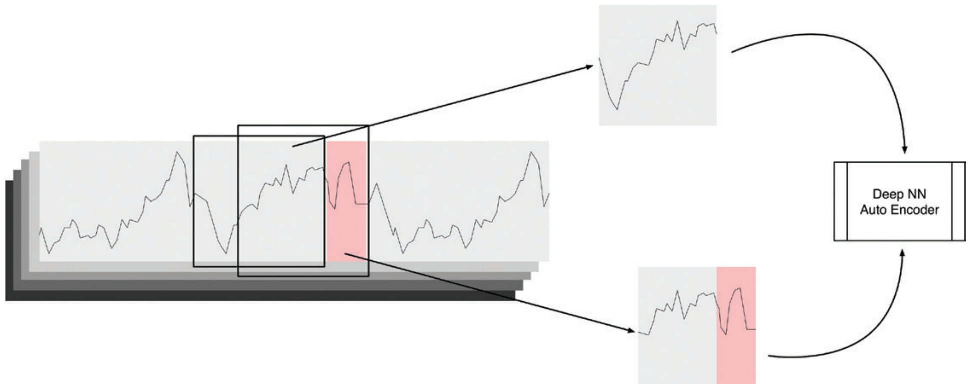
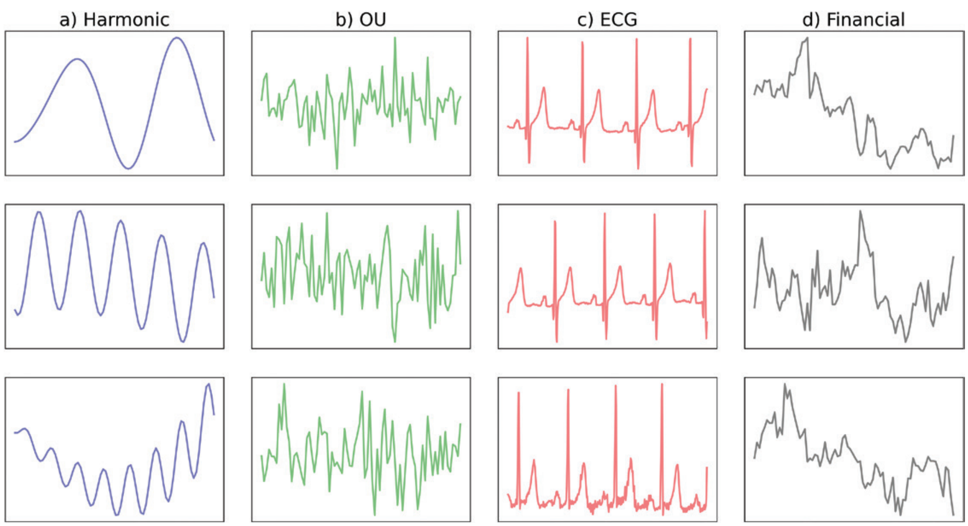


Figure 4: Examples of the four datasets: harmonic, OU, ECG, and financial



referred to as 'OU' in the figures, (iii) real electrocardiogram (ECG) records, and (iv) real historical stock data, referred to as 'financial' in the figures.

Insights and results

We tested the quantitative predictive capability of VisualAE against two more traditional methods: a simple statistical approach we refer to as 'RandomWalk' in which the forecast anticipates future probabilities for the time series to centre on the last numerical observation of the data stream, with a variance that grows with time, and 'NumAE', a basic neural network approach referred to in the DL literature as an *autoencoder*. Results for the various methods are depicted in Figures 5 and 6.

In Figure 5, we illustrate examples of predictions provided as images. In each example, the blue area indicates the region of known data used to create a forecast, while the red area indicates the future, predicted series. Note that the predicted region is

Figure 5: Results on several datasets using each of the prediction methods described in the text, from left to right, various predictive methods: Ground Truth, with 'future' region indicated in red, RandomWalk, NumAE, Visual AE. From top to bottom, examples from the different data sets: harmonic, OU (random walk), ECG, and financial

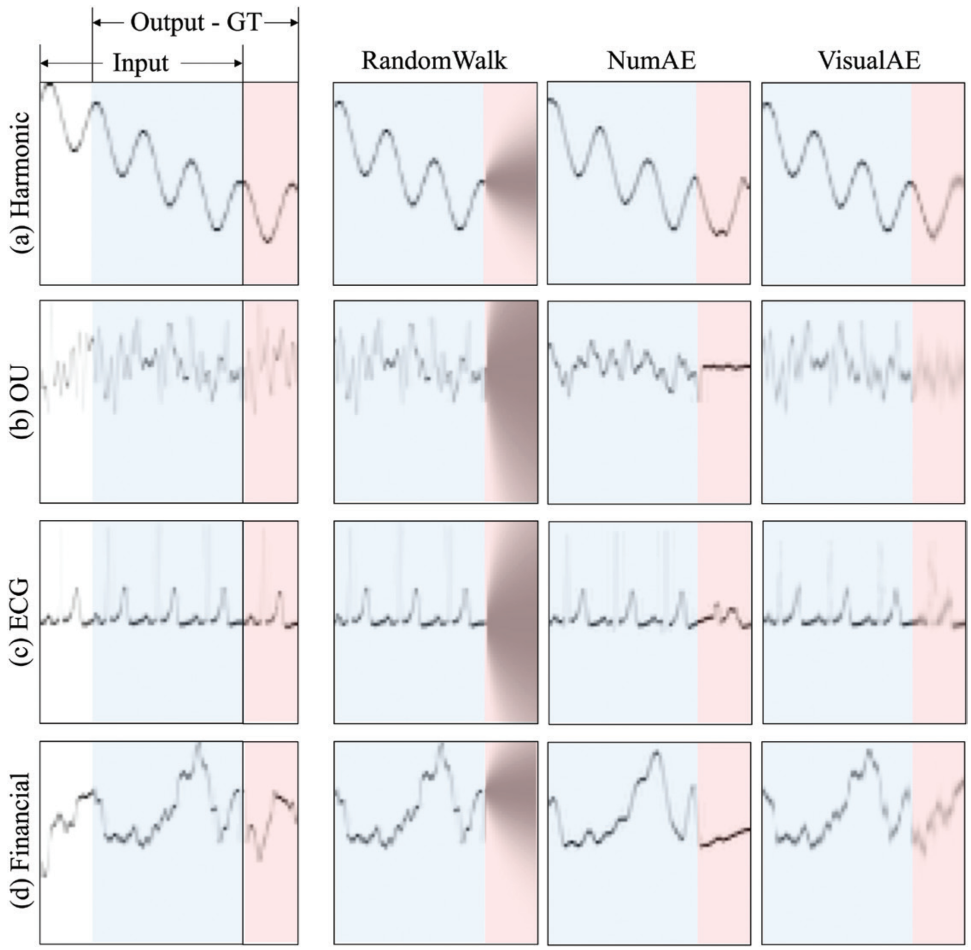
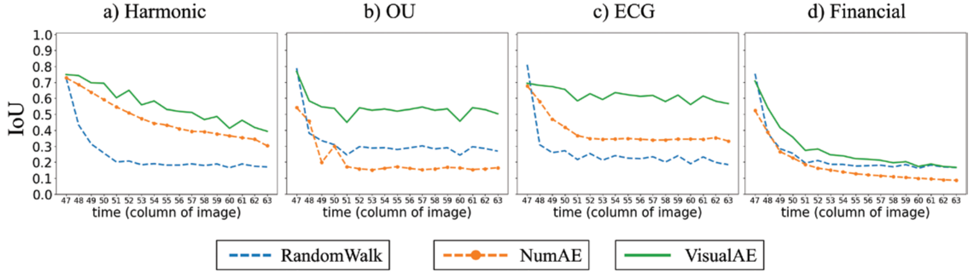


Figure 6: Results on several datasets. Each prediction method is represented by a line from left to right. The vertical value represents the quality of the prediction (larger values are better), while the horizontal axis represents time moving from left to right (further into the future)



sometimes 'fuzzy', indicating some uncertainty about the prediction. In Figure 6 we quantitatively evaluate the aggregated quality of each of the prediction methods across many examples as time moves from left to right. In general, as one would expect, the quality declines the further into the future we go.

Our experiments show that visual forecasting is effective for cyclic data, such as the harmonic and ECG time series, but somewhat less for irregular data, such as stock price. However, for all data sets, we find the proposed visual forecasting method to outperform the other numerical baselines.

Also, as illustrated in Figure 5, visual predictions can provide an uncertainty estimate versus a pointwise forecast. The visual forecast depicts a range of possible future predictions. In addition, financial time series data are often presented and acted upon without having access to the underlying numeric information (e.g. financial trading using the smartphone Robinhood application). Thus, it is important for current and future applications to consider visual solutions to this problem.

(ii) Contribution: synthetic data for finance

There is a vast amount of data generated in financial applications, but it is often inaccessible to researchers for exploration and development due to privacy risks.

The problem

For AI models to be effective, researchers need to train them on great quantities of representative data. Real data are often not available for this purpose. A primary reason for this is because financial data, given their confidential nature, have stringent restrictions in place that safeguard them from being used or shared. Real data may be limited in another way—they may be sparse or insufficient for analytical purposes or to train AI models. But researchers need data, and lots of them, to advance AI innovation. How can we create realistic data to enable research, protect data privacy and security, and address any data gaps?

Approach

These fundamental data challenges can be addressed by generating synthetic data (Assefa *et al.*, 2020) which mirror real data but are only a substitute for them—real data are never revealed. This approach both protects sensitive or impermissible data and can augment them to compensate for limitations.

We define synthetic data as data obtained from a generative process that learns the properties of the real data. Such processes are strictly different from the most commonly used data obfuscation techniques (e.g. anonymization or removing certain sensitive attributes) as our intention is to synthesize new samples that are related to but cannot be mapped back to the real data. Important considerations include the following.

- *Internal data-use restrictions*: Regulatory requirements may prevent data sharing between different lines of business within a company. Alternatively, teams may wish to begin working with data before the relevant approvals have been made.
- *Lack of historical data*: There is a limited amount of historical data to study certain events (e.g. flash crashes in the market, recessions, new regimes of behaviour) that makes studying the underlying mechanisms very challenging. It

is useful in various such settings to have counterfactual data for testing strategies and inferences.

- *Tackling class imbalance*: For uses such as fraud detection, the datasets are usually highly imbalanced, meaning that a very small percentage of the available transactions are actually fraudulent, while the large majority are legitimate. This causes traditional ML and anomaly detection techniques to fail. Realistic synthetic data, along with appropriate data imputation techniques, that serve to fill in the gaps when data are missing offer a promising approach to tackle this challenge.
- *Training advanced ML models*: Large-scale advanced ML (e.g. deep learning) is often carried out using cloud services, requiring compute resources and vast quantities of training data. Institutions may not be able to upload training data to these services for a number of reasons. Synthetic data can be used to train models, which can then be brought back to secure data centres operated on the company's premises to be used on real data. Moreover, training on synthesized data offers some protection from 'membership inference attacks', wherein model parameters can be used to extract training data.
- *Data sharing*: By sharing data between institutions and within the research community, better solutions can be found for technical problems faced by financial institutions. Sharing of realistic synthetic data allows financial institutions to do this in a way that satisfies their data-sharing restrictions.

For the purpose of illustration, we focus on *market microstructure* data, which refers to the data maintained and distributed by exchanges examining historical limit order books for a given financial asset. An order book is a data structure used by equity exchanges to keep track of orders that have not yet been filled because they await another matching order (e.g. a 'buy' order must be matched against a 'sell' order). Market microstructure data typically come in the form of time series describing, for example, a stock price over time. The granularity of the data is dependent on the frequency of the trading activity by the market participants. In the past decade, the rise of algorithmic trading and, specifically, high frequency trading has resulted in a significant increase in the amount of data available for research. However, the available data lack information, such as the ID of the individual trader associated with an order, thus limiting their utility for certain kinds of research. Therefore, an effort to create synthetic data from sets using real market data is necessary.

Of particular interest to the research community is limit order book data. A limit order book is used by exchanges to match buyers and sellers of a particular security (Bouchaud *et al.*, 2018). It is an electronic record of the outstanding orders in the market and represents a snapshot in time describing the supply and demand of the security (see Figure 7). It is based on the 'continuous double auction mechanism' used by major exchanges such as Nasdaq and the New York Stock Exchange. In these continuous auctions participants can submit both buy and sell orders and expect their trades to match instantaneously if a corresponding trade on the opposing side is present. Exchanges offer various order types. The two main types are market and limit orders. A market order is an instruction to buy/sell a specific amount of an asset without specifying the price. In contrast, a limit order specifies the price that should not be exceeded in the case of a buy order or gone below in the case of a sell order.

One technique to generate such data is through the use of multi-agent simulators and the construction of agent-based models composed of various trading agents with different strategies, objectives, and timescales (Byrd *et al.*, 2019). This allows for the generation of agent-specific data describing the behaviour, for example, of a market-maker. However, the limitation of such an approach is in the complexity and realism of the strategy designed in the model. Another interesting approach would be to use techniques of imitation learning (e.g. behavioural cloning—Bain and Sammut (1995)) to synthesize these data using historical observations and actions taken by an expert trading agent. Using these methods the system learns the behaviour of an agent participating in the market, then emulates it in a simulation.

The main technical challenge in synthesizing such order book data is that of representing aggregate decisions of many independent actors with differing risk tolerance, rationality, and motives. We would like the resulting synthetic data to reflect the statistical properties of real market data. The empirical properties of limit order book data have been studied extensively in the literature and are often referred to as *stylized facts* of the real limit order book data (Vyetrenko *et al.*, 2019). It is important to make sure that the empirical properties of the synthetic data follow, as closely as possible, those of real order book data. For example, it is empirically shown that lower spreads (the difference between the best bid and ask prices) are observed during periods of high trading volumes and that trading volumes are typically highest at the beginning and end of the trading day. These are two examples, of many, which would need to be taken into consideration when synthesizing data in the market microstructure domain.

Our multi-agent simulations are composed of over 5,000 separate trading agents, including four different types.

Noise agents:

Noise agents are designed to simulate the action of 'retail' traders who trade on demand once per day by placing a market order. The direction and the size of the trade are chosen randomly. In order to model higher trading activity in the beginning and at the end of a trading day (also known as intraday volume smile), noise agent arrival time is sampled from a distribution that follows the distribution of trading on Nasdaq. This arrival distribution pattern represents human trader propensity to be more active towards the beginning and the end of the trading day.

Value agents:

The value agents are designed to simulate the actions of fundamental traders who trade according to their belief regarding the exogenous value of a stock, but without any view of the order book microstructure. In our simulations, this *fundamental price* represents an estimate of the stock's price. All agents have access to the same, evolving fundamental, but when they observe it, we add a noise term to the observation. Each value agent arrives to the market multiple times according to a Poisson process and chooses to buy or sell a stock depending on whether it is cheap or expensive relative to its noisy observation of a fundamental price. Once the side of an order ('buy' or 'sell') is determined randomly, the value agent places a limit order at a random level either inside the spread or deeper into the order book. Value agents assist order book price formation by bringing external information to the book and are conceptually related to informed traders widely discussed in the literature.

Market maker agent:

The market maker agent acts as a liquidity provider by placing orders on both sides of the order book with a constant arrival rate. It tracks the bid/ask midpoint and places orders above and below that price.

Momentum agents:

The momentum agents base their trading decision on observed order book price trends. Our implementation compares past mid-price observations at two time intervals and places a buy order of random size if there is an upward trend and a sell order otherwise. The momentum agents are configured to arrive to the market at a constant rate.

In our experiments, we created a number of market configurations that differ in that they include different numbers of each type of agent.

Results and insights

In earlier work (Vyetrenko *et al.*, 2019) we enumerate 20 stylized facts drawn from previous studies of exchange markets and evaluate the statistical properties of data generated by our simulations versus example real data from Nasdaq. Space prohibits a discussion of all 20 stylized facts, but here we discuss four concerning the properties of the bid/ask midpoint as it progresses through a trading day.

- *Absence of autocorrelations*: Linear autocorrelation of asset returns over periods longer than 20 minutes are insignificant.
- *Heavy tails and aggregational normality*: The distribution of asset price returns shows fat tails; however, as one increases the period of time over which these returns are calculated, asset returns show slimmer tails.
- *Intermittency*: At all time scales, asset price returns must display a high degree of volatility.
- *Volatility clustering*: High-volatility events tend to cluster in time. A quantity used to measure volatility clustering is the autocorrelation function of the squared returns.

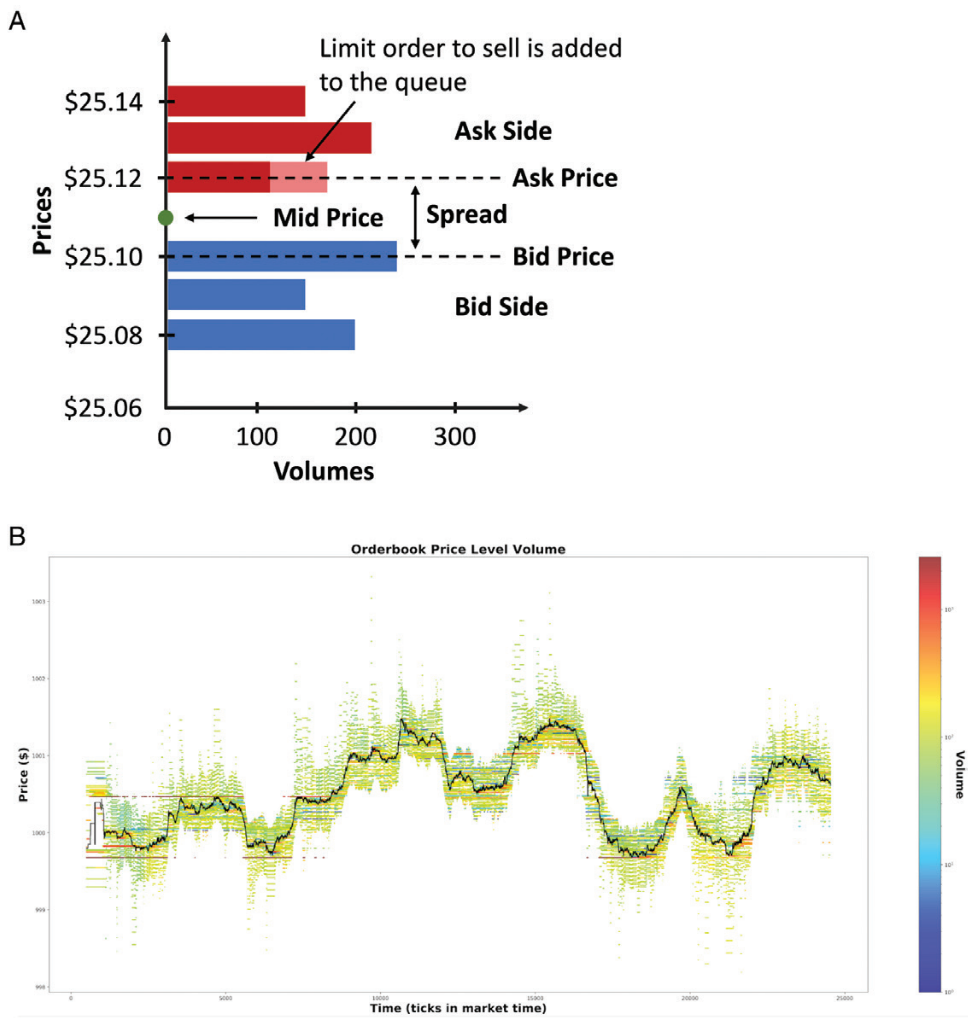
We compared the statistical properties of our synthetic market data against the properties of data from Nasdaq for 30 stocks' intra-day (during the trading day) behaviour in 2019 for these properties and found that they matched well, but not perfectly. Analyses for two of these properties (aggregation normality and absence of autocorrelation) are illustrated in Figure 8.

Overall, we found 'config_3' to provide statistical results closer to real Nasdaq data. The match is not perfect, of course, but we are pursuing two methods for improving the accuracy of our simulations. In particular see Vadori *et al.* (2020).

(iii) Contribution: learning document generation requests

We present how AI representation, matching algorithms, and human–machine interactions through natural language processing (NLP) techniques can transform the manual effort of developing reports, particularly considering the vast amounts of data at hand. The techniques can change a representation of data from one form to another—from numeric data to visualizations or presentations.

Figure 7: Visualizations of an exchange's order book. An order book is a data structure maintained by an exchange such as Nasdaq to facilitate matching buy and sell orders between traders. A: An order book at an instant in time. B: A visualization across an entire market day, the mid-price is indicated by the black line and the volume of orders at different prices is indicated in colour.



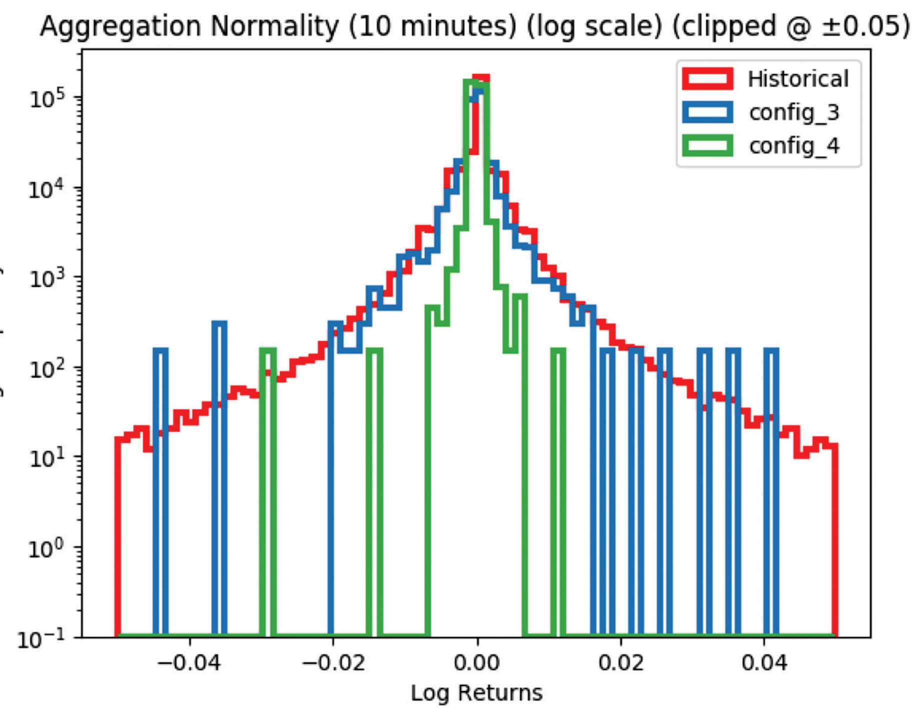
The problem

Employees in the financial services industry traditionally generate voluminous amounts of recurrent reports. The data used to produce reports can be overwhelming, with companies like J.P. Morgan processing more than $1 trillion in volume for merchant clients in 2016 (JPMorgan, 2016), and also generating upwards of several million PowerPoint slides every year based on an internal study. Further, a 2018 study found that 48 per cent of finance employees spend their time creating and updating reports, but most would prefer to spend that time interacting with the business (Deloitte, 2018).

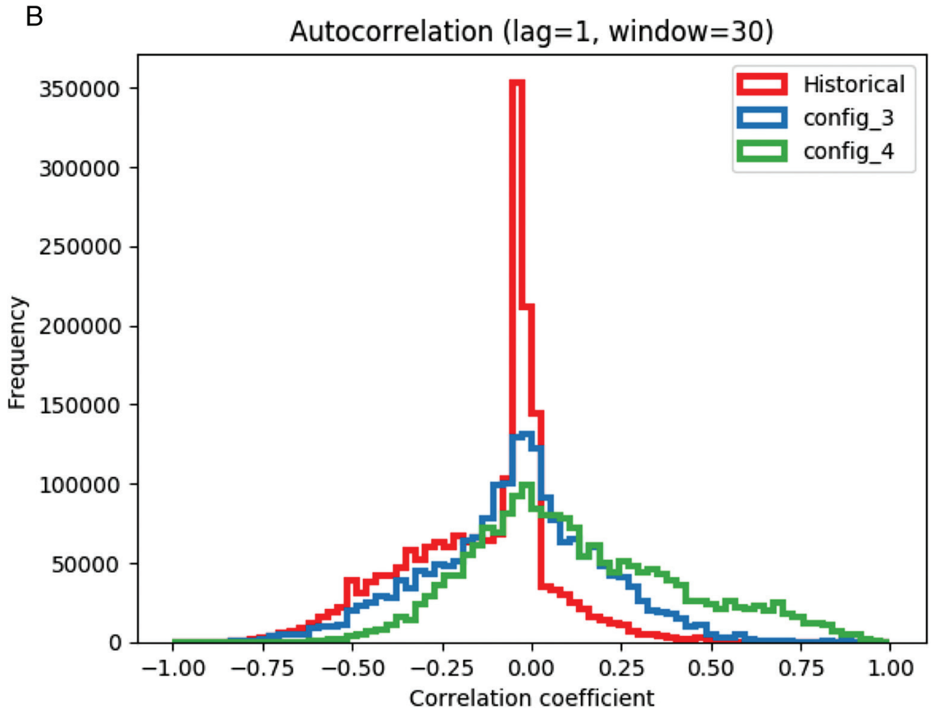
The reporting work involves processing large numeric datasets, parsing data, analysing data relationships, reshaping numeric data into a visualization, and generating

Figure 8: An analysis of two stylized facts for simulated data versus real intra-day price data for equities at Nasdaq. Both data sets are reported as log histograms. Nasdaq data are reported in red. Two configurations of the simulation are reported in blue and green. A: return distribution aggregated over 10-minute intervals. B: autocorrelation over a 30-minute interval

A



B



the report needed, often in MS PowerPoint (PPT) format. The task occurs daily, weekly, monthly, annually, and at any other requested frequency. The effort also requires transforming complex financial data, typically in a time-series format, such as cash-flows, client transactions, stock prices, market risk conditions, etc. As a result, the report-generation process is often tedious, time-consuming, error-prone, and subject to complicated and costly controls. The structure of these reports and underlying data typically do not change across the periodic updates.

Approach

We introduced and developed *DocuBot*, a novel, scalable framework that automates the generation of digital reports, specifically PowerPoint slides, through human–AI interaction (Ravi *et al.*, 2020). Users issue natural language (NL) instructions to DocuBot, which then does the following:

- generates digital reports, specifically PPT slides;
- creates and modifies PPT slides through user-driven NL instructions;
- identifies and clarifies any ambiguity received from the user and learns thereafter;
- adapts and improves through learning domain-specific vocabulary;
- grounds NL instructions to a set of skills to display on MS PPT slides;
- generates insights (or NL explanations of data);
- edits output with any single or global change(s) requested;
- displays language insights from the numerical data on MS PPT slides.

DocuBot's tasks are structured and targeted, which enables the use of automation methods that can leverage known properties of the information, such as how it is used by the business (i.e. 'business logic') and known constraints such as minimum or maximum values, while also enabling flexibility and expressivity. Figure 9 illustrates DocuBot's architecture. Below we describe each of these components in detail.

Parser:

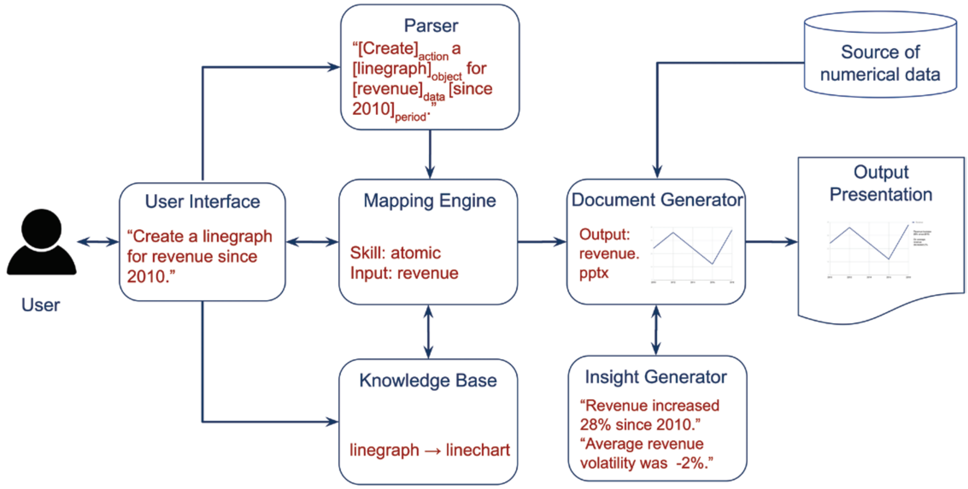
To understand and execute user requests, DocuBot needs to understand the user's intention and convert the NL request to a set of relevant actions that it is asked to perform. Determining intent in a general setting is an extremely hard problem. DocuBot narrows the scope of intent by assuming that the request must be related to the document at hand. This allows intents to be restricted to closed sets such as:

- 'updating client A's slides using client B's data', or
- 'generating a pie chart using 2019 data in "data.csv"'.

Once intent is established, DocuBot infers the remaining critical 'concepts that include: action, data, object, or presentation', from the user command. In other words, the parser needs to transform 'Please create a pie chart using energy data and add it in the weekly report' into 'Please $[create]_{action}$ a $[Piechart]_{object}$ using $[Energy]_{data}$ data and add it in $[weeklyreport]_{presentation}$'.

This above modelling approach is significant in that it allows parsing to be treated as a tagging problem. Tagging is a widely used NLP technique that involves assigning labels to components of a sentence. Our solution uses an approach referred to as conditional random fields (CRFs) (Lafferty *et al.*, 2001), that are a statistical modelling

Figure 9: DocuBot's architecture and components



method used for structured prediction. The key advantage of CRFs in this application is that they are able to take the context of the problem into account while assigning tags.

The CRF tagger was trained on 50 natural language examples. The examples were curated from users across J.P. Morgan and annotated manually to provide the corresponding tags. We used the natural language tool kit (NLTK) library (Bird *et al.*, 2009) to assign part-of-speech tags to the commands. Part-of-speech tags correspond to the syntactic function of words. For instance, the word 'bank' can be a noun or a verb depending on the sentence where it is used. Each word (or token) was represented as a feature vector where each vector element (feature) corresponds to a part-of-speech tag. Next, we trained a CRF with these vectors as examples. In many cases, ML methods require a large number of training examples, but in spite of the small size of the training data, the resulting model (evaluated on a test set of 25 unseen commands) performed well. Among other metrics to evaluate the performance, researchers use precision (given a class, how many times the learned model predicted correctly the examples of that class in a set of test examples), recall (how many of the true examples of the class were predicted by the model to be of that class), and F1 score (a combination of these two metrics). In our case, the results were: macro-averaged F1score of 0.849, precision of 0.86, and recall of 0.84. The parser's performance continued to prove robust during user testing.

It was expected that users would use inconsistent terminology, or context-specific vocabularies that DocuBot might not be aware of. This challenge was overcome by teaching DocuBot to proactively ask clarifying questions to remove ambiguity, and to dynamically adapt its behaviour based on user feedback. This dynamic adaptation method is similar to other approaches that learn from experience (Jordan and Mitchell, 2015). Below we describe in further details how using a knowledge base played a critical role in achieving the adaptability.

Knowledge base:

An initial knowledge base (KB) was created to include information about common main concepts (e.g. bank), sub-concepts (e.g. investment bank), and other financial domain vocabulary. Subsequently, the KB maintains mappings between main-concepts and sub-concepts as well as sub-concepts and vocabulary. It updates those mappings through NL interactions with users. While the KB maintains the data on terminology, the mapping engine, described below, is required to map the concepts presented in each NL instruction into structured actions.

Mapping engine:

The mapping engine interacts with the KB, parser, and user-interface to jointly map the concepts identified in a command to one of the possible action scenarios, aka skills, and also to clarify any ambiguity in intent, enabling the DocuBot to identify the appropriate corresponding available skills to use. DocuBot has a wide range of skills involving content creation and modification, for example, ability to generate documents such as digital presentations (e.g. PowerPoints, PDFs), output formats (e.g. JSON), variety of visualizations (e.g. pie charts, histograms, etc.), and placement. Though varied, the skills are a closed set, which provides the mapping engine the ability to map the user command to the required skill. DocuBot's framework is widely applicable to any general set of skills. Internally at J.P. Morgan, we have used the framework for non-standard visualizations, such as several types of statistical views, as well as diverse presentation formats. DocuBot's skills may vary for different business uses and, similar to human-learning, DocuBot's skills can be continuously enhanced and new skills can be added or learned through experience.

Insight generator:

The previously described components help DocuBot understand and standardize user input. Once the user command is properly mapped to concepts and skills, DocuBot needs to generate the visualizations and commentary necessary to be included in the output presentation. As previously described, the visualizations (e.g. pie charts, histograms, etc.) are created based on directions from the user. But presentations that only include visualizations with no context and no description may not be useful. Each visualization needs to be accompanied by commentary that summarizes it and describes the most important and relevant aspects of its underlying data. We refer to this NL commentary as *insight*. The insight generator component performs this task, using the following components:

- *Insight generator primitives*. We define *primitives* as the set of the various numerical operations that may be applied to the underlying numerical time-series data. Examples of primitives include: (i) absolute value primitives that compute metrics on the raw value of the time series, such as minimum, maximum, rolling average, or volatility; and (ii) comparison primitives that access the full history of the data to compute metrics about the time series and then compare the value at any slice, such as distance to the mean, percentile, or *comparative factor* (Perera *et al.*, 2018). The set of primitives is not fixed and is expected to grow over time.
- *Text generation from insights*. Data-to-text generation is a growing area of research in natural language understanding and natural language generation (Gatt *et al.*, 2017; Shen *et al.*, 2020). However, performance is still far from human baselines, which is a major hurdle to deployment in commercial settings.

Furthermore, user surveys revealed that users were interested in having consistent, well defined, and reliable text output. Therefore, we opted for a template-based approach. Templates define some fixed terms and some variable terms. For instance, the template '<company> share averaged <rate>% daily return' specifies that <company> and <rate> should be changed by relevant values (e.g. JPMorgan&Chase for <company> and 10 for <rate>). This template would work for all the presentations where DocuBot decides to report on the daily return of company shares. Relevant templates were curated as part of DocuBot's development process, which involved extensive engagements with end-users.

- *New primitives*. This process was also useful in collecting new primitives that were relevant for final users. For example, in a particular business use DocuBot was required to generate insights for a new key performance indicator (KPI). DocuBot was beneficial to the firm's executive management by highlighting relevant facts about KPIs amid disruptive events such as the COVID-19 pandemic.
- *Insights ranking and selection*. Not all possible commentary is interesting or relevant to users. Additionally, in certain cases the commentary can be overwhelming. In one case prior to the implementation of this module, DocuBot was generating 12,000 daily insights. DocuBot aims to provide the user with only the most interesting and valuable insights. For that purpose, it needs to rank and/or select insights. In order to sort the insights, we define their importance by a set of utility scoring functions such as *impact on internal revenue*, *anomaly compared to previous period*, or *anomaly compared to peers*. Each insight is ranked by an interpolated aggregate of all utility weights, and the top K insights are displayed. The interpolation weights and K are all configurable by end-users.

Results and insights

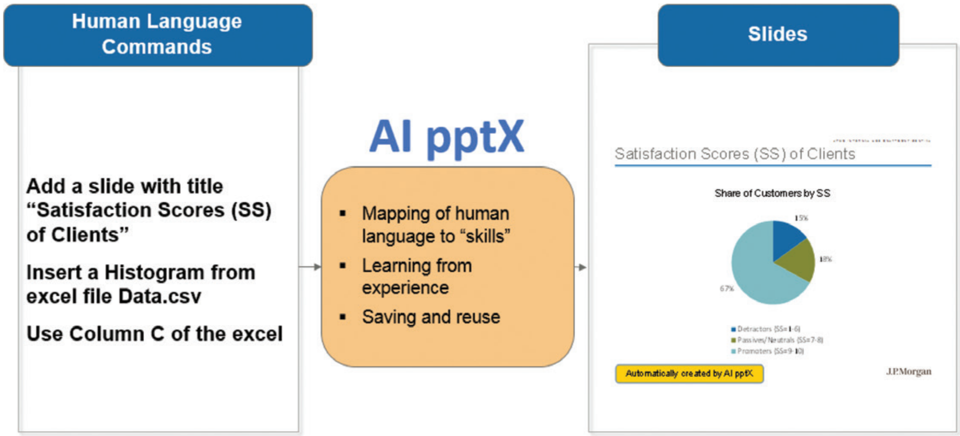
Currently, DocuBot supports multiple types of output including PDFs, MS PPT, and DOC files and webpages. Figure 10 illustrates the generation of a PPT slide as instructed by a user to DocuBot.

User feedback suggested that the automation introduced by DocuBot could potentially reduce analyst time spent in creating and updating PPT slides from *over several hours* to *less than 1 minute*.

III. Summary and conclusion

Over the last 2 years the J.P. Morgan AI Research group has taken on a large number of significant projects aimed at applying AI and machine learning to finance problems, with a view to innovate at the AI level. We frame our work around seven ambitious aspirational goals (enumerated in the introduction) to illustrate and provide a road map to transform the business of finance. We reviewed our experiences in this endeavour, including a discussion of the AI methods found to be most effective against these goals.

Figure 10: Example of DocuBot for PPTX



We also delved more deeply into three example projects that illustrate our approach to AI in Finance.

References

Assefa, S., Dervovic, D., Mahfouz, M., Balch, T., Reddy, P., and Veloso, M. (2020), 'Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls', ICAIF '20: ACM International Conference on AI in Finance, October.

Bain, M., and Sammut, C. (1995), 'A Framework for Behavioural Cloning', *Machine Intelligence*, **15**.

Bird, S., Klein, E., and Loper, E. (2009), *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media.

Bouchaud, J.-P., Bonart, J., Donier, J., and Gould, M. (2018), *Trades, Quotes and Prices: Financial Markets under the Microscope*, Cambridge, Cambridge University Press.

Byrd, D., Hybinette, M., and Balch, T. H. (2019), 'ABIDES: Towards High-Fidelity Market Simulation for AI Research', *arXiv.org*, arXiv:1904.12066.

Cohen, N., Balch, T., and Veloso, M. (2019), 'Trading via Image Classification', *arXiv preprint* arXiv:1907.10046.

Deloitte (2018), *Crunch Time 7: Reporting in a Digital World*, December.

Gatt, A., Krahmer, E., Van Deemter, K., and van Gompel, R. P. (2017), 'Reference Production as Search: The Impact of Domain Size on the Production of Distinguishing Descriptions', *Cognitive Science*, **41**, 1457–92.

Hyndman, R. J., and Athanasopoulos, G. (2018), *Forecasting: Principles and Practice*, 2nd edn, Melbourne, OTexts.

Jordan, M. I., and Mitchell, T. M. (2015), 'Machine Learning: Trends, Perspectives, and Prospects', *Science*, **349**(6245), 255–60.

JPMorgan (2016), 'JPMorgan Chase Tops 1 Trillion in Processing Volume for Merchant Clients in 2016', https://www.bloomberg.com/press-releases/2016-12-12/chase-tops-1-trillion-inprocessing-volume-for-merchant-clients-in-2016

Krispin, R. (2019), *Hands-on Time Series Analysis with R: Perform Time Series Analysis and Forecasting Using R*, Birmingham, Packt Publishing.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018), 'Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward', *PloS one*, **13**(3), e0194889.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001), *'Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data'*, University of Pennsylvania, Departmental Paper, June.

Perera, I., Hwang, J., Bayas, K., Dorr, B., and Wilks, Y. (2018), 'Cyberattack Prediction Through Public Text Analysis and Mini-theories', in 2018 IEEE International Conference on Big Data (Big Data), IEEE, December, 3001–10.

Ravi, V., Amrouni, S., Stefanucci, A., Reddy, P., and Veloso, M. (2020), 'AI pptX: Robust Continuous Learning for Document Generation with AI Insights', *arXiv preprint* arXiv:2010.01169.

Shen, D., Zheng, M., Shen, Y., Qu, Y., and Chen, W. (2020), 'A Simple but Tough-to-Beat Data Augmentation Approach for Natural Language Understanding and Generation', *arXiv preprint* arXiv:2009.13818.

Tillman, R. E., Reddy, P., and Veloso, M. (2020*a*), 'Recommending Missing and Suspicious Links in Multiplex Financial Networks', ICAIF '20: ACM International Conference on AI in Finance, October.

— Potluru, V. K., Chen, J., Reddy, P., and Veloso, M. (2020*b*), 'Heuristics for Link Prediction in Multiplex Networks', in *Proceedings of the 24th European Conference on Artificial Intelligence* (ECAI 2020).

Vadori, N., Ganesh, S., Reddy, P., and Veloso, M. (2020), 'Calibration of Shared Equilibria in General Sum Partially Observable Markov Games', NeurIPS '20: Neural Information Processing Systems.

Vyetrenko, S., Byrd, D., Petosa, N., Mahfouz, M., Dervovic, D., Veloso, M., and Balch, T. H. (2019), 'Get Real: Realism Metrics for Robust Limit Order Book Market Simulations', *arXiv preprint*, arXiv:1912.04941.

Wilks, D. S. (2011), 'Statistical Forecasting', in *International Geophysics*, **100**, 215–300, Cambridge, MA, Academic Press.

Zeng, Z., Balch, T., and Veloso, M. (2021), 'Deep Video Prediction for Time Series Forecasting', *arXiv preprint* arXiv:2102.12061.