AI pptX: Robust Continuous Learning for Document Generation with AI Insights

Vineeth Ravi[1] [*] **Selim Amrouni**[1] [*] **Andrea Stefanucci** [*] **Prashant Reddy** [*]

Manuela Veloso [*] [†]

Abstract

Business analysts create billions of slide decks, reports and documents annually. Most of these documents have well-defined structure comprising of similar content generated from data. We present *AI pptX*, a novel AI framework for creating and modifying documents as well as extract insights in the form of natural language sentences from data. AI pptX has three main components: (i) a component that translates users' natural language input into "skills" that encapsulate content editing and formatting commands, (ii) a robust continuously learning component that interacts with users, and (iii) a component that automatically generates hierarchical insights in the form of natural language sentences. We illustrate (i) and (ii) with a study of 18 human users tasked to create a presentation deck and observe the learning capability from a decrease in user-input commands by up to 45%. We demonstrate the robust learning capability of AI pptX with experimental simulations of non-collaborative users. We illustrate (i) and (iii) by automatically generating insights in natural language using a data set from the Electricity Transmission Network of France (RTE); we show that a complex statistical analysis of series can automatically be distilled into easily interpretable explanations called AI Insights.

1 Introduction

The financial services industry must process huge amounts of data as well as generate a variety of recurrent reports, including PowerPoint decks, as part of their daily business and operations. Slide reports are often created based on manual tedious analysis and visualization of underlying structured data. Billions [Parker, 2001] of slide decks are created across companies and to highlight the problem's significance; an internal study revealed that analysts in a department of J.P. Morgan manually create over 8 million PowerPoint slides every year. We have found through discussions with domain experts that business analysts often create and periodically update standard reports based on the most recent financial data. The structure of these reports and underlying data typically do not change across these periodic updates. In this paper we introduce a novel framework, AI pptX, to automate the generation of data reports; specifically PowerPoint slides in a real-world setting through human-AI interaction. AI pptX provides the ability to create and modify content in PowerPoint presentations through natural language instructions, with the capability to adapt and improve by learning from experience through human-AI interactions. We further introduce AI Insights that automatically generates natural language explanations of data and content displayed on the slides in these presentations.

[*]JPMorgan AI Research, 383 Madison Avenue, New York, NY 10036

[†]{vineeth.ravi, selim.amrouni, andrea.stefanucci, prashant.reddy, manuela.veloso}@jpmchase.com

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2 Framework and Experimental Data

The AI pptX framework consists of three main components based on symbiotic human-AI interactions (i) Automated Document Generation through Mapping Natural Language, (ii) Learning Natural Language From Experience Using Knowledge Base, and (iii) Insight Generation from Structured Data. AI pptX uses (i) to map human language instructions to underlying "skills". Skills refer to the ability of AI pptX to perform a task successfully. In particular, we refer to tasks that involve the generation of contents and template formatting that can be automatically executed by AI pptX on behalf of the human. AI pptX uses (ii) to enable robust continuous learning of mappings and skills through human feedback, by prompting questions and clarifying human instructions. The third component generates meaningful hierarchical explanations of the data, by (i) scanning and processing the data through a set of insight generators, (ii) generating explanations in the form of natural language sentences, (iii) ranking the insights based on predefined measures of relevance, and (iv) automatically generating slides to represent the prioritized trends. Our framework is applicable to any type of insight generators, but in this paper we have implemented insight generators focusing on the variation of time series compared to historical values, due to the primary relevance of this analysis in reports typically produced in our industry.

The novelty of the framework lies in (i) applying AI representation and robust continuous learning techniques to data, (ii) allowing users to combine individual instructions in complex tasks to be saved for future use, (iii) automatically generating explanations and slides based on the trends highlighted by AI Insights. AI pptX has great value and impact in the financial services domain given the prevalence of repetitive tasks in the financial domain by executing instructions. Internally at J.P. Morgan, we have extensively tested AI pptX on financial data. Specifically, we have developed a prototype to create a slide deck presentation having content about the financial performance of companies. The prototype has created excitement among senior business stakeholders and has been showcased to hundreds of employees in demo sessions.

We support our contributions with experimental results, by enabling 18 users to interact with AI pptX for creating content. We also showcase the robustness of AI pptX to non-collaborative users with experimental simulation of a variety of users. Given the confidential nature of the financial data, we demonstrate the AI pptX framework using a data set [RTE, 2019] from the Electricity Transmission Network of France to generate content and explainable insights. The RTE data set has comparable features to the private financial data set. The data set represents the energy production volume in France by regions and energy sources, with data points available for every 30 minutes from January 2013 to June 2019. Applying the AI pptX framework to a non-financial data set demonstrates the broad applicability of this emerging technology to other industries. In addition, to make financial data available for public use, we have launched an effort to create synthetic financial data.

3 Automated Document Generation through Mapping Natural Language

In this section, we introduce how AI pptX parses human language commands and maps them into skills. AI pptX understands human language instructions [Vittorio et al., 2015] by predicting labels [T. Kollar and Roy, 2014] for every token using a frame semantic parser. The output labels obtained are used for mapping human instructions to skills that affect content and formatting of slides in presentation decks. Moreover, AI pptX has the capability to log and save user commands, for future use. This reduces human effort and time in creating or updating recurrent presentations, which is a novel contribution of our paper.

Understanding Human Language Instructions AI pptX receives instructions through natural language commands to perform tasks. AI pptx understands the human language by parsing the sentence for extracting certain words or phrases and tagging them with labels. These labels are used for mapping human instructions to skills, which can be executed by AI pptX. AI pptX contains a trained frame semantic parser for predicting labels of natural language input. The labels we use to demonstrate the capabilities of AI pptX are (i) **action** (tasks the user can request), e.g., *Create*, *Modify*, *Save*, *Add*, *Delete*, *Execute*, (ii) **object** (content that AI pptX can automatically create), e.g., *Piechart*, *Histogram*, *Linegraph*, *Insights*, *CompanyBriefingDeck*, (iii) **data** (data source files), e.g., *energy.csv*, *production.xlsx*, and *RTE data set*, *nuclear energy data*, (iv) **presentation** (digital presentations where AI pptX can add the slide or *object* that has been created), e.g., *energyreport.pptx*, *weekly energy presentation*, *MonthlyUpdate.json*.

The parser is trained on 50 natural language commands (training data), annotated manually with labels, which are commonly used for creating presentations in the firm. The first step of the training process is to tokenize the sentences and find the Parts-of-Speech (POS) tag, of every token in the training data set using the NLTK library [Bird et al., 2009]. AI pptX generates a feature vector for every word comprising of features based on the POS tags of the current, next, and, previous words, as well as features that are directly dependent on the current, next, and, previous words themselves. AI pptX uses conditional random fields (CRF) [Lafferty et al., 2001, Sha and Pereira, 2003, Sutton and McCallum, 2012] as implemented in CRFsuite [Okazaki, 2011], and called through the python-crfsuite package [Korobov et al., 2018] for training to obtain a resultant CRF model. The weights w of each feature are learned using the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) quasi-Newton optimization method [Nocedal, 1980, Liu and Nocedal, 1989]. When the user interacts with AI pptX and gives a natural language command, AI pptX first tokenizes and generates the POS tags for each token in the user input. Next AI pptX generates the feature vector with the same features used during training for every word. Finally the trained CRF model predicts the labels as an output for each token in the natural language input command from the user. An example parse of a user command is: "Please $[create]_{action}$ a $[Piechart]_{object}$ using $[Energy]_{data}$ data and add it in $[weeklyreport]_{presentation}$."

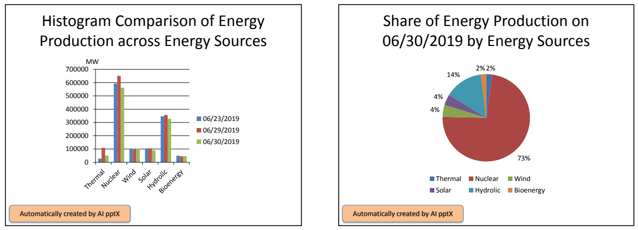
Although in general labelling tasks most often require large training sets, 50 commands proved to be enough for training our parser because (i) The size of our labels set is relatively small, (ii) Limited variation in natural language input by users to interact with AI pptX, and (iii) Limited number of *skills* available for a user to create/modify content and generate AI Insights in digital presentations.

Mapping Human Instructions To Skills AI pptX has the flexibility to use various types of underlying data to automatically generate documents such as digital presentations like PowerPoint's, PDF's, etc, and output files such as JSON requests, which are of great significance to business management and technology teams. We use the predicted output labels for mapping human instructions to *skills*. While AI pptX's framework is widely applicable to any set of skills, we leverage the python-pptx [Canny, 2019] library for generating PowerPoint decks. Skills can vary depending on different business usecases, but we broadly classify skills into two types: (i) Atomic, $\&$ (ii) Macro Skills.

Atomic Skills refers to tasks that create or modify the contents of one or few slides in a digital presentation from a single natural language input command from the user. The PowerPoint slides in fig. 1 are examples of AI pptX automatically generating slides using Atomic Skills. The parameters of date and title in the slides as well as the location of data values in the data source files are auto-generated from templates used in recurrent reports, which are common in business teams. Examples of natural language commands peforming Atomic Skills: (i) "Please $[\text{create}]_{\text{action}}$ a $[\text{Piechart}]_{\text{object}}$ about Energy Production using $[\text{RTE data set}]_{\text{data}}$ and include it in $[\text{'energy report'}]_{\text{presentation}}$ presentation." - fig. 1 (a), and (ii) "Please $[\text{create}]_{\text{action}}$ a $[\text{Histogram Comparison}]_{\text{object}}$ of Energy Production using $[\text{RTE data set}]_{\text{data}}$ and include it in $[\text{'energy report'}]_{\text{presentation}}$ presentation". - fig. 1 (b)

Macro Skills can create or modify the contents of many slides or the entire digital presentation from a single natural language input command from the user. AI pptX creates a predetermined template "CompanyBriefingDeck", of 10 slides generating content using "Finance" data, and adds the slides to a PowerPoint presentation with name "weeklyreport". Example natural language commands of Macro skills: "Please, can you $[\text{create}]_{\text{action}}$ a $[\text{CompanyBriefingDeck}]_{\text{object}}$ using $[\text{Finance}]_{\text{data}}$ data and add it in $[\text{weeklyreport}]_{\text{presentation}}$ deck."

Saving and Reusing Skills A novel contribution of our paper is the capability of AI pptX to log and save natural language commands, so we can reuse the saved combination of atomic and macro skills, adapting to the user's future business use cases. **Saving Skills** refer to tasks that allow the user to encapsulate a combination of atomic and macro skills, as a composite object. This allows the user to easily perform repetitive tasks in the future by reusing a majority of previous natural language commands. e.g: "Kindly $[\text{save}]_{\text{action}}$ the previous $[\text{twenty}]_{\text{data}}$ human commands as an object with name $[\text{Company Briefing Updated}]_{\text{object}}$." AI pptX has the ability to save the previous 20 commands as a new *object* with name "CompanyBriefingUpdated". In particular, this is useful for future recurrent tasks, because the user can get the new updated deck with just one single instruction instead of repeating several previously used natural language commands for creating and modifying



(a) Histogram (b) Piechart

Figure 1: Examples of AI pptX automatically generating slides using Atomic Skills

the deck. This new *object* created based on the user's requirements is particularly useful for saving human time and effort in creating future recurrent presentations.

Reusing Skills refer to previously saved skills that can be reused in the future. This reduces time and effort for the user when creating recurrent presentations which have similar content and templates. e.g: "Can you please $[\text{execute}]_{\text{action}}$ the saved task $[\text{Company Briefing Updated}]_{\text{object}}$." AI pptX's skill in this use case is to execute the saved object with name "CompanyBriefingUpdated", based on the previously saved 20 natural language commands given by the user. The object, encapsulates the combination of 20 atomic and macro skills saved by the user previously.

The automation introduced by AI pptX has the potential to help business analysts reduce their time spent in creating and updating PowerPoint slides from over 5 hours to less than 1 minute.

4 Learning Natural Language From Experience Using Knowledge Base

AI pptX can learn from experience, similar to NELL [Mitchell et al., 2015], by interacting with the user. AI pptX has a continuously learning "Knowledge Base" (KB), which enables the learning to map between natural language input and skills. The vocabulary used by users in a large firm can be inconsistent sometimes due to cultural and language differences. For e.g., *"graph"* could mean either *"Piechart"* or *"Histogram"* depending on an individual user's intentions. It is difficult to have a consistent and exhaustive vocabulary mapping list across all users in a large firm. AI pptX overcomes this limitation by having capability to dynamically adapt and improve its predictions through interactions with the user for feedback, learning from experience. AI pptX understands certain parameters from natural language instructions to execute skills for creating and modifying content in PowerPoint slides.

We define the following parameters used in our experiment: (i) $\boldsymbol{\rho}$ refers to the type of main-concept (e.g., $chart$), (ii) $\boldsymbol{\nu}$ refers to type of sub-concept (e.g., $piechart$, $barchart$) known from prior knowledge by AI pptX, (iii) $\boldsymbol{\omega}$ refers the set of words in the user's vocabulary (e.g., $piegraph$, $histogram$). The goal of the KB is to learn the mapping between $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$. For example, a user needs to specify the type of $chart$(main-concept) to create content in a slide ($\boldsymbol{\rho}$). The user can specify either a (sub-concept) $piechart$, $barchart$, etc. ($\boldsymbol{\nu}$), this sub-concept is declared by the user during the experiment and used by the KBs for growing their vocabulary. The vocabulary the user employs to refer to the sub-concepts refers to the set of words $\boldsymbol{\omega}$ (piechart can be referred as: pie, $piegraph$ or $pizzachart$, while the bargraph can be referred as: $histogram$ or $barplot$). We use $\boldsymbol{\zeta}$ to refer to the ground-truth sub-concept used during the evaluation phase.

Naive Knowledge Base We first validate our approach by designing a Naive Knowledge Base(NKB). The NKB aims to learn the vocabulary employed by users referring to a concept by permanently mapping the first new word $w_c \in \boldsymbol{\omega}$ learned from user input.

Definition 1. *Let NKB be the Naive Knowledge Base. Let Θ be the set of duplet concept-words (ρ-w_c) returned by the parser based on the user's natural language command. Let Ω be the set of sub-concepts. The NKB enables the mapping from Θ to Ω. To do so, the NKB implements the following functionalities: (i)* isInKB*: if w_c written in the KB accept, otherwise reject, (ii)* inferSC*: return the $\boldsymbol{\nu}$ linked to the w_c , and (iii)* addToKB*: add the mapping from w_c to $\boldsymbol{\nu}$.*

In order to demonstrate the NKB's learning ability, we task 18 users to interact with AI pptX for creating 5 slides using natural language commands. In this experiment we observe the learning

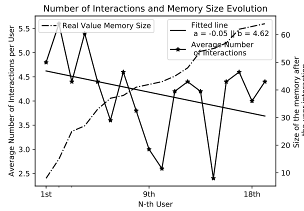


Figure 2: NKB Learning from user interaction

capability of AI pptX due to a decrease in user-input commands by up to 45% as well as identify the limitations of the NKB. Indeed, NKB learns a new word w_c it encounters by permanently mapping it to a sub-concept ν and does not have the ability to forget the mapping. NKB assumes a perfect world scenario, where only collaborative and informed users interact with AI pptX. However, we realize that in a real world setting, there are user generating wrong labelling. As AI pptX is designed to be deployed in a real-world environment, we design a Robust Knowledge Base (RKB) that has the ability to forget incorrect mappings and adapts to the majority of users.

Robust Knowledge Base This part introduces the Robust Knowledge Base (RBK). This new KB is designed to answer the limitations of the NKB.

Definition 2. *Let $\beta(\rho, \nu, w_c)$ be the $BeliefScore$ the RKB has for a word (w_c) in a sub-concept (ν) belonging to concept (ρ). $\beta(\rho, \nu, w_c) = P((w_c \in \nu))$.*

Definition 3. *Let RKB be the Robust Knowledge Base. Let Θ be the set of duplet concept-words (ρ-w_c) returned by the parser based on the user's natural language command.*

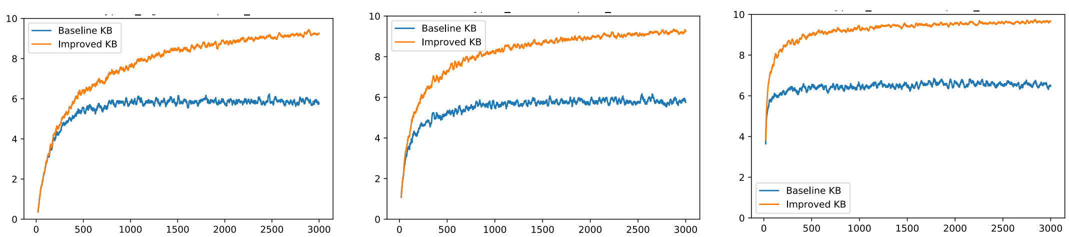
The RKB enables the mapping from Θ to Ω with the ability to *forget* incorrect mappings given by malicious users. Instead of permanently mapping the first input, RKB maintains a $BeliefScore$ of each triple ρ-ν-w_c. Every time a user interacts with AI pptX to choose a sub-concept, the RKB updates the discrete probability distribution and re-normalizes the score. To do so, the RKB implements the following functionalities: (i) *isInKB*: if the triplet ρ-ν-w_c is written in the KB accept, otherwise reject, (ii) *addToKB*: add ρ-ν-w_c to the RKB with an initial $BeliefScore$, (iii) *inferSC*: return $\arg\max_{u \epsilon \Omega}(\beta(\rho, u, w_c))$, (iv) *increaseBelief*: increase the $BeliefScore$ of the triplet ρ-ν-w_c, and (v) *decreaseBelief*: decrease the $BeliefScore$ of the triplet ρ-ν-w_c. Let s be the number of slides created so far. Then we have the following formulas for updating the $BeliefScore$ at each slide creation: $\beta(\rho, \nu, w_c) \xleftarrow{increaseBelief} \beta(\rho, \nu, w_c) * \frac{s-1}{s} + \frac{1}{s}$ or $\beta(\rho, \nu, w_c) \xleftarrow{decreaseBelief} \beta(\rho, \nu, w_c) * \frac{s-1}{s}$

Experimental Results and Discussion We perform many experiments with varying parameters. We represent the set of concepts ρ and subconcepts ν in our experiment using a dictionary. The main motivation to simulate human users is because its expensive for employing over 1000 humans to test AI pptX, we also want to reproduce the experiments with the goal of comparing the naive KB and robust KB without biasing humans. We simulate identical users in exactly the same order to compare the performance between the RKB and Naive KB using the $MatchingScore$ metric as well as demonstrate the superior performance of the RKB. We introduce $MatchingScore$ as an evaluation metric to compare performance of both KB's.

Definition 4. *Let V_{w_c} be the vector of words given by the user to create slides. Let $V_{p\nu}$ be the vector of predicted sub-concepts (ν) by the KB. Let V_ζ be the vector of true sub-concepts. We have: $|V_{w_c}| = |V_{p\nu}| = |V_\zeta| = n$. Let $\lambda(V_{p\nu}, V_\zeta)$ be the $MatchingScore$, then we have: $MatchingScore$ = $\lambda(V_{p\nu}, V_\zeta) = \sum_{i=1}^{n} \mathbb{1}(v_{p\nu_i} = v_{\zeta_i})$*

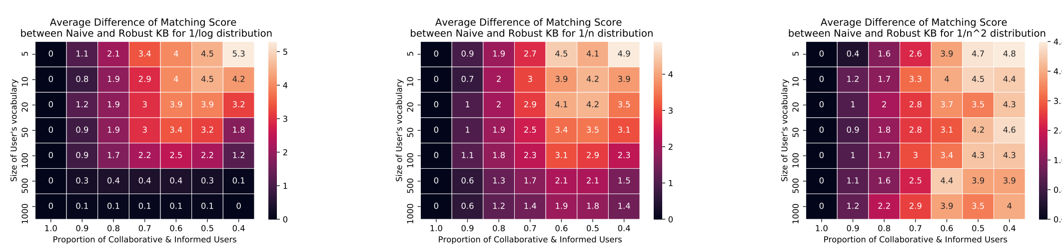
User Simulation To simulate a user, we generate a corpus of potential words humans would commonly use when interacting with AI pptX. We use Gensim [Řehůřek and Sojka, 2010] for loading the word vectors trained on the Google News dataset [Google and mrt033, 2019]. The model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple data-driven approach described in [Mikolov et al., 2013]. Given a ν, we use Word2Vec to retrieve the ordered list L of the N closest neighbors w_c. N can be varied from small to large values to account for the diversity of vocabulary employed by a user. For each ρ, the simulated user selects randomly a ν, then it picks with respect of a given *pdf* a corresponding w_c. The User Distribution (UD) or *pdf* for a user is modeled by being $\propto \frac{1}{log(n)}$, $\propto \frac{1}{n}$ or $\propto \frac{1}{n}$ with $n\epsilon\,[0, N]$. These models encapsulate the diverse behaviours from a wider to a more targeted vocabulary. We identify two types of users for simulation based on the NKB experimental results in fig. 2 (a) Collaborative&Informed users and (b) Non-Collaborative users. Users belonging to category (a) will use a w_c belonging to the list L of the same corresponding ν. Users belonging to category (b) will use a w_c belonging to the list L of another ν. We define α that represents the ratio of Collaborative $\&$ Informed users to the total number of users in our experiments. We use a parameter α to represent the ratio of type (a) users, which is varied between 0.4 and 1. We assume in a real-world scenario at least 40% of users are collaborative. The parameter N is varied between 5 and 1000, assuming that any user would know at least 5 words. The parameter UD (*pdf*) is varied between $\propto \frac{1}{log(n)}$, $\propto \frac{1}{n}$ or $\propto \frac{1}{n}$ distributions with $n\epsilon\,[0, N]$.

Experiment Parameters We repeated the experiment 10 times, resetting NKB and RKB every time. Each experiment is divided in two phases, a *learning phase* and an *evaluation phase*. During the learning phase, the two KBs are exposed to a proportion α < 1 of Collaborative $\&$ Informed Users. In the evaluation phase, both KB's are exposed only to Collaborative $\&$ Informed Users. For each of the experiment (both in training and testing), the KBs are exposed to the creation of 3000 slides. We average the results obtained across the 10 experiments and show them in fig. 3 and fig. 4.



(a) (b) (c)

Figure 3: *Training phase* evolution of $MatchingScore$ in different experimental scenarios of the creation of 3000 simulated slides with $N = 50$ and $\alpha = 0.6$ for different *pdf* (Average of 10 simulations and smoothing rolling window of 20). The less users' vocabulary variety is wide, the faster the KB learns. (a) *pdf* $\propto \frac{1}{log(n)}$ (b) *pdf* $\propto \frac{1}{n}$ (c) *pdf* $\propto \frac{1}{n^2}$



(a) (b) (c)

Figure 4: *Testing phase* simulation results comparing NBK and RBK. Heatmap showing the average difference of $MatchingScore$ (score of RKB minus score of NKB) for the creation of 3000 slides. The $MatchingScore$ is between 0 and 10. The x axis corresponds to the ratio α, the y axis corresponds to the vocabulary size of the user. (a) *pdf* $\propto \frac{1}{log(n)}$ (b) *pdf* $\propto \frac{1}{n}$ (c) *pdf* $\propto \frac{1}{n^2}$

5 Insight Generation from Structured Data

In this section, we introduce the automated generation of AI Insights. We present: (i) the set of *primitives* that generates insights from the raw data, (ii) the mapping from insights to human-friendly text, (iii) a novel technique for the ranking and selection of insights, and (iv) the novel capability for hierarchical analysis. We assume that the structured data set has a temporal dimension. Let $\{x(\tau)\}_{\tau=0}^{t}$ be a time series with data $x(t)$ at time t. We further assume that the data sample at each time step is structured along one or more additional dimensions. Let $\mathcal{D}$ be the complete data set.

Insight Generator Primitives

Definition 5. *Let* $\Psi = \{\psi_j\}_{j=1}^{m}$ *be the set of* primitives *representing any function to be applied to* $\mathcal{D}$*.*

Definition 6. *Let* $\mathcal{I} = \{I_k\}_{k=1}^{n}$ *be the set of insights generated from the primitives* Ψ*:* $\mathcal{I} = \Psi(\mathcal{D})$

Examples of primitives ψ_j include: (1) Absolute value primitives: These ψ_j compute metrics on the raw value of the time series: minimum, maximum, rolling average, volatility, etc. Or, (2) Comparison primitives: These ψ_j use the bot access to the full historic of the data to compute metrics about the time series and then compare the value at any time to these metrics: distance to the mean, percentile, *comparative factor* [Perera, 2018], etc. The set of primitives Ψ is not fixed and is expected to grow over time. We focus on the use of the *comparative factor* as illustration to creatre insights.

Definition 7. *The comparative factor is a scale-independent measure* $function$ *of a current value relative to historical values. We use the Z-score as the comparative factor in our experimental results:* $C(t) = \frac{x(t)-\mu(t)}{\sigma(t)}$

The values of C generated by Ψ are generally numerical. Often, a reader is not able to easily understand easily the significance of a given value of C if it is presented as a real number. Hence, we need to formulate text sentences from the values of C.

Generate Text from Insights

Definition 8. *Let* χ *be the function that generates readable text based on raw insights and let* $\mathcal{T}$ *to be the set of text generated. Then, we have:* $\mathcal{T} = \chi(\mathcal{I}, \mathcal{D})$

Using the RTE data set $\mathcal{D}$ and the insights $\mathcal{I}$ generated by the primitives ψ_j, χ automatically generated the insight shown on the slide in fig. 5 (a): "On [2019-06-30]$_1$, [relative to the previous day]$_2$, the production of [Nuclear]$_3$ energy was [1683997]$_4$ MW, which is [significantly down]$_5$ by [-13.75]$_6$% compared to historical variation."

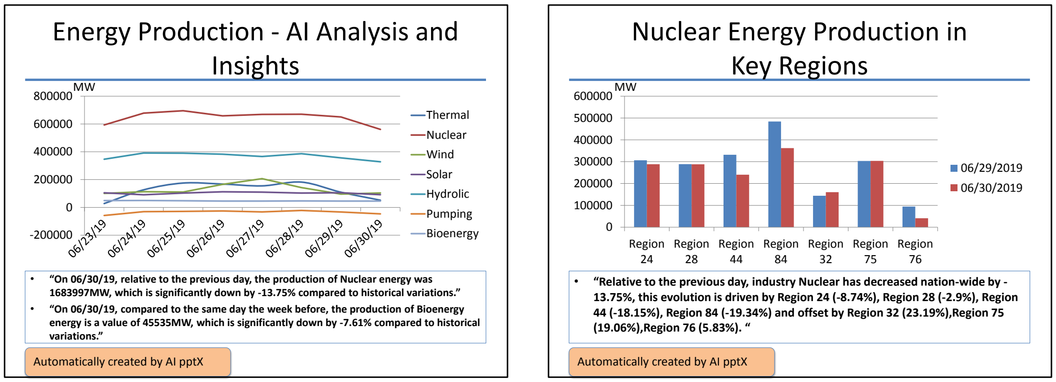
The bracketed parts are dynamically generated while the remainder of the sentence is the fixed skeleton of the insight. (2) is a comparative temporal difference, The comparative factor C of the series is computed for the values of $\Delta_T(t) = x(t) - x(t-T)$, the value of T is a parameter specified when designing the primitives Ψ or by the user at run time. T is taken as input by χ, which outputs a corresponding text description. Here, since $T = 1$, the $\Delta_T(t)$ corresponds to the difference in energy production from one day to the following, hence, χ outputs *relative to the previous day*. (5) is the Comparative Region: Given any uni-modal distribution, [Perera, 2018] defines comparative regions that are based on the mean μ and multiples of the standard deviation σ. Also, we define $C_{\text{min}}(t)$ and $C_{\text{max}}(t)$ to be the smallest and largest values of C values observed in the past (up to but excluding the current time). Let g be the function that maps a C value to its corresponding comparative region. Further, let h be the function that maps the comparative region to a text description, $\theta = h(g(C)$

In our example with the RTE data, $\Delta_T(t)$ follows a Student's t-distribution. AI pptX computes the μ and σ and then $C = -1.91$, which maps to the comparative region $[\mu - 2\sigma : \mu - \sigma[$, which further maps to the text insight *significantly down*. (1), (3), (4) and (6) are straightforward and completed according to the chosen production type. AI pptX aims to give the user only the most interesting ones. For that purpose, it needs to rank and/or select insights.

Insights Ranking and Selection

Definition 9. *Let* $\mathcal{I}'$ *be the set of insights to be shown to the user. Let* l *be a function that takes an insight and returns a score of this insight. Let* m *be a function that takes a score and selects or rejects the insight associated to the score. Let* Φ *be the composition of these two function* $\Phi = m \circ l$*. Then, we have:* $\mathcal{I}' = \Phi(\mathcal{I})$ *with* $\mathcal{I}' \subset \mathcal{I}$

To illustrate this with the RTE data set, we encoded different types of scoring functions l: (1) If $\Delta_T(t)$ is equal either to the maximum or minimum of the series $\{x(\tau)\}_{\tau=0}^{t}$, then the insight score is $+\infty$. (2) Otherwise, l is defined as the absolute value function; i.e., the score of the insight becomes the absolute value of its comparative factor, $|C|$. The selection function m first ranks the insights according to their scores and it then keeps the k insights with the highest scores. k is a parameter that may be fixed or specified by the user at run time. The insight shown above has been included by AI pptX because of its C value of -1.91, which maps to the highest selection score of 1.91. It is interpreted to mean that irrespective of the actual value of Nuclear production on 2019-06-30, its



(a) (b)

Figure 5: (a) Example of user requested slide with insights - Nuclear energy scale is divided by 3 (b) Example of automatically generated dive slide.

change relative to the previous day was particularly remarkable compared to historic daily variation. In fig. 5 (a), AI pptX presents the two most interesting insights $\mathcal{I}'$ along with the line graph of the last week of energy production by industry across all regions of France. In the contrary, for example, m rejected the least interesting insight (according to its score metric of 0.1), which is: "On [2019-06-30]$_1$, [relative to the previous day]$_2$, the production of [Wind]$_3$ energy is a value of [103572]$_4$MW, which is [slightly up]$_5$ by [8.17]$_6$% compared to historical variations."

Hierarchical Explanation of Insights Financial time series data often has a hierarchical structure. For e.g. (i) revenue of the trading division is the sum of the revenues of its desks, and (ii) the profit/loss of a firm is the sum of the profit/loss of its many Lines of Business. This motivates the need to not only create and highlight interesting insights, but also automatically generate further explanations for these insights. In AI pptX, we refer to this type of explanation as *dive analysis*.

Definition 10. *Let $\mathcal{I}$" be the set of insights AI pptX includes in its dive analysis: $\mathcal{I}$" $= \Gamma(\mathcal{I}')$ with $\mathcal{I}$" $\subset \mathcal{I}' \subset \mathcal{I}$ where Γ is a function that selects the insights to be included.*

Definition 11. *Let $\mathcal{T}$" be the set of dive text insights generated by AI pptX: $\mathcal{T}$" $= \eta(\mathcal{I}$", $\mathcal{D})$ where η is the function that generates dive text comments.*

Similar to financial time series data, the RTE data is hierarchical, since the national production of energy for France is an aggregate over regions and over production types. AI pptX selects the day-to-day Nuclear production as a dive insight since it has the highest selection score. It then performs a *driver/offset analysis* by region for a specified production type. The *driver/offset analysis* aims to understand which regions are responsible for the remarkable change in France's Nuclear energy production. $\Delta_T(t)$ is the difference of energy production for all of France between time t and $t - T$. Let $\delta_{i_T}(t)$ be the difference of production for region i. Then, $\Delta_T(t) = \sum_{i=1}^{R} \delta_{i_T}(t)$. The *driver/offset analysis* consists of comparing the sign of the product of $\Delta_T(t)$ and each of the $\delta_{i_T}(t)$. If this product is strictly positive (negative), then $\delta_{i_T}(t)$ is defined as a *driver* (*offset*) of $\Delta_T(t)$. The generated, by η, dive slide with AI insight $\mathcal{T}$" is shown in the textbox of the slide in fig. 5 (b).

6 Conclusion

In this paper, we have introduced a novel framework, AI pptX, to automate the generation of PowerPoint slides through human-AI interaction. To provide an easily interpretable explanation of the data displayed on the slides, we have also introduced the automated generation of AI Insights in these presentations. We have also demonstrated the robustness of AI pptX to adapt for different types of users through several experiments. Internally at J.P. Morgan, we have extensively tested AI pptX on financial data in real world use cases. In addition, by applying the AI pptX framework to a data set from the Electricity Transmission Network of France (RTE), we have demonstrated the broad applicability of this emerging technology to other industries.

Acknowledgements

We would like to thank Vamsi Alla, Jiahao Chen, Kwun Lo, and the Global Investment Banking team at JP Morgan for their contributions.

Author Contributions

[1]The first two authors contributed equally to this work.

Disclaimer

This paper was prepared for information purposes by the AI Research Group of JPMorgan Chase & Co. and its affiliates ("J.P. Morgan"), and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no explicit or implied representation and warranty and accepts no liability, for the completeness, accuracy or reliability of information, or the legal, compliance, financial, tax or accounting effects of matters contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

References

S. Bird, E. Klein, and E. Loper. Natural language processing with Python, 2009. URL http://nltk.org/book.

S. Canny. python-pptx: Create Open XML PowerPoint documents in Python, May 2019. URL https://python-pptx.readthedocs.io/en/latest/. Version 0.6.18.

Google and mrt033. python-crfsuite: A Python binding for CRFsuite, 2019. URL https://github.com/mrt033/word2vec.

M. Korobov, J. Cochrane, F. Gregg, and T. Peng. python-crfsuite: A Python binding for CRFsuite, Aug. 2018. URL https://github.com/scrapinghub/python-crfsuite. Version 0.9.6.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML '01, pages 282–289, 2001. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug. 1989. doi: 10.1007/BF01589116. URL https://doi.org/10.1007/BF01589116.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999792.2999959.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35 (151):773–782, 1980. ISSN 00255718, 10886842. doi: 10.2307/2006193. URL http://www.jstor.org/stable/2006193.

N. Okazaki. CRFsuite: a fast implementation of conditional random fields (CRFs), Aug. 2011. URL http://www.chokkan.org/software/crfsuite. Version 0.12.

I. Parker. Absolute powerpoint: Can a software package edit our thoughts?, 2001. URL https://www.newyorker.com/magazine/2001/05/28/absolute-powerpoint.

V. Perera. *Language-Based Bidirectional Human and Robot Interaction Learning for Mobile Service Robots*. PhD thesis, Carnegie Mellon University, 2018. URL http://www.cs.cmu.edu/~mmv/papers/PereraThesis.pdf.

R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

RTE, Aug. 2019.
https://opendata.reseaux-energies.fr/explore/embed/dataset/eco2mix-regional-cons-def/
table/?disjunctive.libelle_region&disjunctive.nature&sort=-date_heure&dataChart=
eyJxdWVyaWVzIjpbeyJjaGFydHMiOlt7InR5cGUiOiJsaW5lIiwiZnVuYyI6IlNVTSIsInlBeGlzIj-
oiY29uc29tbWF0aW9uIiwiY29sb3IiOiIjZWE1MjU0Iiwic2NpZW50aWZpY0Rpc3BsYXkiOnRyd-
WV9XSwieEF4aXMiOiJkYXRlX2hldXJlIiwibWF4cG9pbnRzIjoyMDAsInRpbWVzY2FsZSI6Im1-
pbnV0ZSIsInNvcnQiOiIiLCJjb25maWciOnsiZGF0YXNldCI6ImVjbzJtaXgtcmVnaW9uYWwtY29-
ucy1kZWYiLCJvcHRpb25zIjp7ImRpc2p1bmN0aXZlLmxpYmVsbGVfcmVnaW9uIjp0cnVlLCJka-
XNqdW5jdGl2ZS5uYXR1cmUiOnRydWV9fX1dLCJ0aW1lc2NhbGUiOiIiLCJkaXNwbGF5TGVn-
ZW5kIjp0cnVlLCJhbGlnbk1vbnRoIjp0cnVlfQ%3D%3D .

F. Sha and F. Pereira. Shallow parsing with conditional random fields. volume 1 of *HLT-NAACL '03*, pages 134–141. Association for Computational Linguistics, 2003. doi: 10.3115/1073445.1073473.

C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012. doi: 10.1561/2200000013.

D. R. T. Kollar, S. Tellex and N. Roy. Grounding verbs of motion in natural language commands to robots. *Experimental robotics*, page 31–47, 2014.

P. Vittorio, S. Robin, and et.al. Learning task knowledge from dialog and web access. *Robotics*, pages 223–252, 2015.