Expert Systems With Applications 211 (2023) 118364

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Commentary generation for financial markets

Di Zhu [a], Theodoros Lappas [b,*], Thami Rachidi [c]

[a] School of Business, Stevens Institute of Technology, Hoboken, NJ, USA
[b] Department of Marketing and Communications, Athens University of Economics and Business, Athens, Greece
[c] Jefferies Group New York, United States

ARTICLE INFO

Keywords:
NLP
NLG
Text mining
Summarization
Financial markets

ABSTRACT

Financial markets are based on the daily movements of thousands of tradable assets, such as stocks, resulting in billion-dollar trade volumes and affecting investors and companies around the globe. In this volatile and high-stakes environment, financial-service firms employ analysts to create compact market commentaries that serve as insightful summaries with key pieces of information. In this work, we attempt to automate this process by formally defining and algorithmically solving the Market Commentary Generation (MCG) problem. In addition to saving time and cost via automation, our approach makes a number of contributions that differentiate it from previous related work. These include the consideration of thousands of underlying time series, the ability to capture and encode significant market events that involve multiple financial entities, and the ability to deliver high quality commentary even in the presence of small and unlabeled historical datasets. Finally, our approach takes into account the strict compliance requirements of the finance domain, which prevent the use of black-box methods that can produce language that violates key rules and regulations. We compare our work against competitive baselines via an evaluation that includes both qualitative and quantitative experiments.

1. Introduction

On a daily basis, analysts from the Sales and Trading departments of financial service firms compose short market commentary emails and distribute them to their clients (Morris et al., 2007). These emails provide a compact summary of the trading day by covering key statistics, trends, and interesting phenomena. For an analyst, creating a piece of daily market commentary requires them to investigate various numerical or graphical tools and user interfaces. Examples include the Bloomberg Terminal,[1] self-defined Excel macros, and other internal visualization tools.

The following are a few examples of daily commentaries written by a financial analyst for three different days of the market:

> ***Tuesday, August 1, 2017 10:30 AM***
>
> *US Market volumes up 10% making today inline with YTD market average volumes. ETF's make up 18% of the tape thus far. Auto's the only subsector with a 3 std volume move.*
>
> ***Wednesday, July 19, 2017 10:57 AM***
>
> *Market Volumes are down 1% versus yesterday at this time, based on this we will not break 6bn shares. Basically, marking close to 2 weeks now where we did not come close to YTD market avg volumes. ETF's actually lack today only comprising 15% of the tape vs 21% YTD avg.*
>
> ***Tuesday, June 20, 2017 11:26 AM***
>
> *US Markets tracking just an average 2017 day, up 8% d/d and looking 6.8B shares. Telco along with energy and biotech leading the volume charge while Utilities, and Food/Stpls retail are the worst Sp5 subsectors by volume. comprising 15% of the tape vs 21% YTD avg.*

These examples, taken from a dataset that we describe in detail in Section 4, illustrate the unique nature of this type of market commentary. They provide evidence of domain-specific phrases and other linguistic constructs that are unlikely to be found in other types of text. For instance, "YTD" refers to the year-to-day value (i.e. the value exactly 1 year ago) of a certain measurable entity. Similarly, "d/d" compares today's with yesterday's value (day-to-day). The term "tape" refers to the total transaction volume across the market. Terms like "Telco", "energy" and "biotech" are names of major subsectors of assets traded in the stock market.

After studying the market's evolution throughout the day, the analyst selects specific entities, such as individual stocks or entire subsectors, to include in the commentary. What makes the selected entities

[*] Corresponding author.
E-mail addresses: dzhu1@stevens.edu (D. Zhu), ted@aueb.gr (T. Lappas), trachidi@jefferies.com (T. Rachidi).

[1] https://www.bloomberg.com/professional/solution/bloomberg-terminal/

https://doi.org/10.1016/j.eswa.2022.118364
Received 27 July 2021; Received in revised form 13 July 2022; Accepted 1 August 2022
Available online 17 August 2022
0957-4174/© 2022 Elsevier Ltd. All rights reserved.

worthy of consideration is their nature as *outliers*: their significant differentiation from their "expected" or "normal" evolution pattern. The definition of "normalcy" is context-specific, and identifying the appropriate baseline context is critical. For example, the normalcy baseline can be the entity's own short-term (e.g. yesterday), medium-term (past few weeks) or long-term (year-to-date) history. A baseline can also be based on the evolution of other related entities, as in the case of comparing a stock with stocks from the same subsector, or the case of comparing an entire subsector with other subsectors.

These examples illustrate the two objectives of an effective solution for the automatic generation of market commentary:

> **Objective 1, Information Selection**: *An effective solution for market-commentary generation has to monitor various market entities (e.g. sectors, tradeable assets) across time and select the information to be included in the daily commentary.*

While Objective 1 focuses on the *information* included in the commentary, Objective 2 focuses on the *presentation* of this information:

> **Objective 2, Linguistic Compliance**: *An effective solution for market-commentary generation has to produce compliant commentaries with a configurable and predictable linguistic diversity that is consistent with the language used by expert analysts.*

In a Natural Language Generation context, "*configurable and predictable linguistic diversity*" implies that an acceptable commentary cannot just use any type of language to encode the selected information, even if that language appears semantically and syntactically correct. Instead, due to the rigorous compliance requirements of this type of client-visible financial content, the language used has to be constrained to a set of linguistic constructs (e.g. phrases, regular expressions) from a pool of approved compliant candidates. This compliance mandate is exemplified by the following excerpt from the Electronic Code of Federal Regulations (eCFR) §230.156[2]:

It is unlawful for any person, directly or indirectly, by the use of any means or instrumentality of interstate commerce or of the mails, to use sales literature which is materially misleading in connection with the offer or sale of securities issued by an investment company. Under these provisions, sales literature is materially misleading if it: (1) Contains an untrue statement of a material fact or (2) omits to state a material fact necessary in order to make a statement made, in the light of the circumstances of its use, not misleading.

The Financial Industry Regulatory Authority (FINRA)[3] and the Municipal Securities Rulemaking Board (MSRB)[4] also stipulate similar investor-protection rules.

The goal of this work is thus to develop a data-to-text solution that satisfies the *Information Selection* and *Linguistic Compliance* objectives. We discuss and formalize both objectives in Section 3, where we combine them toward the formal definition of the `Market Commentary Generation` problem (`MCG`).

Although `MCG` is indeed an instance of the broader data-to-text language generation paradigm, its unique characteristics make it different from the problems tackled by previous work on data-to-text methods, which are becoming increasingly popular for applications such as weather forecasts (Angeli et al., 2010; Belz, 2007), healthcare (Banaee et al., 2013; Portet et al., 2009), and sports (Liang et al., 2009).

A recent stream of literature has led the way in the use of data-to-text techniques for automatically generating commentary for financial markets (Aoki et al., 2021, 2018; Hamazono et al., 2020; Murakami et al., 2017; Uehara et al., 2020). Almost all of these works are focused on generating commentary for a single time series, rather than for an entire market with thousands of financial entities. Even though recent work has extended this to cover multiple (but still far less) entities (Aoki et al., 2021), the generated sentences still focus on the movement of individual entities, rather than considering multi-entity context as we do in our work. It also introduces the need for manually annotated data, which does not scale in our market-wide context. We discuss all related work in detail in Section 2.

In summary, our work introduces a new algorithmic solution for generating informative and linguistically compliant commentaries that can be effectively used to summarize the underlying data in an entire financial market. Our work extends the related literature via the following contributions:

- Our approach can address thousands of underlying time series, without the need for any type of annotated training data. As discuss in detail in Section 2, these two characteristics differentiate our method from previous work, which tends to either (i) focus one a single (or a very small number of) time series or (ii) adopt a supervised approach that assumes the existence of a manually-annotated ground truth dataset.
- Commentaries generated by existing methods focus on capturing and expressing the isolated movements of individual financial entities, such as the upward or downward movements of stocks. Our approach extends this by looking for interesting events that include multiple entities. Examples of such events include significant changes in the ranking of an entity among other peer entities or in the coverage of an entire multi-entity sector achieved by a specific entity. We discuss this in detail in Section 5.1
- Contrary to recent related works based on neural architectures, our approach allows the practitioner to fully control the linguistic structure of the generated commentary via the use of automatically generated templates. The predictability of the produced language is a frequent prerequisite in the finance domain, where market commentaries have to be compliant with strict regulations. Our template-based approach delivers this predictability while also ensuring linguistic diversity via the use of multiple alternative templates for each comment-worthy event type. We discuss this in detail in Section 5.2.
- As we demonstrate in our experiments in Section 6, our method can perform even in the presence of relatively small (and unlabeled) datasets of past market commentaries. This is in contrast to recent related work on neural models, which requires large volumes of data to generate quality commentary.

2. Background and related work

The recent literature is rich with expert systems that generate text-based summaries for different types of applications. The nature of these applications is highly diverse, as evidenced by works tackling data-to-text (Moryossef et al., 2019), text-to-text (Lappas et al., 2012; Oya et al., 2014; Zhan et al., 2009), and even image-to-text (Ilinykh et al., 2018; Xu et al., 2015) problems. Our work falls under the data-to-text paradigm, for which the relevant literature can be classified into three streams: Natural Language Generation (NLG), Linguistic Data Description(LDD), and End-to-End (E2E) text generation.

2.1. Natural language generation methods

Research on Natural Language Generation (NLG) goes as back as the 1980s (Kittredge et al., 1986). Data-to-text NLG is defined as the task of automatically generating natural language to describe (typically) numerical data. According to the most widely-cited relevant work (Reiter & Dale, 1997, 2000), an applied NLG system contains three typical stages: (i) Text Planning — selecting what information to

[2] https://www.ecfr.gov/cgi-bin/text-idx?node=17:3.0.1.1.12&rgn=div5#se17.3.230_1156

[3] https://www.finra.org/rules-guidance/rulebooks/finra-rules/2210

[4] http://www.msrb.org/layouts/Disclaimer.aspx?url=Rules-and-Interpretations/MSRB-Rules/General/Rule-G-21.aspx

be conveyed, and determining the presentation order of the selected information to readers; (ii) Sentence Planning — deciding how to group multiple messages into one sentence, and finding the proper words and phrases to express the messages; and (iii) Linguistic realization — nesting all words and phrases into well-formed sentences. Based on this architecture, NLG applications have been applied to a diverse spectrum of industries and applications. Examples include BT-Nurse (Hunter et al., 2012), an auto-report system that describes nursing shift summaries from patient electronic records, TEMSIS (Busemann & Horacek, 1997), an air-quality reports generator, and SumTime-Turbine (Yu et al., 2007), a prototype system which creates textual summaries of sensor data from gas turbines. Recent work has refined NLG frameworks by leveraging statistical training methods that require a significant amount of historical data. Kondadadi et al. (2013) combined Sentence Planning and Linguistic Realization into one single process by building contents and template rankers. Gardent and Perez-Beltrachini (2017) integrated a small handwritten grammar, a statistical hypertagger for the Sentence Planning stage.

2.2. Linguistic-Data Description Methods

Compared to NLG, Linguistic Data Description (LDD) is a younger field that emerged in the late 1990s with the idea of performing computation on words using fuzzy logic (Zadeh, 1999). Although LDD also deals with the generation of linguistic summaries for numerical data, it mainly focuses on subjectively expressing the human perception of numerical quantities using fuzzy quantified propositions (Boran et al., 2016). Thus, LDD approaches include basic elements of fuzzy sets (e.g., "very cold", "cold", "mild", "warm", "hot" for "temperature") and fuzzy quantifiers (e.g., "a few", "most", "several", "about ten", etc. for both absolute and relative quantifiers) (Ramos-Soto et al., 2016). Applications based on LDD can be found, among others, in utility consumption reports (van der Heide & Triviño, 2009) and road traffic live descriptions (Alvarez-Alvarez et al., 2012). Contrary to NLG, LDD does not have a high-level architecture and is instead based on target-specific or attribute-specific linguistic constructs. A characteristic example is the Granular Linguistic Model of a Phenomenon (GLMP) approach, which aims at describing phenomena at different levels of granularity (Trivino & Sugeno, 2013). Using LDD-only methodologies to fulfill practical needs across industries is often problematic, as these solutions lacks the richness and completeness of generated text that emulates realistic natural language (Ramos-Soto et al., 2016). To address this, recent work has started to embed LDD into NLG systems. For instance, Conde-Clemente et al. (2018) proposed an architecture to generate linguistic advice about the energy consumption behavior at households. In the same direction, Ramos-Soto et al. (2014) presented an application that automatically generates real-time textual short-term weather forecasts.

2.3. End-to-End language generation methods

The End-to-End (E2E) language generation approach emerged in the 2010s and has greatly benefited from the recent rapid growth of deep learning methods. E2E creates text from input data in a single generation step. It requires a training process on data-text pairs. Recent examples include deep neural sequence-to-sequence models (Sutskever et al., 2014), equipped with an encoder–decoder (Cho et al., 2014) and attention mechanisms (Bahdanau et al., 2014). Recent relevant work tackles the generation of restaurant references based on structured *Meaning Representations* (MRs). An example is shown in Fig. 1 (Dušek et al., 2020). Nevertheless, neural-based text generation often manifests *hallucinations*: cases where the generated summaries are entirely irrelevant from the input data (Rohrbach et al., 2018). This makes the approach a poor fit for applications where faithful representations are critical, and therefore such irrelevant occurrences are unacceptable. The financial decision-making scenario that we tackle in this work is an example of such an application.

MR	*name[The Wrestlers], priceRange[cheap], customerRating[1 of 5]*
reference	The Wrestlers offers competitive prices, but isn't rated highly by customers.

Fig. 1. Example pair of an MR and a corresponding textual description.

2.3.1. Textual summaries of financial time-series data

We further focus our literature review to look specifically for E2E text-generation methods based on time-series data from financial markets, which is also the focus of our own work.

In their seminal and highly-cited work, Murakami et al. (2017) proposed an encoder–decoder model for generating a market comment, trained on news headlines on the time series of the stock price of the Nikkei 225 index. The model is designed to take into account both the long-term and short-term longitudinal effects in the time series, as well the timing of the training headlines. While our own focus is on generating commentary for an entire financial market that includes thousands of time series, their approach focuses on a single time series and does not generate any commentary that covers comparisons, rankings, or any other types of association among the behavior of multiple financial entities. In addition, the employed neural architecture can generate realistic but also unpredictable text, thus jeopardizing the need for linguistic compliance that is a key deliverable for our MSG problem.

The work by Murakami et al. (2017) has inspired a number of follow-up works on data-to-text for financial markets. Hamazono et al. (2020) acknowledge and address the issue of noisy temporal alignments between the input time series and the available training documents. This is a common problem in practice, as it is hard to find training commentaries that perfectly align with the timing of the movements in the underlying data. The authors address this via an extended multi-timestep architecture that treats the actual timing of an event in the underlying data as a hidden variable, and uses multiple input vectors to enable the learning of the data-text correspondence even in the presence of misalignment in the training corpus.

Aoki et al. (2018) extended the encoder–decoder architecture of Murakami et al. (2017) to take into account not only the time series of the focal financial entity (e.g. the Nikkei 225 index) but also external data sources, such as the Dow Jones Industrial Average or the US dollar/Japanese yen exchange rate). The objective was to capture the relationship between the focal entity and such external sources that could affect or provide context on the entity's behavior. The proposed approach thus allows such relationships to be reflected in the generated commentary (e.g. *Nikkei Stock Average opens at a high price after Dow Jones Industrial Average closes at a high price*).

Uehara et al. (2020) observe that previous works tend to generate fluent (syntactically correct) but semantically incorrect sentences that often say the opposite of what is actually happening in the market. For instance, "Nikkei gains" is generated when "Nikkei drops" is appropriate given the movement of the underlying time series data. They address this by templating important terms such as "gain" or "drop" and using them to create rules (e.g. gain vs. loss, high vs. low) that can produce contrastive examples for actual reference commentaries. They then extend the encoder–decoder architecture of Aoki et al. (2018) with loss functions that take into account contrastive examples and train models that correctly distinguish between reference sentences and their contrastive counterparts. While our own work differs significantly both in terms of the focal problem and the solution design, the use of linguistic templates based on the semantic association among terms is a theme that is also prevalent in our own work.

We note that, just like the work of Murakami et al. (2017), the follow-up contributions by Hamazono et al. (2020), Aoki et al. (2018), and Uehara et al. (2020) described above also focus on generating commentary for a single time series.

A recent pioneering work that also adopts the encoder–decoder neural architecture is that by Aoki et al. (2021). In this work, the authors propose a commentary generator to create a daily summary of the Japanese financial market, while being guided by a specific collection of topic labels. The introduction of the labels thus allows the customization of the output and focuses the solution to produce commentaries that can cater to the interests of a specific type of user. While the previous works that we described above focused on a single time series, this work considered 9 financial indicators, guided by a set of 91 human-designed topic labels. The results demonstrate superior results when the labels are used. The main differences from our work are that:

i. While human-generated labels can indeed help control the output of the algorithm, they also introduce the need for a manually annotated training dataset, which is prohibitively expensive to create in the presence of thousands of indicators. While (Aoki et al., 2021) also experiment with an automated way of generating and assigning labels, their results demonstrate this can lead to ambiguous commentaries and sentences that do not include relevant information.
ii. The black-box nature of the neural encoder–decoder architecture, combined with the controlling context of the labels and the consideration of multiple indicators jeopardize the linguistic compliance of the commentaries, especially if automatically generated labels are used to make the approach feasible when applied to large populations of indicators. While this is not necessarily a key consideration for the work by Aoki et al. (2021), compliance is an important requirement and one of the two primary objectives in our context, as we described in the Introduction.
iii. Our approach generates commentaries by monitoring the thousands of indicators (e.g. stocks, ETFs, sectors) in a financial market, rather than focusing on a small subset. Employing the same architecture to this scale would introduce a very large number of parameters and the need for a very large training dataset with thousands of samples. Instead, as we demonstrate in our experiments, our approach can deliver high-quality commentaries with a far smaller number of samples.
iv. While the approach by Aoki et al. (2021) addresses multiple financial time-series, the sentences in the commentaries are still limited to the individual movement of each financial entity and do not address comparisons, rankings or any other multi-entity association (other than parallel/similar movement).

2.4. Evaluation methods for data-to-text solutions

Evaluation plays an important role in improving data-to-text solutions It is also a significant challenge, given that large-scale quantitative evaluations require ground truth data that is typically unavailable. To address this, a common approach is to compare the machine-generated text with one or more human-generated texts using similarity-based metrics (Gatt & Krahmer, 2018). These metrics are typically based on lexical or semantic similarity. Popular metrics include, but not limited to, BLEU (Papineni et al., 2002), RE (Flesch, 1979), Rouge-L (Lin, 2004), and CIDEr (Vedantam et al., 2015).

However, even state-of-the-art similarity-based metrics are insufficient and cannot replace human judgments. Novikova et al. (2017) demonstrated that metric-based evaluations only weakly reflect human judgments of system outputs. In this direction, researchers often recruit human annotators to assign ratings of the readability, accuracy, and usefulness of the linguistic outputs produced by the various methods. Examples can be found in the works by Sripada et al. (2014) and Hunter et al. (2012).

3. Defining the market commentary generation problem

As we discussed in the Introduction, our goal is to address the `Market Commentary Generation` problem (`MCG`). The input to the problem includes a set of market commentaries $\mathcal{D} = \{C_1, C_2, \ldots, C_N\}$ written by expert analysts. Then, given a snapshot M^*, the goal is then to create a commentary that C^* such that:

1. **Objective 1 - Information Selection:** C^* includes information that is likely to be of interest to the audience (i.e. the customers reading the commentaries).
2. **Objective 2 - Linguistic Compliance**: encodes the selected information via compliant linguistic constructs.

In order to formally define the problem, we first need to operationalize these two objectives. The first objective is challenging, due to the subjectivity of what qualifies as interesting information. To address this, we model the first objective as an event-mining problem, calibrated on the expert commentaries included in the given dataset $\mathcal{D}$. An event type is defined as recurring commentary theme that captures a distinct market phenomenon related to one or more financial entities. Each event type thus represents a phenomenon that was consistently evaluated as comment-worthy (interesting) by human experts. We describe event types in detail in Section 5.1, where we also present our algorithm for mining them.

Given a market snapshot M^* we can then address the first objective by using the calibrated event-miner to check if any of the event types mined from $\mathcal{D}$ actually occur in M^*. We can then encode each of these occurrences as text by sampling one of the actual linguistic patterns used in $\mathcal{D}$ to encode events of this type. This effectively addresses the second objective, as it ensures that the language used in the generated commentaries falls within the "safe" space defined by the compliant commentaries in $\mathcal{D}$. We can now formally define the problem as follows.

Problem 1 (*Market Commentary Generation Problem*). Let $\mathcal{D} = \{C_1, C_2, \ldots, C_N\}$ be a dataset of market commentaries authored by experts, where each commentary focuses on a specific market snapshot. In addition, let $\mathcal{V}_D$ be a set of prevalent event types that are discussed consistently in the commentaries of $\mathcal{D}$ and let $\mathcal{L}_V$ be the set of linguistic patterns used in the commentaries to encode each event type $V \in \mathcal{V}_D$. Then, given a market snapshot M^*, we want to automatically produce a synthetic commentary C^* such that:

1. C^* covers all the occurrences of events types from $\mathcal{V}_D$ in M^*
2. C^* encodes the occurrence of each event type $V \in \mathcal{V}_D$ via one of the compliant linguistic patterns from $\mathcal{L}_V$.

In Section 4, we present the Jefferies dataset that serves as the input dataset $\mathcal{D}$ in our instance of the `MCG` problem. Then, in Section 5, we present our complete methodology solving the problem.

4. The Jefferies dataset

Jefferies Financial Group (Jefferies)[5] is a diversified financial services company engaged in investment banking and capital markets, asset management, and direct investing. On a daily basis, analysts from the Sales and Trading department at Jefferies compose short market commentary emails and distribute them to their clients. For our analysis, we obtain 550 such market commentaries generated by a senior analyst at Jefferies, ranging from 2014-08-05 to 2018-06-15. We provide descriptive statistics on this dataset in Table 1 and Fig. 2.

As also evidenced by the example commentaries that we included in the Introduction, commentaries tend to be short (2–3 sentences), with short sentences that lack redundant information and are instead focused on the interesting entities and phenomena that the analyst chose to include the commentary.

[5] https://www.jefferies.com

Fig. 2. Word cloud of the words/phrases in the Jefferies Dataset, with font size proportional to frequency.

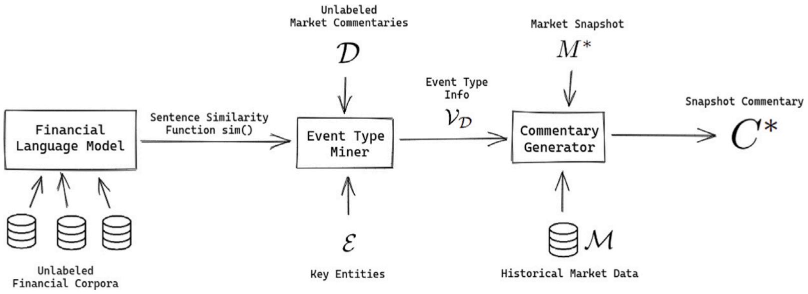

Fig. 3. A flowchart of our methodology for market commentary generation.

Table 1
Descriptive statistics for the Jefferies dataset.

Average	Std	Max	Min
Number of sentences per commentary			
2.85	0.85	7	1
Number of characters per sentence			
167.12	115.04	572	13

5. Methodology

In this Section we describe our methodology for generating market commentary. Fig. 3 presents a flowchart of our 2-step approach. The first step focuses on mining an unlabeled market-commentary dataset $\mathcal{D}$ to identify the prevalent event types that are worthy of being included in a synthetic commentary. The input to this step also includes (i) a sentence-similarity function $sim(\cdot)$, derived via unsupervised learning on various financial corpora, and (ii) a set of key entities $\mathcal{E}$ that are particularly relevant in the financial market context, such as names of stocks and metrics. We discuss this first step in detail in Section 5.1.

The output of the first step is a set of event types $\mathcal{V}_D$. This is provided as input to the second step, which focuses on commentary generation for a given market snapshot M^*. Another input to this step is a standard historical market dataset $\mathcal{M}$, which serves as a record of the financial market of interest. In practice, $\mathcal{M}$ consists of times series that encode the key metrics (e.g. volume, price) of financial entities (e.g. stocks, sectors) across time. We discuss the second step in detail in Section 5.2.

To enhance the clarity of our presentation, we list and describe the various elements of mathematical notation used throughout the description of our methodology in Table 2.

5.1. Mining comment-worthy event types

In this section, we describe the `EventTypeMiner` algorithm for identifying the prevalent types of market events in a given market dataset $\mathcal{D}$.

Table 2
Mathematical notation reference.

Variable	Description
$\mathcal{D} = \{C_1, C_2, \dots, C_N\}$	The input dataset of expert-authored market commentary samples, with each commentary summarizing a specific market snapshot.
C_i	A single expert-authored market commentary
$\mathcal{M}$	Historical dataset of market snapshots that encode the key metrics (e.g. volume, price) of financial entities (e.g. stocks, sectors) across time.
M^*	The focal market snapshot for which we want to generate commentary, as per the definition of the Market Commentary Generation (MCG) Problem.
C^*	The commentary generated automatically by our method for the focal market snapshot M^*.
$\mathcal{V}_D$	The set of prevalent event types that are discussed consistently in the commentaries of $\mathcal{D}$
$V \in \mathcal{V}_D$	A single event type from $V \in \mathcal{V}_D$
$\mathcal{L}_V$	The set of linguistic patterns used in the commentaries of the input dataset to encode a specific event type $V \in \mathcal{V}_D$.
$L \in \mathcal{L}_V$	A single linguistic pattern from $\mathcal{L}_V$.
$\mathcal{E}$	The set of key financial entities discussed in the commentaries (see Table 3.
s	A single sentence from a specific market commentary. Used in the `EventMiner` algorithm.
$\mathcal{G}_D$	A graph that includes 1 node for each sentence s in the input commentary dataset $\mathcal{D}$. Used in the `EventMiner` algorithm.

The input to the algorithm consists of:

- **A commentary dataset** $\mathcal{D}$, as defined in the MGC problem that we defined in Section 3.
- **A sentence-similarity function** $sim(\cdot)$, which can be used to evaluate the similarity between any two sentences from the

commentaries in $\mathcal{D}$. This function will be used as a module by EventTypeMiner, in order to create large and cohesive groups of similar sentences.

- **A set of key entities** $\mathcal{E}$ mined from $\mathcal{D}$. These entities will be used as the focal points of event types mined by EventTypeMiner.

Our framework is compatible with any definition of the sentence-similarity function $sim(\cdot)$ and any definition of the set of entities $\mathcal{E}$. Next, In sections 5.1.1 and 5.1.2 we describe how we obtain these two components in the implementation that we use in our experiments. Then, in Section 5.1.3, we present the pseudocode of the EventTypeMiner algorithm and discuss each of its steps in detail.

5.1.1. *The sentence-similarity function*

In our implementation of the sentence-similarity function $sim(\cdot)$, we first train a language model on a large textual corpus from the broader Finance domain. Our training corpus is a combination of publicly available market-focused articles from multiple related websites: 350,027 articles from Motley Fool,[6] 312,750 from SeekingAlpha,[7] 25,722 from ZeroHedge,[8] and 14,433 entries from Investopedia,[9] for a total of 6.6 GB financial data. We use the Distributed Memory Model (Le & Mikolov, 2014) to map the corpus within a low-dimensionality semantic space, in which similar documents are embedded as neighboring points. The learned model allows us to evaluate the similarity between any two text segments by embedding them and measuring their distance within that space. The primary advantage of this approach is that it goes beyond lexical and into semantic similarity, which allows us to identify similar segments that use different linguistic constructs (words, phrases, syntax) to express similar content (e.g. *Tech leads all sectors in volume today* VS *Technology stocks are the most active so far*). Prior to the training of the model, we use the popular Spacy library[10] to mark noun-chunks and expand our set of tokens beyond single keywords.

5.1.2. *Populating the set of key entities*

To create the set of key entities $\mathcal{E}$, we have to take into account the nature of the trading domain. First, stock markets include a large population of tradable assets. For instance, there were 3119 and 3469 companies listed in the NYSE and NASDAQ stock markets in 2019.[11] The market also includes assets that do not directly represent a specific company, such as Exchange Traded Funds (ETFs). Every asset in the market is represented by two daily time series with *multiple*: one focused on price and one focused on transaction volume. In addition, in order to generate a holistic market commentary, monitoring the price and volume for each individual asset is not sufficient. Instead, the solution also needs to monitor time series at various *aggregate* levels that provide context for individual assets. Examples of such aggregate times series include the total volume of the US market, the total volume of ETFs, and the volume of entire market sub-sectors,[12] such as Energy, Telecommunications, and Utilities. Taking all this into account, we consider the entity types listed in Table 3.

As listed in the table, we use synonym detection to expand the terms included in some entity types. We detect synonyms via a lower bound on the distance between two words within the semantic space of the language model. The value of the bound is calibrated on an actual set of synonyms from WordNet.[13]

5.1.3. *The* EventTypeMiner *Algorithm*

We present the pseudocode of EventTypeMiner in Algorithm 1.

Table 3
Set of key entities provided as input to EventTypeMiner.

Tag	Description
FIN	Includes financial sector and subsector names (e.g. *Energy, Technology*), trading codes (e.g. *AAPL, SPY*), company names, and the terms *Market, US Market*, which represent the entirety of the market and consistently appear in the opening almost all the commentaries in our dataset.
FINTYPE	Includes keywords used to describe the various FIN types (e.g. *sector, subsector, stock, ETF*).
METRIC	Includes the different types of metrics that are interesting in a financial market context. In our experiments, we focus on *volume* and *price*. We also use our finance-trained language model to identify domain-specific synonyms of these words, such as the word *tape* which is often used as an alias of *volume* in this space.
UNIT	Includes the units used to measure METRICs (e.g. *USD, dollars* for price *shares* for volume.
PERC	Includes all percentage instances that appear in the text.
NUM	Includes all non-percentage numeric instances that appear in the text.
TIME	Includes n-grams that encode a point in time or timeframe (e.g. *yesterday, last year, today, last 4 years, d/d*).
DIR/POS	Includes n-grams that encode the direction or position of A FIN (e.g. *up, down, above, below, top, bottom, first, last , fourth*).

Algorithm 1: EventTypeMiner

```
1  Input: Dataset $\mathcal{D} = \{C_1, C_2, ..., C_N\}$, Set of Entities $\mathcal{E}$,
     sentence-similarity function $sim(\cdot)$.
2  Output: A segmentation of all sentences from $\mathcal{D}$ into
     non-overlapping groups, such that each group represents a
     distinct event type covered in dataset's market commentaries.

3  Create an empty Graph $\mathcal{G}_D$.
4  for every commentary $C_i$ covered in $\mathcal{D}$ do
5      Split $C_i$ into a list of sentences $S_i$.
6      for every sentence $s$ in $S_i$ do
7          Transform $s$ by replacing all occurrences of entities from
             $\mathcal{E}$ in $s$ with their corresponding Tags (see Table 4).
8          Add the transformed $s$ as a node to graph $\mathcal{G}_D$.
9      end
10 end
11
12 Set $pw = \{sim(s, s') : \forall s, s' \in \mathcal{G}_D\}$ //All pairwise sentence
     similarities
13 Set $LB = avg(pw) + 2 \times stdev(pw)$ //Lower similarity bound
14
15 for Every pair of nodes $s, s'$ in $\mathcal{G}_D$ do
16     if $sim(s, s') \geq LB$ then
17         Add an undirected edge $(s, s')$ to $\mathcal{G}_D$
18     end
19 end
20
21 Compute the set $\mathcal{V}_D$ of all non-overlapping maximal cliques of
     $\mathcal{G}_D$.
22 Return the set of cliques $\mathcal{V}_D$.
```

In lines 3–10, the algorithm creates a graph $\mathcal{G}_D$ that includes one node for each of the sentences that appear across the commentaries in the given dataset $\mathcal{D}$. Each sentence is first transformed by replacing all

[6] https://www.fool.com/
[7] https://seekingalpha.com/
[8] https://www.zerohedge.com/
[9] https://www.investopedia.com/financial-term-dictionary-4769738
[10] https://spacy.io/usage/linguistic-features#noun-chunks
[11] https://www.nasdaq.com/screening/companies-by-industry.aspx?exchange=NASDAQ
[12] https://eresearch.fidelity.com/eresearch/markets_sectors/sectors/sectors_in_market.jhtml
[13] https://wordnet.princeton.edu/

Table 4
Event Types mined by EventTypeMiner from the Jefferies dataset.

Movement (occurs in 98% of all commentaries)	
Description	A [METRIC] value of a [FIN] moves with respect to a past baseline [TIME]. The move is optionally described via a [DIR/POS] and quantified via a [NUM] or [PERC].
Example 1	[FIN:US Market] [METRIC:Volumes] [DIR/POS:up] [PERC:25%] vs [TIME:yesterday].
Example 2	[FIN:Market] [METRIC:volumes] [DIR/POS:up] [PERC:9%] [TIME:d/d] and tracking [NUM:6.5B].
Self-Ranking (25.2%)	
Description	A [METRIC] value of a [FIN] moves to position ([DIR/POS]) within a [TIME] baseline. The move is described via a [DIR/POS] and quantified via a [NUM] or [PERC], with an optional [UNIT] mention.
Example 1	[FIN:US Market] [METRIC:volumes] are [DIR/POS:down] [PERC:30%], [DIR/POS: one of the lowest] [METRIC: volume] [TIME:days of the year].
Example 2	[FIN:US Market] [METRIC:volumes] tracking [NUM:5.3B] [UNIT:shares], which is [DIR/POS:the 2nd slowest day of the year].
Coverage (62.9%)	
Description	One or more [FIN]s cover a certain [PERC] of a [METRIC] (either across the market or for a specific [FINTYPE]).
Example 1:	[FIN:ETFs] are higher today at [PERC:22%] of total [METRIC:volume].
Example 2	[FIN:FB] alone is [PERC:3%] of the [METRIC: tape].
Group-Ranking (41.2%)	
Description	A [METRIC] value of a [FIN] moves to certain position ([DIR/POS]) within a [FINTYPE]. The move is optionally be described via a [DIR/POS] and quantified via a [NUM] or [PERC], with an optional [UNIT] mention.
Example 1	[FIN:Retail] is [DIR/POS: the second lowest] [FINTYPE:subsector] by [METRIC:volume].
Example 2	[FIN:Energy] [DIR/POS:leads all] [FINTYPE:subsectors] in [METRIC:volume] and [METRIC:price].

instances of known entities from $\mathcal{E}$ with their corresponding Tags, as listed in the first column of Table 4.

In lines 12–13, the algorithm first creates the set of all pairwise similarities for all sentences. It then computes a lower bound as two standard deviations above the average similarity.

In lines 15–19, the algorithm creates an edge between any two sentences whose similarity is over the lower bound. We calibrate the lower bound on a set of 200 pairs of sentences, in which 100 were manually verified to contain two semantically similar sentences and 100 to contain semantically dissimilar sentences.

In lines 21–22, the algorithm computes and returns the set of all non-overlapping maximal cliques of $\mathcal{G}_D$. This is done by iteratively computing the largest clique in the graph and then removing the nodes (sentences) from that clique (Rossi et al., 2014).

In Table 4, we present the 4 event types mined by EventTypeMiner from the market data included in the Jefferies dataset.

5.2. Commentary generation

We present the process of generating synthetic commentary for a given market snapshot in Algorithm 2. The input to the algorithm consists of:

- A Historical Market Dataset $\mathcal{M}$, including key values (price, volume) of different financial entities (e.g. stocks, ETFs) across time.
- The Market Snapshot M^* for which we want to generate the commentary.
- A Set of Event Types $\mathcal{V}_D$ mined by EventTypeMiner. Note that each event type $V \in \mathcal{V}_D$ is essentially a cluster (maximal clique) of all the linguistic patterns $\mathcal{L}_V$ that can be used to encode events of this type.

In line 4, the algorithm starts by creating an empty commentary. This is then gradually extended to encode important events related to the given market snapshot M^*.

In line 5, the algorithm iterates over each event type V returned by EventTypeMiner.

In line 6, the algorithm mines the given snapshot M^* to detect the set V_{M^*} of events of type V. We describe this process in detail in Section 5.2.1.

In lines 7–11, the algorithm iterates over the events in V_{M^*}. For each such event e, we probe $\mathcal{L}_V$ to sample a linguistic pattern that matches e. We then create a new sentence by replacing the pattern's wildcards with the parameters of e. We describe this process in detail in Section 5.2.2. Finally, the sentence is appended to the commentary C^* in line 10.

5.2.1. Identifying event occurrences

The parameterized nature of the event types produced by EventTypeMiner (see examples in Table 4) allows us to translate event-detection into a standard task of outlier detection in a time series, simply by monitoring the corresponding series type. In this setting, any of the numerous methods in the rich time series literature is applicable (Ljung, 1993). The detection process for all event types is thus based on:

- Creating a time series based on the event type's parameters.
- Monitoring the created time series for outliers.

Next, we describe the detection process that we use to identify events for each of the 4 types returned by EventTypeMiner on the Jefferies Dataset.

Algorithm 2: Commentary Generator

```
1  Input: Historical Market Dataset $\mathcal{M}$, Market Snapshot $M^*$, Set
   of Event Types $\mathcal{V}_D$, Set of linguistic patterns $\mathcal{L}_V$ that can be
   used to encode each event type $V \in \mathcal{V}_D$.
2  Output: A synthetic commentary $C^*$ that covers all occurrences
   of events types from $\mathcal{V}_D$ in $M^*$ via compliant linguistic patterns
   from $\mathcal{L}_V$.
3
4  Start an empty commentary $C^*$.
5  for every Event Type $V \in \mathcal{V}_D$ do
6      $V_M$ =detect($V, \mathcal{M}, M^*$)
7      for every Event Occurrence $e \in V_M$ do
8          Sample a matching linguistic pattern $L$ from $\mathcal{L}_V$.
9          Parameterize $L$ based on $e$ to create a new sentence
           $s$.
10         Append $s$ to $C^*$.
11     end
12 end
13 Return $C^*$
```

Movement: Movement events can be detected by monitoring the (METRIC) time series for each FIN and identifying outliers that significantly deviate from the expected baseline value. We can consider the baseline as the average over any timeframe (TIME). The extremeness of the outlier can then be measured via the extent of the deviation from the baseline, which can be expressed either in absolute value (NUM) or as a percentage (PERC). The direction (DIR/POS) of the outlier is also based on whether it is above or below the baseline.

Self-Ranking: The detection of Self-Ranking events is very similar to Movement events, with the only difference being that, rather than monitoring the value of a METRIC across time, we focus on the time series that encodes the relative position (DIR/POS) of each FIN entity's current value with respect to all other values within the baseline timeframe (TIME).

Coverage: For Coverage events, detection focuses on the time series that encodes the METRIC coverage (PERC) offered by a FIN to a baseline group. This group can be generically defined to include all the entities in the market or it can focus on a specific group of related entities (FINTYPE), such as those in the same sector. For example, if a FIN consistently covers a very low percentage (PERC) of the trading volume (METRIC) of a specific sector, then a sudden spike in that coverage would qualify as an event.

Group-Ranking: Finally, for Group-Ranking events, detection focuses on the time series that encodes a FIN's position (DIR/POS) within a group (i.e. the entire market or a specific FINTYPE). For example, if an entity is consistently ranked first within a sector, a significant drop to a far lower position would be detected as an event.

5.2.2. Sampling and parameterizing linguistic patterns

As per the definition of the MCG problem, every occurrence of a specific event type V has to be represented in the generated commentary via a compliant linguistic pattern from a set of approved candidates $\mathcal{L}_V$. Conveniently, as described in detail in Section 5.1.3, EventTypeMiner already captures event types as clusters of similar linguistic patterns. Table 4 illustrates how each pattern is essentially a sentence template, parameterized via wildcards that represent different types of entities (listed in Table 3).

Once an event e of type V has been identified as described in Section 5.2.1, we can then simply probe the cluster of linguistic patterns $\mathcal{L}_V$ (exactly as returned by EventTypeMiner), randomly sample a pattern, and convert it into a sentence by replacing its wildcards with the corresponding parameters of e.

For example, suppose that the outlier-detection process identifies a significant 20% spike in the volume of the AAPL stock compared to yesterday's value. In this case, the 5 parameters of the event are: [FIN:AAPL], [METRIC:volume], [DIR/POS:up], [PERC:20%], [TIME:yesterday]. We can then sample a linguistic pattern that matches these 5 parameters for the **Movement** clusters (e.g. see Example 1 in Table 4) and create the sentence: *AAPL volume up 20% vs yesterday*. We present three examples of human-authored commentaries from the Jefferies dataset and their algorithm-produced counterparts in Table 5.

In practice, one could optionally apply various automatic or manual filters to eliminate certain linguistic patterns from the clusters reported by EventTypeMiner. We do not apply any such filters in our experiments. Instead, we tune the sampling process in order to select patterns with a probability proportional to the number of times they actually appear in the available dataset of market commentaries $\mathcal{D}$.

6. Evaluation

In this Section, we present the experiments that we conducted to evaluate the proposed methodology.

6.1. Comparative user study

First, we conduct a user study to compare our method with a baseline from the relevant literature.

Table 5
Commentary Examples, Human Expert Vs Algorithm.

Source	Commentary
Human Expert	Market Volumes coming out down 33% verse Friday. ETF's make up 24% of the tape.
Algorithm	US market volumes down 33%. It is tied for one of the slowest days of this month. ETF is 24% of the total mkt volume. Telecommunication Services sector is having a 3 std dev down move.
Human Expert	US Market volumes off 11%. It would be the 3rd slowest day of year. BABA is making up 1.5% of all volume. Biotech and Software names driving the volumes today while Food/Beverage, Mats and Banks all seeing 30% decline in volumes.
Algorithm	Volumes are off 8% d/d. ETF volume has avg @ around 23% of total. Consumer Staples, Materials and Banks are the worst sectors.
Human Expert	US Market volumes pick up a bit of steam. Volumes up 9% d/d. As expected VZ/AKS etc. have Telco and Materials leading the volume. Transportation and Retail are the weakest volume SP5 subsectors.
Algorithm	Market volumes are up 9% vs yesterday. ETFs an inline 17% of shares traded. Materials/Telecommunication Services sector leading the volume with a 3 st dev move. VZ along is 27% of the Telecommunication Services.

Baseline Selection: We considered all of the data-to-text methods designed for financial time series as potential baselines. The seminal work by Murakami et al. (2017) was not selected because, even though it has inspired many follow-up works, it has also been extended and outperformed by them. The work by Hamazono et al. (2020) was not selected due to its focus on temporal alignment, which is a concern for finer temporal granularities (e.g. "live" commentaries generated every few minutes or hours) and not an impactful consideration when creating daily market commentaries as we do in our work. The work by Aoki et al. (2021) was excluded due to its need for manually annotated data, which is not tractable in a setting with thousands of times series, such as ours. The work of Aoki et al. (2018) was not selected as it has been extended and outperformed by Uehara et al. (2020), which is the one that we adopt as a baseline in our experiments. As external reference data sources, we use the Dow Jones Industrial Average (also used in the original paper) and the S&P 500, arguably the most commonly followed equity index in the US market. As we discussed in Section 2.3.1, previous works, including (Uehara et al., 2020), are designed to create commentary for a single time series, rather than for an entire market. Concatenating the thousands of individual commentaries would create an unreasonably large market-wide commentary that would not be fit for purpose.

To address this, we choose to use the baseline to generate and concatenate commentaries only for the financial entities that were actually included in the human-generated commentaries for the corresponding days from the Jefferies dataset. In order to have a fair comparison, we also focus our own method on the same entities. We note that, while we take this step to enable the use of the baseline, it is important to acknowledge that this type of guidance would not be available in a practical scenario where the goal is to generate a market-wide commentary without already having access to one. Therefore, we also present our results without this type of guidance.

Experimental Design: First, we randomly sampled 50 days from the Jefferies dataset and retrieve the following four commentaries for each day:

1. The commentary written by the human analyst.
2. The encoder–decoder approach by Uehara et al. (2020), guided to produce commentary only for the financial entities that were actually included in the human-generated commentaries for the corresponding days (as described above).

Table 6
User Study results fluency, correctness, informativeness.

Approach	Fluency		Correctness		Informativeness	
	Yes	No	Yes	No	Yes	No
Human	49	1	174	14	159	12
Ours (guided)	46	2	166	17	144	19
(Uehara et al., 2020) (guided)	31	17	112	37	81	22
Ours (unguided)	47	2	168	14	149	12

3. Our own approach, similarly guided to only consider the financial entities in the human analyst's commentary.
4. Our own approach without any guidance, applied on the entire market.

Note that an unguided version of the baseline is not possible, as it would create a commentary for each of the thousands of time series in the market, leading to a massive concatenated super-document.

Our design leads to a total of 200 commentaries. Following previous work, we then showed each of the 200 commentaries to 5 expert analysts and asked them to assign three scores: correctness, informativeness, and fluency (Aoki et al., 2018; Murakami et al., 2017; Uehara et al., 2020). For fluency, the analysts were asked to rate the commentaries in terms of readability, regardless of their content. Following previous work, we adopt a binary label (fluent, not fluent) for fluency. We only consider commentaries for which at least 4 of the 5 analysts agreed on (non)fluency. To evaluate correctness and informativeness, the analysts used both the generated commentaries and the corresponding time series data. For correctness, if the analyst deemed that a generated commentary *correctly or incorrectly describes an event from the underlying data of the entities included in the commentary*, they would then record that event and assign it to the commentary.

We only consider events that at least 4 of the 5 analysts identified as correct. For informativeness, if the analyst deemed that *a correctly described event is also important and should be included in the commentary*, they would then record that event and assign it to the commentary. Similar to correctness, we only consider events that at least 4 of the 5 analysts identified as informative. We present the results in Table 6.

The first 2 columns of the table show the number of commentaries that were identified as fluent for each method. The third and fourth columns show the number of events that were identified as correct and incorrect, respectively. Finally, the fifth and sixth columns show the number of events that were identified as **correct and informative** and **correct and non-informative**, respectively.

We begin by comparing the guided versions of our approach and the baseline. We observe that our method has an advantage over the baseline in terms of fluency, correctness, and informativeness. The fluency advantage was largely anticipated, given that our approach is based on predictable linguistic patterns, while the baseline employs a neural architecture that is trained on real data but is not entirely predictable. In terms of correctness and informativeness, the baseline is handicapped by the small size of the available dataset (largely due to the data-demanding nature of its neural architecture), while our approach delivers competitive results even with limited training data. As indicated via Fisher's exact test, the difference between our method and the baseline was statistically significant at $p < 0.01$ for fluency and correctness, and significant at $p < 0.05$ for informativeness.

A further study into the events that were identified as both correct and informative for the two approaches, showed an overlap of about 74% (i.e. events that appear in both commentaries) between our method and the human annotator and about 66% between our method and the baseline. This indicates that our method can be used to complement the results of human efforts or of other automated approaches based on neural architectures, such as the baseline by (Uehara et al., 2020). We illustrate this via a Venn diagram in Fig. 4. The diagram encodes the relationship between the market events that are considered comment-worthy (correct and informative) by humans (or the baseline) and those that are determined as comment-worthy by our own approach.

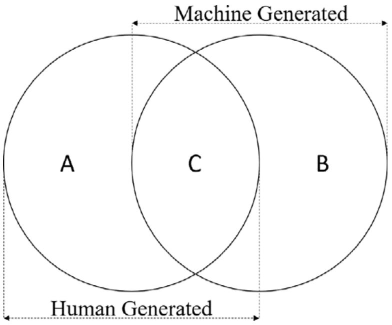

Fig. 4. Content informativeness between human and machine generated text.
A - in human generated text but not in machine generated
B - in machine generated text but not in human generated
C - in both human and machine generated text.

Area *A* includes the set of events reported by humans but not the algorithm. This area represents interesting pieces of information that the algorithm missed. Area *B* includes the set of events reported by algorithm but not by humans. This area is worthy of deeper exploration, as it represents pieces of information that could either be (i) redundant or uninformative and therefore are falsely included by the algorithm or (ii) interesting, comment-worthy pieces of information that were missed by humans but were rightfully included by the algorithm. Finally, Area *B* includes the set of events reported by both humans and the algorithm.

The final line of Table 6 provides additional evidence on the ability of our approach to identify and encode important, comment-worthy phenomena. We observe that, out of the 50 commentaries, 92% were fluent. Out of 183 events, 90% were correct. Finally, out of 163 events, 88% were both correct and informative. This demonstrates that our approach can deliver quality commentary at the market level, without any guidance in terms of the financial entities that it should focus on.

6.2. Evaluation using the Bilingual Evaluation Understudy Score

Next, we describe an experiment based on the popular valuation using the Bilingual Evaluation Understudy Score (BLEU) (Papineni et al., 2002). BLEU is a metric for automatically evaluating machine-translated text. The BLEU score is a number between 0 and 1 that measures the similarity of the machine-translated text to a set of high quality reference translations. A value of 0 means that the machine-translated output has no overlap with the reference translation (low quality) while a value of 1 means there is perfect overlap with the reference translations (high quality).

BLEU is a good fit for machine translation, given that there is a small number of possible reference translations (i.e. a limited number of ways to rewrite a specific sentence in the same language). However, BLEU is less appropriate in the context of financial commentary, due to the much larger and more diverse set of language constructs that one can use to comment on financial time series. While previous work has used BLEU (Aoki et al., 2018; Hamazono et al., 2020; Murakami et al., 2017; Uehara et al., 2020), it has done so in the very limited context of one time series for a single financial entity. Instead, the market-wide commentary that is the focus of our work takes into account thousands of time series and, therefore, the number and diversity of possible commentaries are vast. In addition, the BLEU-evaluated models proposed by previous work were trained on short headlines, rather than on the multi-sentence commentaries in our Jefferies dataset. Taking the above into account we present the BLEU results for:

Table 7
BLEU scores for the four approaches.

Ours (guided)	Uehara et al. (2020) (guided)	Ours (unguided)	Ours (iterative)
0.33	0.22	0.14	0.68

- The 3 methods that we used in the comparative user study described in Section 6.1: the two guided versions of our method and the baseline by Uehara et al. (2020) and the unguided, market-wide version of our method.
- A version of our method that iterates over all the available linguistic patterns returned by `EventTypeMiner`.

We present the results in Table 7, which includes the scores achieved on the 30% of the Jefferies dataset that was used for testing, after training on the remaining 70%.

As anticipated, the unguided version of our algorithm performs poorly, due to the fact that the commentary it produces is likely to differ significantly from the one created by the human analyst on the same day, both in terms of the events covered and of the language used. However, as we discussed in detail in Section 6.1 and is also illustrated in Fig. 4, the events encoded by our method can be just as correct and informative as those included by the human analyst, even if they are in fact different.

The guided version of our algorithm performs better, but still not at a level that is considered high quality,[14]. Again, this is not surprising, as there are a number of different ways to discuss the events in a market. As we showed in Section 6.1 selecting a different phraseology than that used by a specific analyst does not imply that the commentaries are of low quality. The fact that the baseline underperforms can be largely attributed to the small size of the training set (70% of just 550 commentaries). Neural models introduce a number of parameters and therefore require larger numbers of samples to train (such as the one used in the original work). Therefore, while this result does not prove that the baseline is not an effective method, it demonstrates that our approach can be effectively used even in the presence of smaller datasets.

Finally, the results of the iterative version of our algorithm show that the BLEU score can be easily inflated by simply picking linguistic patterns that are the most similar to the ones used in the reference commentary. We consider this to be less of an interesting finding and more of another piece of evidence that BLEU can be a misleading metric for our context. Still, this result demonstrates that, by providing multiple alternative patterns for each event type, our method enables linguistic diversity and can be used to customize the output for different contexts.

6.3. Sensitivity analysis

As we discuss in Section 5.2.1, our approach for identifying event occurrences in market data is based on outlier detection. In this experiment, we perform a sensitivity analysis of this approach for the Movement event type.

Specifically, we create and monitor the price time-series of all SP500 stocks[15] between 2016-05-01 and 2017-05-01. As a baseline, we adopt a moving average window of length n, where n is a parameter that we tune. A day's value is then considered an outlier if it deviates more than z standard deviations from the baseline, where z is the second parameter that we tune. For every (z, n) combination, Fig. 5 reports the percentage of stocks that reported at least one Movement event (i.e. a value that deviates more than z standard deviation above the moving average for a window of length n).

As anticipated, the percentage is sensitive to the z score. When $z = 1$, around 55% to 65% stocks are marked as outliers, regardless of the value for n. Meanwhile, when n is fixed and z increases, the percentage of outlier stocks decreases significantly. The z parameter is thus significantly more influential.

As we discussed in Section 5.2.1, our framework is compatible with any method for detecting outliers in time series. As is also the case with the simple approach that we applied in this experiment, a practitioner trying to select outliers for a specific day can start by looking for extreme outliers (i.e. high values of z) and then gradually relax the extremity threshold until outliers start emerging.

7. Discussion, limitations and future work

In this work, we presented a data-to-text framework for automatically generating informative commentaries based on market data. Our work takes into consideration both informativeness and linguistic compliance, which is a hard constraint for such commentaries given the nature of the data and the effects that these commentaries can have on financial decisions. The proposed approach extends the recent stream of relevant literature by addressing an entire financial market (rather than just a single time series or small set of series), by being able to perform even in the presence of a relatively small training dataset (contrary to the thousands of samples that are understandably needed by previous neural methods), and by generating commentary that goes beyond just the movement of financial entities to consider multi-entity contexts (e.g. rankings, coverage).

In order to identify the different types of comment-worthy market events, we combine established methods for modeling text with a graph-based formalization that allows us to identify and locate events as cliques in a network of sentences. Our application of this approach on a real market dataset reveals four different types of comment-worthy events in market data and describes the nature and parameterization of each type. Our parameterization allows us to formalize the task of identifying occurrences of each event type as a standard problem of outlier-detection in time series, and enables the use of any related method from the rich literature on outlier detection.

The experiments provide promising evidence that our proposed method can be effectively used in practice to mine real market data and automatically extract linguistically compliant commentaries that provide valuable information for customers in the finance domain. This provides a valuable tool for financial firms and entities around the world that want to summarize large longitudinal market data in an intuitive and explainable way. Explainability is an important trait in the finance domain, as is in any setting where the goal is to support critical decisions with real impact on the affected stakeholders. The commentaries generated by our algorithm are explainable by definition, as they consist of text-encoded occurrences of specific and fully parameterized events types. The reader of the commentary can thus select any commentary sentence and trace it back to the type of event that it represents.

While our focus is clearly on commentary generation for financial markets, we design our framework to be applicable to any domain with the following characteristics:

- The goal is to automatically create an informative, text-based summary of a given snapshot from an underlying set of multiple numerical time series.
- The language in the summary cannot be arbitrary. Instead, it is subject to specific constraints that it has to respect. These constraints are either explicitly stated (e.g. a set of rules) or they have to be automatically mined from a dataset of already-compliant (approved) commentaries.
- The produced textual commentaries have to be explainable, so that the existence of every piece of text (e.g. sentence) in the commentary can be justified and attached to an intuitive reason.

[14] https://cloud.google.com/translate/automl/docs/evaluate

[15] https://en.wikipedia.org/wiki/List_of_S%26P_500_companies

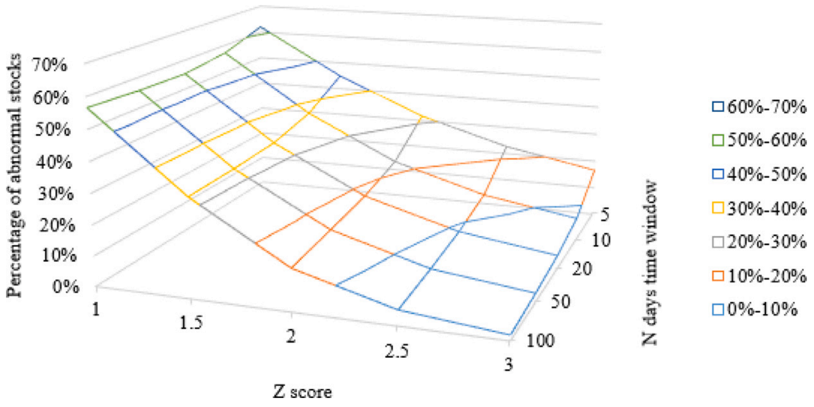

Fig. 5. Percentage of abnormal SP500 stocks price daily measured by moving average.

Domains based on rich sensor data (e.g. IoT streams, environmental sensors) are examples of settings that could benefit from our work. Our methodology can also be applied to either different types of longitudinal financial data, such as the set of time series capturing the evolution of firms (e.g. revenue, growth, debt) or countries (e.g. unemployment, GDP, poverty indicators). Our work can also be applied to spatiotemporal data, by employing an event (outlier) detection method that focuses on spatiotemporal rather than just temporal data.

Limitations and future work: Our methodology and analysis of the Jefferies dataset focused on commentaries authored by a single human expert. In a more generalized context, the input could consist of contributions from multiple experts, each with their own linguistic style and perception of which events should be included in a commentary. A framework that extends our own would then be required to analyze the different perspectives and reconcile their differences toward the identification of prevalent and comment-worthy market events (see Section 5.1). Further, the application of our EventTypeMiner algorithm on the Jefferies dataset revealed four major types of market events: Movement, Self-Ranking, Coverage, and Group Ranking. Future work could lead to the identification of additional types, each with their own nature and parameters.

Our approach to addressing the linguistic compliance objective of the MCG problem is based on using single-sentence text patterns, with each pattern encoding a single occurrence of an event type. In practice, however, human experts can encode multiple event occurrences from one or more types in the same sentence, leading to more complex language. By taking this into account, an improved solution could support more flexible and realistic commentary language, while still maintaining compliance. This flexibility could extend to the EventTypeMiner algorithm (see Section 5.1, which could be improved to better handle sentences that cover multiple event occurrences and to more accurate estimations of the prevalence of each event type in the given data.

CRediT authorship contribution statement

Di Zhu: Data Curation, Conceptualization, Methodology, Software, Writing – review & editing, Validation. **Theodoros Lappas:** Conceptualization, Methodology, Software, Supervision, Writing – review & editing, Validation. **Thami Rachidi:** Conceptualization, Methodology, Supervision, Reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was funded by a fellowship grant by the Jefferies Group, USA. The grant provided full funding for one PhD Student (Di Zhu) from Stevens Institute of Technology for three years.

References

Alvarez-Alvarez, A., Sanchez-Valdes, D., Trivino, G., Sánchez, Á., & Suárez, P. D. (2012). Automatic linguistic report of traffic evolution in roads. *Expert Systems with Applications*, *39*(12), 11293–11302.

Angeli, G., Liang, P., & Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 502–512). Cambridge, MA: Association for Computational Linguistics, URL: https://aclanthology.org/D10-1049.

Aoki, K., Miyazawa, A., Ishigaki, T., Aoki, T., Noji, H., Goshima, K., Takamura, H., Miyao, Y., & Kobayashi, I. (2021). Controlling contents in data-to-document generation with human-designed topic labels. *Computer Speech and Language*, *66*, Article 101154. http://dx.doi.org/10.1016/j.csl.2020.101154.

Aoki, T., Miyazawa, A., Ishigaki, T., Goshima, K., Aoki, K., Kobayashi, I., Takamura, H., & Miyao, Y. (2018). Generating market comments referring to external resources. In *Proceedings of the 11th international conference on natural language generation* (pp. 135–139). Tilburg University, The Netherlands: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W18-6515, URL: https://aclanthology.org/W18-6515.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Banaee, H., Ahmed, M. U., & Loutfi, A. (2013). Towards NLG for physiological data monitoring with body area networks. In *Proceedings of the 14th European workshop on natural language generation* (pp. 193–197). Sofia, Bulgaria: Association for Computational Linguistics, URL: https://aclanthology.org/W13-2127.

Belz, A. (2007). Probabilistic generation of weather forecast texts. In *Human language technologies 2007: the conference of the North American chapter of the association for computational linguistics; proceedings of the main conference* (pp. 164–171). Rochester, New York: Association for Computational Linguistics, URL: https://aclanthology.org/N07-1021.

Boran, F. E., Akay, D., & Yager, R. R. (2016). An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, *61*, 356–377.

Busemann, S., & Horacek, H. (1997). Generating air quality reports from environmental data. In *Proceedings of the DFKI workshop on natural language generation* (pp. 15–21).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Conde-Clemente, P., Alonso, J. M., & Trivino, G. (2018). Toward automatic generation of linguistic advice for saving energy at home. *Soft Computing*, *22*(2), 345–359.

Dušek, O., Novikova, J., & Rieser, V. (2020). Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Computer Speech and Language*, *59*, 123–156.

Flesch, R. (1979). *How to write plain english: a book for lawyers and consumers*. Harper & Row New York, NY.

Gardent, C., & Perez-Beltrachini, L. (2017). A statistical, grammar-based approach to microplanning. *Computational Linguistics*, *43*(1), 1–30.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170.

Hamazono, Y., Uehara, Y., Noji, H., Miyao, Y., Takamura, H., & Kobayashi, I. (2020). Market comment generation from data with noisy alignments. In *Proceedings of the 13th international conference on natural language generation* (pp. 148–157). Dublin, Ireland: Association for Computational Linguistics, URL: https://aclanthology.org/2020.inlg-1.21.

Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S., & Sykes, C. (2012). Automatic generation of natural language nursing shift summaries in neonatal intensive care: BT-nurse. *Artificial Intelligence in Medicine*, *56*(3), 157–172.

Ilinykh, N., Zarrieß, S., & Schlangen, D. (2018). The task matters: Comparing image captioning and task-based dialogical image description. In *Proceedings of 11th international conference on natural language generation*.

Kittredge, R., Polguere, A., & Goldberg, E. (1986). Synthesizing weather forecasts from formatted data. In *Coling 1986 volume 1: the 11th international conference on computational linguistics*.

Kondadadi, R., Howald, B., & Schilder, F. (2013). A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1406–1415).

Lappas, T., Crovella, M., & Terzi, E. (2012). Selecting a characteristic set of reviews. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 832–840). ACM.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196). PMLR.

Liang, P., Jordan, M. I., & Klein, D. (2009). Learning semantic correspondences with less supervision. In *ACL '09, Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP: volume 1 - volume 1* (pp. 91–99). USA: Association for Computational Linguistics.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Ljung, G. M. (1993). On outlier detection in time series. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *55*(2), 559–567.

Morris, M. W., Sheldon, O. J., Ames, D. R., & Young, M. J. (2007). Metaphors and the market: Consequences and preconditions of agent and object metaphors in stock market commentary. *Organizational Behavior and Human Decision Processes*, *102*(2), 174–192.

Moryossef, A., Goldberg, Y., & Dagan, I. (2019). Step-by-step: Separating planning from realization in neural data-to-text generation. arXiv preprint arXiv:1904.03396.

Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H., & Miyao, Y. (2017). Learning to generate market comments from stock prices. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1374–1384). Vancouver, Canada: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P17-1126, URL: https://aclanthology.org/P17-1126.

Novikova, J., Dušek, O., Curry, A. C., & Rieser, V. (2017). Why we need new evaluation metrics for NLG. arXiv preprint arXiv:1707.06875.

Oya, T., Mehdad, Y., Carenini, G., & Ng, R. (2014). A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th international natural language generation conference* (pp. 45–53).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318).

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., & Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, *173*(7), 789–816. http://dx.doi.org/10.1016/j.artint.2008.12.002, URL: https://www.sciencedirect.com/science/article/pii/S0004370208002117.

Ramos-Soto, A., Bugarín, A., & Barro, S. (2016). On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets and Systems*, *285*, 31–51.

Ramos-Soto, A., Bugarin, A. J., Barro, S., & Taboada, J. (2014). Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, *23*(1), 44–57.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87.

Reiter, E., & Dale, R. (2000). *Building natural language generation systems*. Cambridge University Press.

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., & Saenko, K. (2018). Object hallucination in image captioning. arXiv preprint arXiv:1809.02156.

Rossi, R. A., Gleich, D. F., Gebremedhin, A. H., & Patwary, M. M. A. (2014). Fast maximum clique algorithms for large graphs. In *Proceedings of the 23rd international conference on world wide web* (pp. 365–366).

Sripada, S., Burnett, N., Turner, R., Mastin, J., & Evans, D. (2014). A case study: NLG meeting weather industry demand for quality and quantity of textual weather forecasts. In *Proceedings of the 8th international natural language generation conference* (pp. 1–5).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Trivino, G., & Sugeno, M. (2013). Towards linguistic descriptions of phenomena. *International Journal of Approximate Reasoning*, *54*(1), 22–34.

Uehara, Y., Ishigaki, T., Aoki, K., Noji, H., Goshima, K., Kobayashi, I., Takamura, H., & Miyao, Y. (2020). Learning with contrastive examples for data-to-text generation. In *Proceedings of the 28th international conference on computational linguistics* (pp. 2352–2362). Barcelona, Spain (Online): International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.213, URL: https://aclanthology.org/2020.coling-main.213.

van der Heide, A., & Triviño, G. (2009). Automatically generated linguistic summaries of energy consumption data. In *2009 ninth international conference on intelligent systems design and applications* (pp. 553–559). IEEE.

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4566–4575).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, *13*(1), 25–49.

Zadeh, L. A. (1999). Fuzzy logic=computing with words. In *Computing with words in information/intelligent systems, vol. 1* (pp. 3–23). Springer.

Zhan, J., Loh, H. T., & Liu, Y. (2009). Gather customer concerns from online product reviews–A text summarization approach. *Expert Systems with Applications*, *36*(2), 2107–2115.