Online Supplement for "Let the Laser Beam Connect the Dots: Forecasting and Narrating Stock Market Volatility"

Appendix A: Preambles: Recurrent Independent Mechanisms (RIMs)

The forecasting problem formulation naturally needs a sequential model backbone. RIMs (Goyal et al. 2021) is a state-of-the-art architecture built on the assumption that the data has been generated by a set of independent mechanisms. Assuming K RIMs (effectively, K LSTM cells interacting via attention mechanisms), the following four stages occur between the input x_t^{RIM} and RIM k (having hidden state $h_{t,k}$) at time step t.

1. Stage 1 (Producing a query): In this stage, current hidden state, $h_{t,k} \in \mathbb{R}^d$, of any RIM is linearly projected to its' corresponding query. Formally,

$$q_{t,k} = h_{t,k} \times W_k^q \tag{19}$$

where $W_k^q \in \mathbb{R}^{d \times d}$ is a RIM-specific learnable matrix, $k \in \{1, 2...K\}$.

2. Stage 2 (Competition mechanism): The goal of this stage is to dynamically select RIMs relevant to the current input [13], $x_t^{RIM} \in \mathbb{R}^d$. It is possible that none of the RIMs are relevant to the current input, x_t^{RIM}. Hence, firstly, a row of full of zeros ($\emptyset$) is concatenated with x_t^{RIM} to compute the input matrix, $X_t \in \mathbb{R}^{2 \times d}$.

$$X_t = \emptyset \oplus x_t^{RIM} \tag{20}$$

where $\oplus$ is a concatenation operation. Next, similar to Vaswani et al. (2017), linear projections ($W^e \in \mathbb{R}^{d \times d}$ for keys and $W^v \in \mathbb{R}^{d \times d}$ for values) of input matrix X_t are computed. Then, for RIM k, attention mechanism is employed as below.

$$\begin{aligned} a_{t,k}^{(in)} &= \text{softmax}\left(\frac{q_{t,k} \times (X_t \times W^e)^T}{\sqrt{d}}\right) \\ A_{t,k}^{(in)} &= a_{t,k}^{(in)} \times (X_t \times W^v) \end{aligned} \tag{21}$$

$a_{t,k}^{(in)}$ contains the normalized attention scores which represent how much RIM k is associated with the current input, x_t, and the null-input ($\emptyset$). Based on this, `top-m` RIMs (out of total K RIMs) are extracted which have the least attention on null-input (consequently, highest attention on the current input). At time step t and for input x_t, the set of `top-m` extracted RIMs is called the activated set, S_t.

3. Stage 3 (Independent RIM dynamics): After retrieving the activated set S_t from the previous step, Stage 3 aims to independently compute the internal dynamics of every RIM k. Formally, new hidden state $\tilde{h}_{t,k}$ of RIM k is computed as shown below.

$$\tilde{h}_{t,k} = h_{t,k}, \forall k \notin S_t \tag{22}$$

$$\tilde{h}_{t,k} = \text{LSTM}(h_{t,k}, A_k^{(in)}), \forall k \in S_t \tag{23}$$

[13] Input can be a set of elements or an individual element. In this work, input is a single element.

From Eqn. 22, it can be inferred that the hidden states of inactive RIMs ($k \notin S_t$) remains un-affected. On the other hand, *active* RIMs ($k \in S_t$), independently, utilize LSTM cells (Eqn. 23, parameters are not shared between RIMs) to get their updated hidden states, $\tilde{h}_{t,k}$s', based on the current input information (available in $A_k^{(in)}$) and the hidden states, $h_{t,k}$s'.

4. Stage 4 (Communication between RIMs): Post Stage 3, Goyal et al. (2021) consider sharing information among all the RIMs (activated or not). Intuitively, non-activated RIMs may store contextual information relevant for activated RIMs later on. That is why another attention mechanism with residual connection (Santoro et al. 2018) is employed as below.

$$q_{t,k} = \tilde{h}_{t,k} \times \tilde{W}_k^q, \forall k \in S_t \tag{24}$$

$$\begin{aligned} e_{t,k} &= \tilde{h}_{t,k} \times \tilde{W}_k^e, \forall k \\ v_{t,k} &= \tilde{h}_{t,k} \times \tilde{W}_k^v, \forall k \end{aligned} \tag{25}$$

$$h_{t+1,k} = \text{softmax}(\frac{q_{t,k} \times e_{t,:}^T}{\sqrt{d}}) \times v_{t,:} + \tilde{h}_{t,k}, \forall k \in S_t \tag{26}$$

where $\tilde{W}_k^q$, $\tilde{W}_k^e$ and $\tilde{W}_k^v$ represent RIM-specific learnable parameters. It must be noted that hidden states ($\tilde{h}_{t,k}, \forall k \in S_t$) of only activated set S_t are updated in Eqn. 26.

In summary, RIMs features a concurrent and dynamic temporal backbone suitable for the complex news event interactions in volatility forecasting. According to Goyal et al. (2021), RIMs can be viewed as a drop-in replacement for a LSTM layer, but potentially more powerful. For the sake of simplicity, we summarize the full RIMs architecture into the following abstraction:

$$h_{out} = \text{RIMs}(X) \tag{27}$$

where X is the full input sequence, and h_{out} is the final hidden representation produced by the model.

Appendix B: Time Complexity Analysis of LASER

Table 3 shows the time complexity of each component in LASER. Since the event encoding dominates other components, the time complexity of LASER is $O((N+P) \cdot l \cdot d^2)$. It is worth noting that our contribution is to process long-term memory more efficiently. Because $P \gg N$, by using a long-term memory retriever, we significantly decrease the sequential modeling time from $O(P \cdot d^2)$ to $O(N \cdot d^2)$. If we perform RNNs on the long-term memory directly without pruning (see the NAIVE model in Section 6.1), the theoretical time complexity is the same as the long-term memory retriever. The latter is, however, much faster in practice, since it can be implemented using the bi-linear transformation and access the long-term memory with a constant number of sequentially executed operations, whereas an RNN operating on the entire long-term memory requires $O(P)$ sequential operations.

Appendix C: Preambles: Beam Search and Dynamic Beam Allocation (DBA)

Language models aim to generate a text sequence, $y = (y_1, y_2 ... y_l)$ where every y_i belongs to a fixed set of vocabulary V, and l is the length of the sequence. Formally, the model seeks to learn *p(y)* which, according to Bengio et al. (2003), can be decomposed as below.

$$p(y) = \prod_{m=1}^{l} p(y_m | y_{<m}) \tag{28}$$

Model Component	Complexity per Component	Corresponding Equation
Event encoding	$O((N+P)\cdot l\cdot d^2)$	Eqn. 4
Short-term seed extractor	$O(N\cdot d^2)$	Eqn. 6
Long-term memory retriever	$O(P\cdot d^2)$	Eqn. 7
Sequential modeling	$O(N\cdot d^2)$	Eqn. 8 and Eqn. 9
Volatility prediction	$O(d^2)$	Eqn. 11

Table 3 The time complexity of each component in LASER. N is the number of events in the short-term memory M_t^s, P is the number of events in the long-term memory M_t^l, l is the sentence length, and d is the representation dimension.

In Eqn. 28, at every time step m, the model produces a distribution over V and retrieves the most-probable (top-1) token or hypothesis. The model stops generating tokens when either a special end-of-sentence token is produced or a maximum length of the sentence is reached. An alternative to improve the performance of the language model is to utilize *beam search* (Sutskever et al. 2014).Briefly, at every time step, the model keeps track of a beam (`top-b` hypotheses with probability scores, $p(y)$'s). When stopping conditions are met, a single hypothesis based on the highest recorded score is returned.

Recently, Hokamp and Liu (2017) introduce the Grid Beam Search (GBS) algorithm which ensures that the output sequence, y, contains some predefined constrains, c. Briefly, GBS keeps a track of $(c+1)\times r$ beams or *banks*, $B_0..B_c$, instead of 1 beam as in classical beam search. The purpose of each bank, B_i, is to store the set of hypotheses (with their probability scores $p(y)$'s) which meet i constraints. For example, bank B_c stores the set of hypotheses satisfying all the constraints, whereas bank B_0 has the set of hypotheses satisfying no constraints. When the stopping conditions are met, the language model returns a single best hypothesis from bank B_c, based on the highest recorded score. According to Post and Vilar (2018), the time complexity of GBS is $\mathcal{O}(cbl)$, where b represents beam-size and l represents the length of sentence. It restricts the number of constraints that the GBS algorithm can digest. Therefore, Post and Vilar (2018) develop Dynamic Beam Allocation (DBA), an improved decoding algorithm which has better efficiency ($\mathcal{O}(bl)$). DBA utilizes the following three rules to generate the set of candidate hypotheses:

Rule 1: `top-b` hypotheses; similar to classical beam search

Rule 2: for each hypothesis, all unmet constraints (to ensure coverage of constraints)

Rule 3: for each hypothesis, the single best token

It must be noted that both GBS and DBA rely on RNN-based, task-specific language models. More importantly, both treat the clue words as hard constraints without considering their temporal order. We devise an enhanced decoding algorithm simply named BEAM, which (1) imposes the temporal ordering of clue/constraint words naturally arising from the thread construction component in LASER, in text generation, (2) treats the clues as soft constraints and allows for semantically close words to be used in generation. For example, the word `gain` can be treated as a legitimate satisfaction of the contraint `return`, and (3) leverages the pre-trained GPT-2 language model (Radford et al. 2019) as glue to generate fluent narratives.

Appendix D: BEAM: More Details

D.1. BEAM Decoder

Fig 7 illustrates the working of the BEAM decoder with a real example.

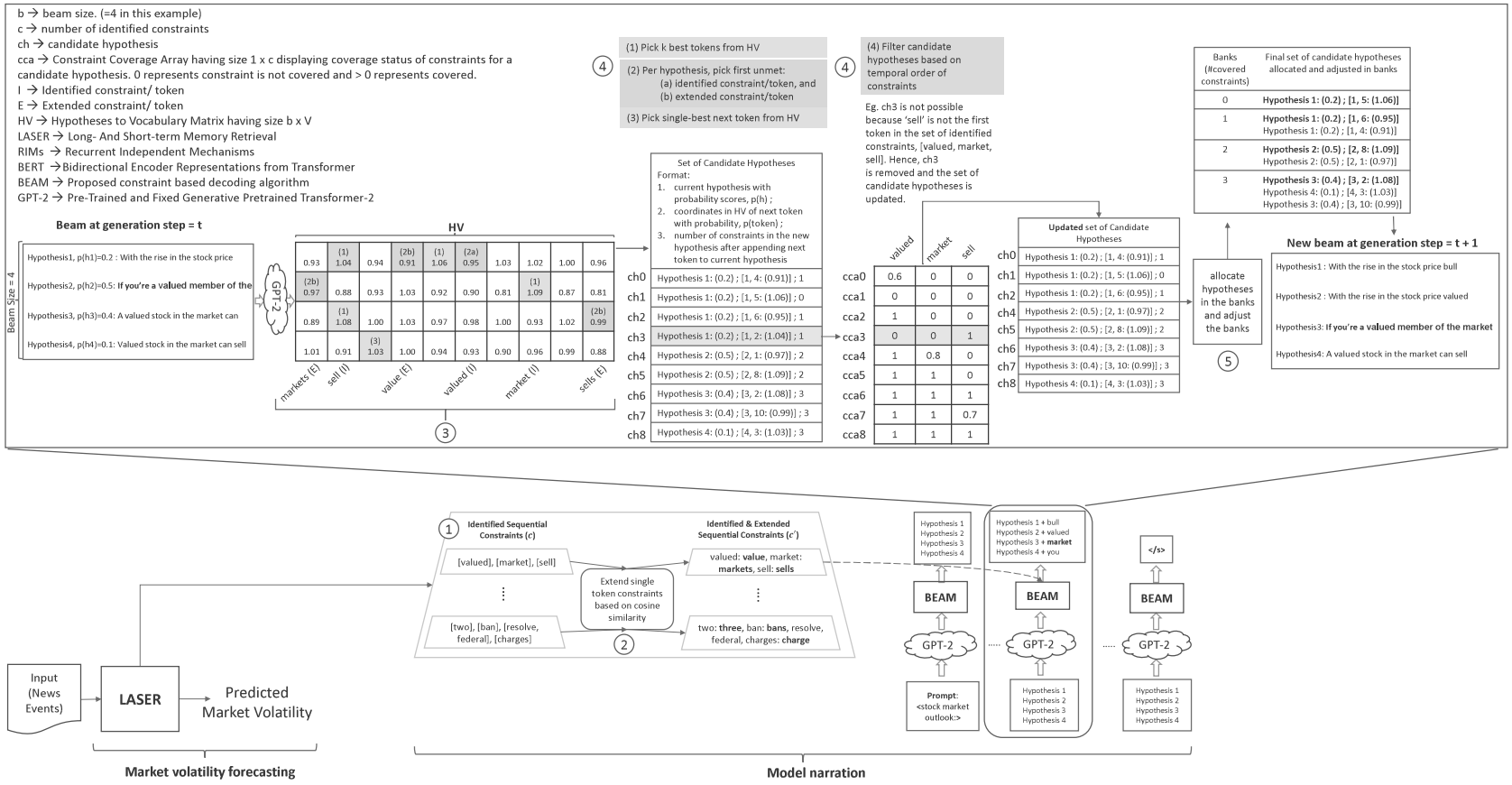

Figure 7 **A step of the BEAM decoder (to maintain clarity, one generation step t with one thread is shown) (1) Identify sequential constraints. For details, see Algorithm 1. (2) Extend the set of identified sequential constraints, c, based on cosine similarity to retrieve c'. (Our contribution). (3) Produce Hypothesis-to-Vocabulary matrix (HV). (4) Apply four rules (4^{th} rule is our contribution) to formulate set of candidate hypotheses. (5) Allocate candidate hypotheses in banks and adjust the banks to generate new beam for step t + 1.**

D.2. Comparison with Other Models

Table 4 summarizes characteristics of BEAM and its predecessor, with examples. A unique feature not shown in the table is BEAM's deployment of GPT-2 as glue in controlled text generation.

Algorithm	Characteristics		Time Complexity	Example Constraints	Example Narrations
	Constraints oriented	Upholds sequential order of constraints			
Classical Beam Search (CBS)	✗	✗	$\mathcal{O}$(bl)	N/A	The rate for the first quarter of 2017 was down.
Grid Beam Search (GBS)	✓	✗	$\mathcal{O}$(cbl)	[[rise], [key, inflation], [stock, market], [index], [annual, loss], [feel]]	The **rise** of **key inflation feel** like the beginning of the end.
Dynamic Beam Allocation (DBA)	✓	✗	$\mathcal{O}$(bl)	[[rise], [key, inflation], [stock, market], [index], [annual, loss], [feel]]	The **key inflation** rate for the first quarter of 2017 was 1.8 percent, down from 2.
BEAM	✓	✓	$\mathcal{O}$(bl)	[[rise: rises], [key, inflation], [stock, market], [index: Index], [annual, loss], [feel: feels]]	The **rise** of **key inflation** in the United States has been a **stock market index**'s biggest **annual loss** since the Great Depression.

Table 4 Comparison of decoding algorithms. c represents the number of constraints, b represents the beam size, and l represents sentence length.

Appendix E: Experiments: More Details

E.1. Dataset Statistics

Table 5 summarizes the descriptive statistics of our data. There is clearly a distribution shift problem, which justifies our choice of RIMs as the sequence backbone of LASER.

Dataset	Days	News	High-Volatility Incidents (H=1)	High-Volatility Incidents (H=22)
Training	2,898	75,365	32%	33%
Validation	723	16,353	8%	5%
Test	702	15,471	12%	17%

Table 5 Descriptive statistics of the dataset

E.2. Engineering Configurations

In LSTM versions of our proposed models, we utilize one-layer LSTM, and dimension of the hidden state is 300. We use the Xavier uniform initializer (Glorot and Bengio 2010) to initialize the parameters. In the RIMs-based models, we utilize 4 RIMs (K=4) and keep 2 RIMs (`top-m` = 2) activated at any time. Moreover, the dimension of the hidden state of an individual RIM is kept to 75. The short-term query extractor extracts 3 top queries (`top-j`, size of M_t^{xs}, and number of threads = 3); the long-term memory retriever outputs 30 long-term precursors per thread (`top-g` and size of $M_t^{'rl}$ = 30). In the LTM, the parameters of LSTM or RIMs are shared across threads.

For all news titles, we set a fixed length limit for encoding, $l = 20$. We train all the models by an Adam optimizer (Kingma and Ba 2014) with a learning rate of 10^{-4}, the dropout rate of 0.2, and the batch-size equal to 32.

Lastly, for model narration, γ_{token} is the attention score threshold that allows us to extract top-5 tokens from the news events for each thread in LASER. Any narrative is generated to the maximum length $r = 30$. The hyperparameter β in Equation 15 is set to be 70 with validation experiments.

E.3. LASER Performance: Instance-Level Evaluation

We also examine instance-level performance by demonstrating model output on real testset data points corresponding to high-volatility incidents. In both examples (see Table 6), we observe that the market volatility predicted by LASER is much closer to the ground-truth value than all baseline models.

Example ID	Ground-Truth MV (v_{tf} in %age)	Model	Predicted MV ($\hat{v_{tH=1}}$ in %age)
1	4.2	ARIMA	0.9
		NSVM	2.3
		StockNet	1.7
		NAIVE	1.8
		LASER	**3.9**
2	3.1	ARIMA	0.8
		NSVM	2.2
		StockNet	1.8
		NAIVE	2.4
		LASER	**2.7**

Table 6 Instance-level model performance when $H = 1$

Appendix F: Model Narration: Example Narratives

Table 7 and Table 8 illustrate narratives generated by the BEAM algorithm.

Appendix G: Full LASER-BEAM Pipeline

The full LASER-BEAM pipeline is illustrated through a real example in Fig. 8.

Thread Index	Constraints	Algorithm	Generation	Generated Explanations	GenScore	Coverage Ratio	Fluency	Informativeness
1	[['cut'], ['problem'], ['scared'], ['emails'], ['expenses']]	Constraint	N/A	cut problem scared emails expenses	70.33	1.00	1	1
		Sentence	N/A	Hanergy Thin Film wants to **cut** its workforce by more than a third in a restructuring California's earthquake **problem**.	165.55	0.4	2	2
		DBA	GPT-2	The **problem** with the stock market is that it 's **scared** of the future . It 's **scared** of the future because it 's afraid of the future.	40.85	0.4	2	2
		BEAM	GPT-2	A **cut** in the federal debt ceiling would be a **problem** for **scared** investors , but it would be a good thing for the economy.	50.5	0.6	5	5

Table 7 Example ID#1: Generated narratives for short-term thread (forecasting model is LASER, and horizon $H = 1$)

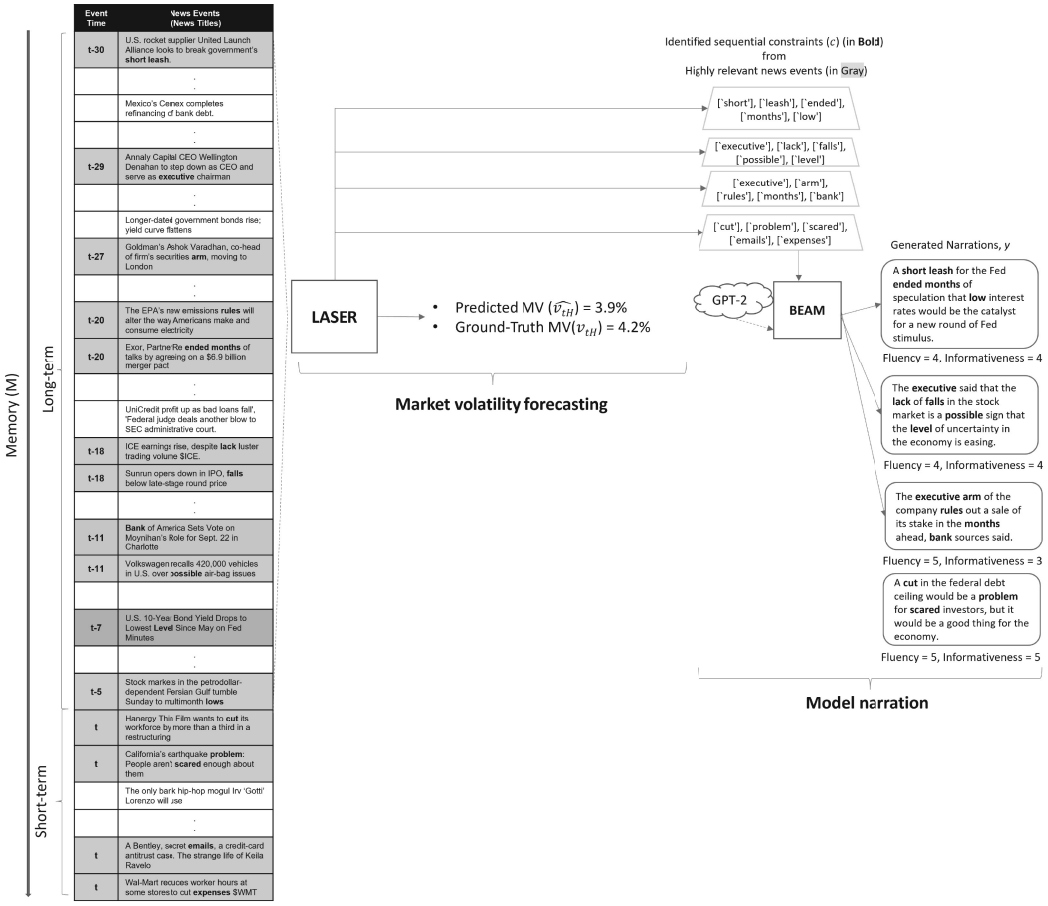

Figure 8 Volatility forecasting and model narration for ExampleID#1. The left-hand-side depicts events in short- and long-term memory; those retrieved by LASER are highlighted and time-tamped (e.g., t-11 means it is 11 days before anchor time).

Thread Index	Constraints	Algorithm	Generation	Generated Explanations	GenScore	Coverage Ratio	Fluency	Informativeness
1	[[‘short’], [‘leash’], [‘ended’], [‘months’], [‘low’]]	Constraint	N/A	short leash ended months low	53.42	1.0	1	1
		Sentence	N/A	U.S. rocket supplier United Launch Alliance looks to break government’s ‘**short leash**’ Exor, PartnerRe **ended months** of talks.	178.26	0.6	2	2
		DBA	GPT-2	It’s a good time to be a stock investor. The S&P 500 is up more than 20% since the beginning of the year.	26.56	0.0	3	3
		BEAM	GPT-2	A **short leash** for the Fed **ended months** of speculation that **low** interest rates would be the catalyst for a new round of Fed stimulus.	65.96	1.0	4	4
2	[[‘executive’], [‘lack’], [‘falls’], [‘possible’], [‘level’]]	Constraint	N/A	executive lack falls possible level	68.31	1.0	1	2
		Sentence	N/A	Annaly Capital CEO Wellington Denahan to step down as CEO and serve as **executive** chairman ICE earnings rise, despite **lack**luster trading volume ICE Sunrun opens	206.26	0.2	3	2
		DBA	GPT-2	It’s possible that the **lack** of **executive** level leadership at the top falls on the shoulders of the board.	57.67	1.0	3	2
		BEAM	GPT-2	The **executive** said that the **lack** of **falls** in the stock market is a **possible** sign that the **level** of uncertainty in the economy is easing.	69.11	1.0	4	4
3	[[‘executive’], [‘arm’], [‘rules’], [‘months’], [‘bank’]]	Constraint	N/A	executive arm rules months bank	63.34	1.0	2	1
		Sentence	N/A	Annaly Capital CEO Wellington Denahan to step down as CEO and serve as **executive** chairman Goldman’s Ashok Varadhan.	164.05	0.2	2	1
		DBA	GPT-2	The **bank**’s outlook for the U.S. economy is unchanged, with a slight increase in the probability of a rate hike in the **months** ahead.	37.12	0.4	4	4
		BEAM	GPT-2	The **executive arm** of the company **rules** out a sale of its stake in the **months** ahead , **bank** sources said.	74.95	1.0	5	3

Table 8 Example ID#1: Generated narratives for long-term threads (forecasting model is LASER model, and horizon $H = 1$)