Appendices

A Limitations and Societal Impacts

Limitations. Our framework utilizes large language models such as GPT-3.5 and thus inherits the issues of those models such as hallucination and biased content. Moreoever, the GPT access charges fees. Our compatibility function c is a Transformer-based neural network which is known to be data-hungry. Although it worked well in our experiments, it might still suffer compared to non-neural models if starved of data. Additionally, our method requires a pretrained event model to propose predictions so its overall performance is conditioned on this event model; if it is misspecified or poorly trained, our framework may not work well.

Societal Impacts. Our paper develops a novel method to integrate large language models into temporal modeling. By describing the model and releasing code, we hope to facilitate the modeling of continuous-time sequential data in many domains. However, like many other machine learning models, our method may be applied to unethical ends. For example, its abilities of better fitting data and making more accurate predictions could potentially be used for unwanted tracking of individual behavior, e.g. for surveillance.

B Method Details

B.1 ANHP on Amazon Data

On Amazon data, the ANHP model we used is directly from Yang et al. (2022).

B.2 Multi-relational ANHP on GDELT Data

On GDELT data, we apply a multi-relational variant of ANHP to model the event sequences that contain evolving knowledge about entity interactions.

Continuous-time Transformer. For any multi-relational event $e@t$ where $e := (u, v, r)$ denotes a tuple of the subject, object and predicate, we first compute an initial **embedding** $\llbracket e \rrbracket(t) = [\llbracket e^u \rrbracket \cdot \llbracket e^r \rrbracket \cdot \llbracket e^v \rrbracket(t)]$, then attend it to its **history** $\mathcal{H}(e@t)$, with the continuous-time transformer structure (Yang et al., 2022), to compute the relational score. More precisely, $\llbracket e \rrbracket(t)$ is the concatenation of layer-wise embeddings $\llbracket e \rrbracket^{(0)}(t), \llbracket e \rrbracket^{(1)}(t), \ldots, \llbracket e \rrbracket^{(L)}(t)$. For $\ell > 0$, the layer-ℓ embedding of $e@t$ is computed as

$$\llbracket e \rrbracket^{(\ell)}(t) \overset{\text{def}}{=} \underbrace{\llbracket e \rrbracket^{(\ell-1)}(t)}_{\text{residual connection}} + \tanh\left(\sum_{f@s \in \mathcal{H}(e@t)} \frac{\mathbf{v}^{(\ell)}(f@s)\ \alpha^{(\ell)}(f@s, e@t)}{1 + \sum_{f@s \in \mathcal{H}(e@t)} \alpha^{(\ell)}(f@s, e@t)}\right) \tag{6}$$

where the unnormalized attention weight on each relevant event $f@s \in \mathcal{H}(k@t)$ is

$$\alpha^{(\ell)}(f@s, e@t) \overset{\text{def}}{=} \exp\left(\tfrac{1}{\sqrt{D}}\, \mathbf{k}^{(\ell)}(f@s)^{\top} \mathbf{q}^{(\ell)}(e@t)\right) \in \mathbb{R} \tag{7}$$

In layer ℓ, $\mathbf{v}^{(\ell)}$, $\mathbf{k}^{(\ell)}$, and $\mathbf{q}^{(\ell)}$ are known as the **value**, **key**, and **query** vectors and are extracted from the layer-($\ell-1$) event embeddings using learned layer-specific matrices $\mathbf{V}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{Q}^{(\ell)}$.

$$\mathbf{v}^{(\ell)}(e@t) \overset{\text{def}}{=} \mathbf{V}^{(\ell)} \left[1; \llbracket t \rrbracket; \llbracket e \rrbracket^{(\ell-1)}(t)\right] \tag{8a}$$

$$\mathbf{k}^{(\ell)}(e@t) \overset{\text{def}}{=} \mathbf{K}^{(\ell)} \left[1; \llbracket t \rrbracket; \llbracket e \rrbracket^{(\ell-1)}(t)\right] \tag{8b}$$

$$\mathbf{q}^{(\ell)}(e@t) \overset{\text{def}}{=} \mathbf{Q}^{(\ell)} \left[1; \llbracket t \rrbracket; \llbracket e \rrbracket^{(\ell-1)}(t)\right] \tag{8c}$$

As the base case, $\llbracket e \rrbracket^{(0)}(t) \overset{\text{def}}{=} [\llbracket e^u \rrbracket \cdot \llbracket e^r \rrbracket \cdot \llbracket e^v \rrbracket(t)]$ is a **learned embedding of the multi-relational event** e. $\llbracket t \rrbracket$ denotes an embedding of the time t. See Section 2 in (Yang et al., 2022) for more details.

Training. Given a collection of events recorded in a temporal window $[0, T)$, let $x_{[0,t)}$ be the event sequence at time interval $[0, t)$, we learn the model by minimizing the negative log-likelihood

$$\mathcal{L} = -\sum_{i=1}^{I} \log \lambda_{r_i}^{u_i, v_i}(t_i \mid x_{[0,t_i)}) + \sum_{r=1}^{n_r} \sum_{u=1}^{n_e} \sum_{v=1}^{n_e} \int_{t=0}^{T} \lambda_r^{u,v}(t \mid x_{[0,t)})dt \tag{9}$$

DATASET	# OF	# OF	# OF EVENT TOKENS		
	ENTITIES	PREDICATES	TRAIN	DEV	TEST
GDELT	2279	20	83000	9000	15000
AMAZON	1	24	50000	7000	13000

Table 1: Statistics of dataset.

The first term maximizes the probability of specific type of event between two entities; the second term penalizes non-presence of all possible types of events between all possible entity pairs in a given observation window. The overall training procedure is adopted from (Trivedi et al., 2017) where the Backpropagation Through Time (BPTT) training is conducted over a global sequence. A simple variant of Monte Carlo trick to compute the survival term of log-likelihood equation equation (9). See Appendix H in (Trivedi et al., 2017).

C Experimental Details

C.1 Dataset Details

GDELT (Global Database of Events, Language, and Tone) (Leetaru & Schrodt, 2013). The GDELT Project monitors events all over the world, with live datasets updated every 15 minutes. We only focused on the political events that happened in G20 countries from 2022-01-01 to 2022-07-31, ending up with a corpus of 109000 time-stamped event tokens. The event type k of each token has a structured name of the format "predicate(subject, object)". Each "predicate" is one of the twenty CAMEO codes such as "CONSULT" and "INVESTIGATE" (see Appendix C.4 for a full list), and each "subject" or "object" is one of the 2279 political entities such as "JOE BIDEN" and "RUSSIA". Each event token has a news headline that concisely describes the event. We split the dataset into disjoint train, dev, and test sets based on their dates: the 83000 events that happened before 2022-07-05 are training data; the 15000 events after 2022-07-15 are test data; the 9000 events between these dates are development data.

Amazon (Jianmo Ni, 2019). This dataset contains time-stamped user review behaviors on Amazon shopping pages from January 4, 2014 to October 02, 2016. Each user has a sequence of produce review events with each event containing the timestamp, the category of the reviewed product and a text of reviews, with each category corresponding to an event type. The categories of all items are first ranked by frequencies and the top 23 are kept while the rests are merged into one category, with each category corresponding to an event type. We work on a subset of 2500 most active users with average sequence length 32 and end up with $K = 24$ event types. We split the dataset into disjoint train, dev and test set based on time line: events before August 1, 2015 and after February 1, 2016 are categorized into train and test set while the events in between the two dates are grouped into dev set. The total number of event tokens for train, dev and test are $50000, 7000, 13000$, respectively.

Table 1 shows statistics about each dataset mentioned above.

C.2 Evaluation Metric Details

Here we give the precise definitions of MAP and MAR.

MAP is computed as follows. We first initialize $N = C = 0$. Then, for each partial heldout event (i.e., only t given in Amazon Review, t and some attributes given in GDELT), the model gives its top M predictions on the attribute of interest (e.g., type k on Amazon Review and "object" on GDELT). There might be multiple ground-truth attributes for a heldout event since multiple events may be recorded at the same time (due to time quantization and recording errors). If any of the ground-truth is in the top M list, we update N and C: first, we have $N \mathrel{+}= J$ where $J \leq M$ is the number of the ground-truth attributes that are covered in the top M list; then, for j^{th} covered ground-truth, we let $C \mathrel{+}= j/R$ where R is its rank in the top M list. In the end, MAP is defined as C/N. Each i/R is a pseudo-count for this event: it is in $(0, 1]$ since the rank R of the jth correct prediction will be surely $\geq j$; if it ranks at the top, i/R is close to 1 and it is counted as "predicted correctly"; if it ranks at the bottom, i/R is close to 0, meaning that it is almost missed by the top M proposals.

MAR is defined as follows. We first initialize $N = C = 0$. For each partial heldout event, the model gives its top M proposals on the attribute of interest; for the jth correct prediction in the list, we let

$C \mathrel{+}= j/R$ where R is its rank in the top M list; in the end, MAR is defined as C/N where N is the number of actual events.

The mean rank in section 5.2 is not affected by the fact that there might be multiple ground-truth attributes. That is because the contribution of jth ground-truth (among the multiple) to the final number is only dependent on its rank in the list but not its j index.

C.3 Implementation Details

All models are implemented using the PyTorch framework (Paszke et al., 2017).

For the implementation of NHP, AttNHP and energy functions, we used the code from the public Github repository at `https://github.com/ant-research/hypro_tpp` (Xue et al., 2022) with Apache License 2.0.

For the implementation of Know-Evolve, we used the code from the public Github repository at `https://github.com/rstriv/Know-Evolve` (Trivedi et al., 2017) without any license.

For the implementation of DyRep, we used the code from the public Github repository at `https://github.com/uoguelph-mlrg/LDG` (Trivedi et al., 2019) without any license.

For the implementation of MAR@M and MAP@M, we used the code from the public Github repository at `https://github.com/statisticianinstilettos/recmetrics` without any license.

For the implementation of Levenshtein distance, we used the code from the public Github repository at `https://github.com/maxbachmann/Levenshtein.git` with General Public License 2.0.

To compute the dense representations of text descriptions of events, we use the code from the public Github repository at `https://github.com/UKPLab/sentence-transformers` with Apache License 2.0.

C.4 Event Prompting

We show one example of the prompt structures used for GDELT dataset as below. The effect event consists of an event type, an event time, a subject name and an object name while the causal events consists of an event headline additionally.

```
I want you to do the reasoning over social events. I given you an effect event
and you give me four or five causal events. An effect event is an event that
happens. A causal event is an event that is believed to be one of the causes that
 trigger an effect event to happen. Each event consists of an event headline, an
event type, an event time, subject name and object name of describing the event.

The event types of the effect and cause events are restricted to 20 types, with
names (in capital) and the descriptions below.
1. MAKE STATEMENT: decline comment, make comments.
2. APPEAL: appeal for material, economic, military, humanitarian aid or
cooperation.
3. EXPRESS INTENT TO COOPERATE: Express intent to engage in material, diplomatic,
 military aid.
4. CONSULT: make a visit, host a visit.
5. ENGAGE IN DIPLOMATIC COOPERATION: praise or endorse, defend verbally.
6. ENGAGE IN MATERIAL COOPERATION: cooperate economically, militarily, judicially.

7. PROVIDE AID: provide economic, military, humanitarian aid.
8. YIELD: ease admin or political sanctions or military blockade, return and
release.
9. INVESTIGATE: investigate crime, corruption, human rights abuses, military
actions.
10. DEMAND: demand any type of cooperation, aid, reforms, rights, easing of
sanctions.
11. DISAPPROVE: criticize or denounce, accuse of crime, human rights abuses,
complain officially and make lawsuit against.
12. REJECT: reject material, economic, military, judicial operations, requests or
 plans.
13. THREATEN: threaten to reduce aid, to boycott, to reduce or break relations,
to impose sanctions.
```

```
14. PROTEST: civilian demonstrations.
15. EXHIBIT MILITARY POSTURE.
16. REDUCE RELATIONS: reduce or break any relations.
17. COERCE: seize or damage properties, impose administrative sanctions or
restrictions.
18. ASSAULT: use of unconventional forms of violence.
19. FIGHT: uses of conventional force.
20. ENGAGE IN UNCONVENTIONAL MASS VIOLENCE.

Now I give you 10 examples of event reasoning. In each example, the first event
is the effect event, the next three to five events are cause events that happen
earlier.


## Example 1

effect event
event type: Use unconventional violence
event time: 2022-03-08
subject name: JOE BIDEN
object name: RUSSIA

-----------------------------------
reasoning:
cause event 1
event headline: Joe Biden Welcomes Visa & Mastercard's Decision To Suspend
Operations In Russia
event type:  ENGAGE IN DIPLOMATIC COOPERATION
event time: 2022-03-06
subject name: RUSSIA
object name: JOE BIDEN

cause event 2
event headline: Putin says Ukraine's future is in doubt as cease-fires collapse
event type: Threaten
event time: 2022-03-06
subject name: RUSSIA
object name: UKRAINE

cause event 3
event headline: Zelensky Urges US Lawmakers to Ban Russian Oil, Provide Fighter
Jets
event type:  Make statement
event time: 2022-03-05
subject name: COLONEL
object name: RUSSIA

cause event 4
event headline: Marshall, Moran seek ban on Russian oil imports, urge expansion
of domestic production
event type: Use unconventional violence
event time: 2022-03-05
subject name: SENATOR
object name: RUSSIA
```

Listing 3: Prompt Structures used for GDELT dataset.

The following is one exemplar of the prompt used for Amazon dataset. The effect event consists of product category and event time while the causal events consists of a review text.

```
I want you to do the reasoning over the events that are extracted from online-
shopping review data. I given you an effect event and you give me two to four
cause events. An effect event is an event that happens. A causal event is an
event that is believed to be one of the causes that trigger an effect event to
happen. Each event corresponds to an review submitted by the customer, which
```

```
consists of an product category(event type), a product title, an event time,
summary text and review text from the user that describes the feedback of the
shopping event.

The product categories are restricted to the following set:
1. Women Shoes,
2. Men Shoes,
3. Men Clothing,
4. Women Clothing,
5. Novelty & More,
6. Men Uniforms, Work & Safety,
7. Women Jewelry,
8. Costumes & Accessories,
9. Men Accessories,
10. Luggage & Travel Gear,
11. Men Watches,
12. Women Accessories,
13. Children Shoes,
14. Children Clothing
15. Shoe, Jewelry & Watch Accessories,
16. Women Watches,
17. Women Uniforms, Work & Safety,
18. Men Surf, Skate & Street,
19. Women Handbags & Wallets
20. Men Jewelry
21. Children Accessories
22. Women Maternity
23. Women General
24. Others

# Example 1

effect event
product category: Luggage & Travel Gear
event time: 2013-10-19

-----------------------------------
reasoning:
cause event 1
product category: Novelty & More
product title: Sports Katz Peace Out Socks
event time: 2013-09-24
summary text: Peace Out Socks
review text: We ordered these for soccer for my daughter and they worked out well.
 They are very cute and have held up. They are a quite thick, which is why I only
 gave them 4 stars. They will work better for fall season; I would not recommend
them for summer soccer, if soccer is your intended use.

cause event 2
product category: Luggage & Travel Gear
product title: Wildkin Pink Retro Kickstart Messenger Bag
event time: 2013-09-09
summary text: Wildkin Kickstart Messenger Bag
review text: My daughter, going to 4th grade, wanted a messenger type bag this
year for school. I searched and searched on Amazon for the perfect bag, but all
of the adult bags seemed huge (like they would definately look funny with her
carrying them). Thank goodness I found this bag. It is the perfect size for her,
and her favorite color!.

cause event 3
product category: Children Clothing
product title: Levi's Girls' 715 Thick Stitch Bootcut Jean
event time: 2013-09-01
summary text: Thick Stitch Bootcut Jeans
```

```
review text: I love Levis and this style is sooo cute! My daughter even likes
them! The price was amazing in these, as we found similar Levis at Dillard’s for
36. The waist of these Levi in particular run small compared to other Levis of
the same size. During the Levis sale on Amazon, we purchased these, the Sweetie
Skimmers, and the Scooter. All of the others ran almost half an inch larger in
the waist. I worry the larger size would be huge on her, but I think I will try
the larger size, as they must be comfortable on her or she will not wear them.

cause event 4
product category: Children Shoes
product title: New Balance KJ750 Running Running Shoe (Lttle Kid/Big Kid)
event time: 2013-02-02
summary text: Cute and durable!
review text: I ordered these for my 8 year old daughter. She usually destroys
shoes within months, but these have held up!
```

Listing 4: Prompt Structures used for Amazon dataset.

For the full set of prompts used in the paper, please see `scripts/gdelt/gdelt/gdelt_prompts` and `scripts/amazon/amazon/amazon_prompts` in our submitted code.

C.5 Does GPT remember Amazon Review data?

We have a conversion with GPT-3.5-turbo to verify that it does not remember the exact information of Amazon Review data.

```
input: We are using Amazon Review Data, published by Jianmo Ni, , 2018,  which is
 firstly used in the paper "Justifying recommendations using distantly-labeled
reviews and fined-grained aspects", EMNLP 2019.

The dataset includes millions of reviews (ratings, text, helpfulness votes),
product metadata (descriptions, category information, price, brand, and image
features) in the range May 1996 - Oct 2018. Do you remember any of the review
information if I give you a review id?

GPT-3.5 output: As an AI language model, I am not capable of remembering any
review information unless I have been programmed to store and retrieve specific
pieces of information. However, I can search for and retrieve review information
from the dataset if given a review ID.

input: can you recall any information with reviewer ID A28GK1G2KDXHRP on date 04
17, 2014.

GPT-3.5 output: I apologize, but the Amazon Review dataset does not include
information about the specific date when a review was posted. It only includes
the year and month when the review was posted. Therefore, I cannot provide you
with any information about reviews posted by reviewer ID A28GK1G2KDXHRP on April
17, 2014.

input: Reviewer A0148968UM59JS3Y8D1M bought Hanes Men’s Classic Crew on 01 18,
2014, what review did he wrote?

GPT-3.5 output: I apologize, but as an AI language model, I do not have access to
 real-time or updated information about specific Amazon reviews or purchases made
 by individual reviewers. Additionally, the Amazon Review dataset does not
include information about the specific content of a specific review. It only
includes metadata such as the reviewer ID, product ID, and overall rating.
Therefore, I cannot provide you with the review written by the reviewer
A0148968UM59JS3Y8D1M for the Hanes Men’s Classic Crew on January 18, 2014 or any
other date.

input: Reviewer A0148968UM59JS3Y8D1M bought Hanes Men’s Classic Crew on 01 18,
2014, his review starts with ’they seem very well made..’, can you recall the
complete review he wrote?
```

```
GPT-3.5 output: I apologize, but as an AI language model, I do not have access to
 real-time or updated information about specific Amazon reviews or purchases made
 by individual reviewers. Additionally, the Amazon Review dataset does not
include information about the specific content of a specific review. It only
includes metadata such as the reviewer ID, product ID, and overall rating.
Therefore, I cannot provide you with the complete review written by the reviewer
A0148968UM59JS3Y8D1M for the Hanes Men's Classic Crew on January 18, 2014, or any
 other date.

input: on which date did reviewer A0148968UM59JS3Y8D1M buy Hanes Men's 7-Pack
ComfortSoft Briefs?

GPT-3.5 output: Sorry, as an AI language model, I do not have access to personal
information or purchase history of individuals.
```

Listing 5: Prompt examples used for testing the memory of GPT-3.5-turbo

C.6 Training and Testing Details

Language Models. We use the variants of `davinci` and `gpt-3.5-turbo` for GPT-3 and GPT-3.5 models respectively and access them via the OpenAI API. To produce a stable output, we set the temperature to be 0.

Training Base Models. For each method, we used grid search on hyperparameters and chose the ones providing the best performance on the held-out dev set (see Table 2). For KE, DyRep, ANHP, the main hyperparameters to tune are the entity embedding size D_1, predicate embedding size D_2 , RNN hidden size D_3 used in the network and the number of layers L of the attention structure (DyRep and ANHP). In practice, the optimal D for a model was usually $4, 8, 16, 32$; the optimal L was usually $1, 2, 3, 4$. In the experiment, to train the parameters of the base model, we performed early stopping based on log-likelihood on the held-out dev set.

Causal Events Retrieval. For each event, we conduct few-shot prompt (see Appendix C.4 for detailed exemplars) to generate a set of causal events. For each generated causal event, we compute the distance of textual descriptions between itself and the event in the history and then retrieve the closest α_{retro} events as the 'real causal events'.

- For GDELT dataset, the textual description of the event consists of event headline, event type, event time, subject name and object name. We set $\alpha_{retro} = 2$ so that the average length of retrieved event sequence is 10.

- For Amazon dataset, the textual description consists of product category, product tile, event time and review text. We set $\alpha_{retro} = 4$ so that the average length of retrieved event sequence is 10.

Training Ranking Model. The score function of the ranking model is built on ANHP with 3 MLP layers to project the hidden states into a scalar energy value. The ANHP used by the ranking model is set to have the same structure as the base model 'ANHP'. The hyperparameters are tuned within a range of values that make the score function to have a similar size of total paramters with the base ANHP model. During training, each pair of training sample contains 1 positive sample and N negative samples: $N = 5$ for the task of predicate prediction and $N = 20$ for the task of object prediction on GDELT data, $N = 5$ for the task of both event type and time prediction on Amazon data.

Computation Cost. All the experiments were conducted on a server with 256G RAM, a 64 logical cores CPU (Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz) and one NVIDIA A100 GPU for acceleration.

For training, our batch size is 8. For GDELT data, training the baseline KE, DyRep and ANHP approximately takes 1 hour, 1.3 hour, and 2 hours respectively (4.5, 5.8, 9 milliseconds per sequence). On Amazon data, training the baseline NHP and ANHP approximately takes 1 hour and 2 hours, respectively. On both dataset, training the ranking model takes around 30 minutes (9 milliseconds per sequence pair) respectively.

Model	Description	Value used	
		GDELT	Amazon
Know-Evolve	Entity embedding size	16	NA
	Predicate embedding size	4	NA
	RNN hidden size	16	NA
DyRep	Entity embedding size	16	NA
	Predicate embedding size	4	NA
	RNN hidden size	16	NA
	Attention Layers Number	1	NA
NHP	RNN hidden size	NA	36
	Entity embedding size	NA	16
	Predicate embedding size	NA	8
ANHP	Entity embedding size	16	32
	Predicate embedding size	4	32
	Heads number	2	1
	Attention Layers Number	1	2
Ranking Model	Temporal embedding size	40	24
	Hidden size	60	56
	Heads number	4	16
	Attention Layers Number	3	3

Table 2: Descriptions and values of hyperparameters used for models trained on the two datasets.

C.7 More Results

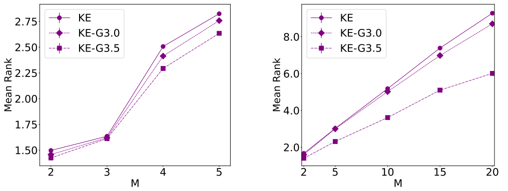

Figure 10: GPT-3.0 vs. GPT-3.5: mean rank on predicate prediction (left) and object prediction (right) on GDELT dataset. KE as the base model

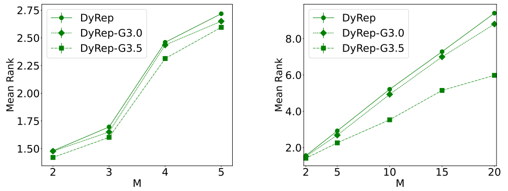

Figure 11: GPT-3.0 vs. GPT-3.5: mean rank on predicate prediction (left) and object prediction (right) on GDELT dataset. DyRep as the base model.

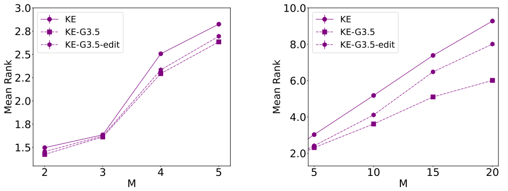

Figure 12: Bert embedding distance vs edit distance: mean rank on predicate prediction (left) and object prediction (right) on GDELT dataset. KE as the base model

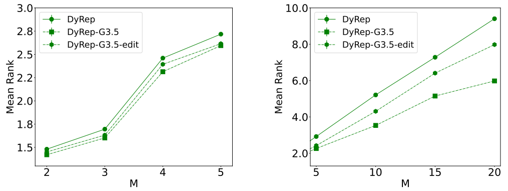

Figure 13: Bert embedding distance vs edit distance: mean rank on predicate prediction (left) and object prediction (right) on GDELT dataset. DyRep as the base model.

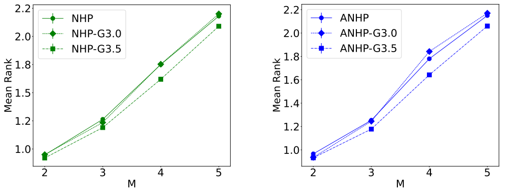

Figure 14: GPT 3.0 vs GPT 3.5: mean rank on type prediction on Amazon dataset.

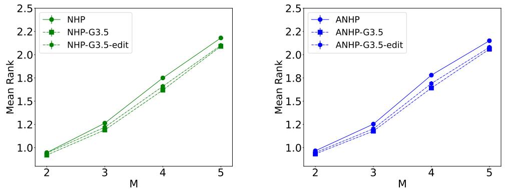

Figure 15: Effect of distance measure: mean rank on type prediction on Amazon dataset.