IDEATE: Detecting AI-Generated Text using Internal and External Factual Structures

Quan Wang[1], **Licheng Zhang**[2], **Zikang Guo**[2], **Zhendong Mao**[2,*]
[1]MOE Key Laboratory of Trustworthy Distributed Computing and Service,
Beijing University of Posts and Telecommunications, Beijing, China
[2]University of Science and Technology of China, Hefei, China
wangquan@bupt.edu.cn, {zlczlc,gzk170401}@mail.ustc.edu.cn, zdmao@ustc.edu.cn

Abstract

The effective detection of AI-generated text is a vital principle to ensure responsible use of large language models (LLMs). Previous studies mainly focused on discovering and utilizing internal evidences contained in the text itself to perform the detection, while ignoring external evidences implicated in an established knowledge graph (KG) which may also be key discriminative factors between AI-generated and human-written text. To address this deficiency, we propose IDEATE, a novel hierarchical graph network that utilizes both internal and external factual structures to detect AI-generated text. IDEATE consists of a mention-level subgraph at the bottom to describe internal factual structures of mentioned entities reflected in the input text, and an entity-level subgraph at the top to describe external factual structures of mentioned entities reflected in an external KG. Hierarchical graph convolution is then applied successively on the two subgraphs, through which the two types of factual structures will be embedded into the output and used for the final detection. Extensive experiments on four benchmarking datasets show that IDEATE consistently outperforms current state-of-the-art methods in detecting text generated by various LLMs, ranging from GPT-2 to the more powerful ChatGPT, verifying the necessity and superiority of introducing external evidences for AI-generated text detection.

Keywords: AI-generated text detection, internal factual structures, external factual structures

1. Introduction

The past few years have witnessed tremendous advances in generative AI, particularly natural language generation (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; Ouyang et al., 2022). Large Language Models (LLMs), such as the recently developed ChatGPT (OpenAI, 2022), can now generate text of supreme quality which demonstrates exceptional performance in various tasks like writing documents, answering questions, and composing emails. The increasing capability of LLMs to produce human-like text at high efficiency, however, also raises concerns about their misuse for malicious purposes, e.g., disinformation (Zellers et al., 2019; Stiff and Johansson, 2022), phishing (Giaretta and Dragoni, 2020), fraudulent product reviews (Adelani et al., 2020), and academic dishonesty (Dehouche, 2021). The effective detection of AI-generated text thus becomes a vital principle to ensure responsible use of generative AI tools like LLMs.

AI-generated text detection is typically formalized as a binary classification task, with a focus on discovering and exploiting textual disparities that discriminate between AI-generated and human-written text (Tang et al., 2023). Such textual disparities can be expressed implicitly in neural representations (Zellers et al., 2019; Guo et al., 2023), or explicitly as statistical or linguistic features (Gehrmann et al., 2019; Mitchell et al., 2023). In addition to textual disparities, Zhong et al. (2020) and Liu et al. (2022) recently find that the two types of text also differ in entity coherence. AI-generated text is more likely to mention inconsistent entities while human-written text would not. Whether textual or entity coherential, the previous studies focus mainly on internal disparities reflected in the text itself, while ignoring external disparities that may be implicated in other resources, e.g., a knowledge graph (KG).

In this work we claim that besides internal disparities, AI-generated and human-written text also differ in their external factual structures, i.e., structures of entities mentioned in the text as reflected in an external KG. Our claim is inspired by the well-known hallucination phenomenon of generative AI tools, in particular LLMs, which reveals that AI-generated text, though syntactically sound and fluent, could often be factually incorrect or unfaithful. Therefore, we conjecture that compared to human-written text, **AI-generated text is more likely to mention fewer entities that can be linked to an external KG, and the mentioned entities also show fewer connections that can be supported by the KG.** To verify this conjecture, we conduct a statistical analysis between human-written answers to Reddit ELI5 questions and parallel AI-generated answers from ChatGPT (Wang et al., 2023), and we find that the ChatGPT text, as shown in Figure 1, indeed mentions substantially fewer Wikipedia entities (2.28 ChatGPT vs 5.48 Human on average) with much sparser connections in Wikidata (1.19 ChatGPT vs 3.37 Human on average). A similar conclusion can also be reached for human-written and AI-generate

*Corresponding author: Zhendong Mao.

LREC-COLING 2024, pages 8556–8568
20-25 May, 2024.

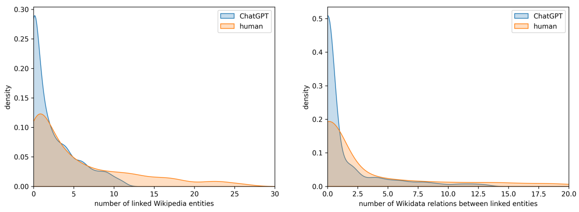

Figure 1: Disparities in external factual structures between human-written answers to Reddit ELI5 questions and parallel AI-generated answers from ChatGPT. Left visualizes the distribution of the number of linked Wikipedia entities, and right the number of Wikidata relations between linked entities in the two types of text.

news in GROVER (Zellers et al., 2019) (see Appendix A for details). These observations indicate that the disparities in external factual structures universally exist irrespective of the generative model (ChatGPT/GROVER) or the text's genre (community question answers/news). Modeling such disparities would definitely be beneficial for detecting AI-generated text.

Motivated by the above observations, we propose **IDEATE**, a novel hierarchical graph network that utilizes both *I*nternal an*D* *E*xternal f*A*ctual s*T*ructur*E*s to detect AI-generated text. Given a piece of input text, IDEATE first represents it as a two-layer graph, with a mention-level subgraph at the bottom and an entity-level subgraph at the top. In the bottom layer subgraph, nodes are automatically extracted entity mentions and edges stand for coreference or cooccurrence relationships between the mentions. This subgraph depicts the factual structure of the mentioned entities reflected in the input text, which we refer to as **internal factual structure**. In the top layer subgraph, coreferential entity mentions are resolved and linked to an external KG. The linked entities are then taken as nodes, with their semantic relations in the KG taken as edges. This subgraph further depicts the factual structure of the mentioned entities reflected in the external KG, which we refer to as **external factual structure**. After that, a hierarchical graph convolution operation (Schlichtkrull et al., 2018) is applied successively on the two subgraphs, where the mention representations output in the bottom layer subgraph are aggregated and taken as the initial entity representations in the top layer subgraph. The output of the hierarchical convolution on the two subgraphs, which naturally encodes the internal and external factual structures, are then concatenated and used for the final detection.

To rigorously evaluate the effectiveness of IDEATE, we conduct extensive experiments on four benchmarking datasets, with AI-generated text from GPT-2 (Radford et al., 2019), GROVER (Zellers et al., 2019), Text-Davinci-003[1] and ChatGPT (OpenAI, 2022), respectively. Experimental results show that IDEATE can consistently outperform state-of-the-art detection methods based on internal textual or structural disparities across all the four datasets, irrespective of which LLM is used for the generation, demonstrating the necessity and superiority of incorporating external factual structures for the detection task. Ablation studies further validate the effectiveness of individual components in IDEATE. Finally, we show that IDEATE is quite robust against targeted adversarial attacks, e.g., those with deliberate entity use created by specific prompts. Our main contributions in this paper are summarized as follows:

- We propose the idea of using both internal and external factual structures for AI-generated text detection. To our knowledge, it is the first time that external factual structures have ever been used for the task.
- We devise a detection method based on hierarchical graph convolution, with mention-level convolution at the bottom to model internal factual structures and entity-level convolution at the top external factual structures.
- Extensive experiments on four benchmarking datasets verify the effectiveness and superiority of the proposed method.

2. Related Work

This section reviews three lines of related work: AI-generated text detection, hallucinations in LLMs, and fact verification.

AI-Generated Text Detection With the explosive development of generative AI, especially LLMs, AI-generated text detection has received increasing attention in the past few years. A straightforward solution is to discover statistical features that discriminate between AI-generated and human-written text, e.g, TF-IDF unigrams/bigrams (Solaiman et al., 2019), per-token likelihood (Gehrmann et al., 2019), perplexity (Ippolito et al., 2020), and curvature of log probability (Mitchell et al., 2023), and then build classifiers with these features (Uchendu et al., 2020; Fröhling and Zubiaga, 2021) or directly perform thresholding on these features (Gehrmann et al., 2019; Mitchell et al., 2023; Su et al., 2023) to detect AI-generated text. Another line of work that fine-tunes a pre-trained language model such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) for the detection task also demonstrates superior performance (Zellers et al., 2019; Fagni et al., 2021; Rodriguez et al., 2022; Guo et al., 2023). In

[1] https://platform.openai.com/docs/models/gpt-3-5

addition to exploiting various explicit or implicit textual disparities, some recent studies (Zhong et al., 2020; Liu et al., 2022) find that AI-generated and human-written text also differ in entity coherence, which can be modeled as internal factual structures and handled by graph neural networks.

Our work is inspired by that of Zhong et al. (2020) and Liu et al. (2022), but differs in that we consider not only internal factual structures, i.e., coreference and cooccurrence relationships between entities mentioned in the given text, but also external factual structures, i.e., semantic relations between the mentioned entities linked to an external KG, and we further integrate the two types of structures into a novel hierarchical graph. Zhong et al. (2020) also considers using external evidences for the detection task. But it simply uses entity representations pre-trained on Wikipedia that contain only textual information without any structure.

There is another stream of work that explores watermarks for AI-generated text to ease their detection (Kirchenbauer et al., 2023; Zhao et al., 2023; Christ et al., 2023; Yoo et al., 2023). But this kind of work usually requires full access to the generative model so as to control its generation behavior for traceability, which is not always feasible and out of the scope of this work. We refer readers to (Jawahar et al., 2020; Tang et al., 2023) for thorough literature reviews.

Hallucinations in LLMs Within the context of NLP, hallucination refers to the generation of text that seems syntactically sound and fluent but is factually incorrect or unfaithful to the provided source input (Maynez et al., 2020; Ji et al., 2023). It occurs when LLMs generate text based on their internal logic, rather than the true context. Hallucination, in particular factually incorrect, to some extent means fewer entities and relations that be linked to and supported by an external KG, which motivates the basic idea of our work. However, current research on hallucination mainly focuses on detecting and mitigating such phenomenon in AI-generated text (Lee et al., 2022; Manakul et al., 2023; Azaria and Mitchell, 2023; Du et al., 2023), while our work uses factuality disparities as evidence to distinguish between AI-generated and human-written text.

Fact Verification Fact verification (or fact checking), is the task to assess the veracity of a given claim with evidence retrieved from plain text (Vlachos and Riedel, 2014; Wang, 2017; Thorne et al., 2018; Wadden et al., 2020). In this task, the claim to be verified is typically a short statement describing a single fact. But in our work, we often have to deal with much longer text containing multiple facts. And instead of assessing the veracity of each separate fact therein, we make use of the whole factual structure to discriminate between AI-generated and human-written text.

3. Our Approach

This section presents our approach, referred to as IDEATE, which uses both *I*nternal an*D* *E*xternal f*A*ctual s*T*ructur*E*s for AI-generated text detection. The key idea of IDEATE is to apply hierarchical convolution successively on a two-layer graph, with mention-level subgraph convolution at the bottom to capture internal factual structures and entity-level subgraph convolution at the top external factual structures. Figure 2 illustrates the overall architecture of IDEATE which consists of four consecutive modules, i.e., context encoding, mention-level subgraph convolution, entity-level subgraph convolution, and final classification. Below we formally introduce the problem formulation, and then describe the four modules in detail.

3.1. Problem Formulation

As in previous work, we formulate AI-generated text detection as a binary classification problem, which determines whether a piece of text is generated by AI tools or written by humans. Specifically, suppose we are given a piece of text consisting of n tokens as input, denoted as $X = \{x_i\}_{i=1}^n$, where x_i stands for the i-th token. And our aim is to predict a binary label $y \in \{0, 1\}$ for X, where $y = 1$ means X is AI-generated and $y = 0$ human-written.

To extract factual structures of the input X, we use an off-the-shelf toolkit to recognize entity mentions in X and link these mentions to Wikipedia entries. Each mention m_j is a contiguous span in the input, along with an entity type tag $t_j \in \mathcal{T}$ and a Wikipedia entry ID $e_j \in \mathcal{E} \cup \{\texttt{NIL}\}$. We typically have $\mathcal{T} = \{\texttt{PER}, \texttt{ORG}, \texttt{LOC}, \texttt{MISC}\}$, which stand for person, organization, location or miscellaneous entities respectively, and NIL indicates that the mention cannot be linked to any Wikipedia entry. In addition, suppose we also get an external KG aligned to Wikipedia, e.g., Wikidata (Vrandečić and Krötzsch, 2014), which means that all entities in the KG have their own Wikipedia IDs. This KG is represented as a group of facts, where each fact is a subject-predicate-object triple (s, r, o) stating that there is a specific relation r between the two entities s and o. Given two entities with their Wikipedia IDs, we can retrieve all relations between them from the KG. By using the entity recognition and linking results and the external KG, we can construct internal and external factual structures, which will be detailed in subsequent sections.

3.2. Context Encoding

This module encodes the input X to produce contextualized token representations. For each token $x_i \in X$, we construct its input embedding as a concatenation of token, entity type, and coreference embeddings:

$$\mathbf{x}_i = E_{token}(x_i) \oplus E_{type}(t_i) \oplus E_{coref}(e_i) \quad (1)$$

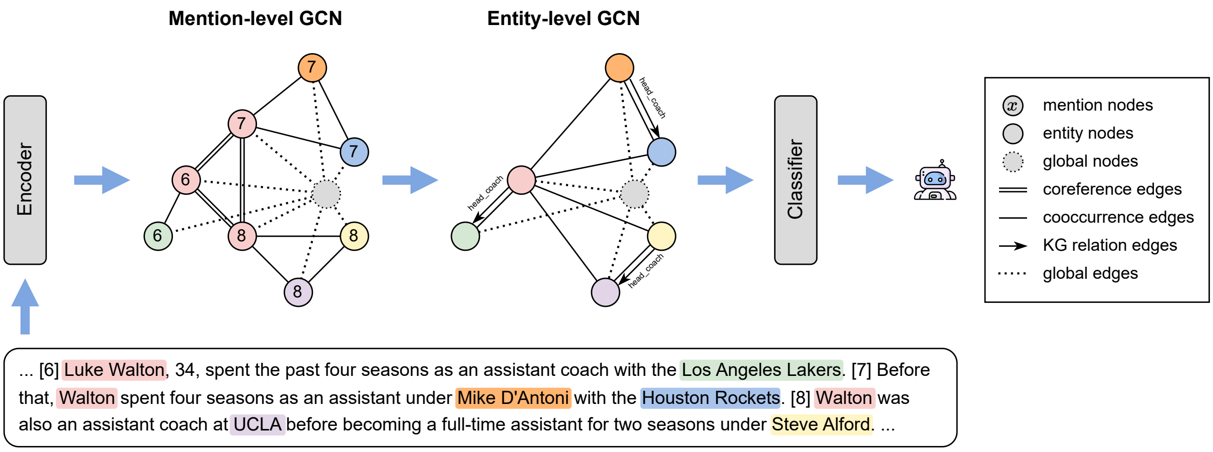

Figure 2: Overview of IDEATE. First, a context encoder consumes the input text to produce contextualized token representations. Then, mention- and entity-level subgraphs are constructed, with a global node introduced into each subgraph to aggregate information from other nodes. Graph convolution is applied consecutively on these two subgraphs, so as to capture internal and external factual structures therein. Finally, a binary classifier concatenates the output of previous modules and uses the concatenation to predict whether the input text is AI-generated or human-written. Different entities are visualized in different colors. The number i in a mention node denotes that it belongs to the i-th sentence.

where $E_{token}(\cdot)$, $E_{type}(\cdot)$, $E_{coref}(\cdot)$ are embedding lookup operations for the three types of embeddings, respectively. t_i is the entity type and e_i the entity ID of the mention to which x_i belongs. Tokens within the same mention share the same entity type and entity ID, and tokens that do not belong to any mention are assigned to a special `None` entity type and `None` entity ID. $\oplus$ denotes the concatenation operation. The constructed input embeddings are fed into an encoder to produce contextualized representations for the tokens:

$$[\mathbf{h}_1, \cdots, \mathbf{h}_n] = \mathrm{Encoder}([\mathbf{x}_1, \cdots, \mathbf{x}_n]) \qquad (2)$$

where $\mathrm{Encoder}$ is a BERT-like pre-trained language model. This encoder typically prepends a special `[CLS]` token at the beginning to aggregate information of the whole input. We use $\mathbf{h}_0$ to denote the output contextualized representation of this special `[CLS]` token.

3.3. Mention-Level Subgraph Convolution

After context encoding, we construct a heterogeneous mention-level subgraph which describes the internal factual structure of the input, and then we apply graph convolution on this subgraph to further model interactions along the factual structure.

Graph Construction Inspired by previous work that uses mention structures for entity relation extraction (Zeng et al., 2020; Xu et al., 2021), we construct a heterogeneous mention-level subgraph consisting of two types of nodes and three types of edges. The two types of nodes are:

- **Mention Node:** Each mention node denotes a particular mention of an entity, which is a contiguous sequence of tokens that appear in the input text.
- **Global Node:** Besides the mention nodes, we introduce a global node which serves as a pivot to interact with other mentions and aggregate the overall input information.

The three types of edges between these nodes are:

- **Coreference Edge:** Mentions referring to the same entity are fully connected with coreference edges. As such, the interactions among different mentions of the same entity could be modeled. We simply take mentions linked to the same Wikipedia entry ID as coreferential mentions, without conducting any further coreference resolution.
- **Cooccurrence Edge:** Two mentions that refer to different entities and co-occur in the same sentence are connected with a cooccurrence edge. In this way, the interactions among different entities could be modeled by cooccurrences of their mentions.
- **Global Edge:** All mentions are connected to the global node with global edges. By introducing such connections the global node could interact with all the mentions and therefore aggregate information from them.

The mention-level subgraph constructed in this way depicts the factual structure of the mentioned entities reflected in the input text, which we refer to as internal factual structure. Note that there are a few previous studies which also employ some kinds of internal factual structures to detect AI-generated text (Zhong et al., 2020; Liu et al., 2022). But their graph structures are different, with only one type of nodes (mention) and one type of edges (two mentions are connected if they have the same surface form or cooccur in the same sentence), which are much simpler than ours.

Graph Convolution We then apply a Relational Graph Convolution Network (R-GCN) (Schlichtkrull et al., 2018) on the mention-level subgraph to propagate and aggregate information along the internal factual structure. To do so, we first construct initial node representations with the encoding output. Each mention node is initialized by averaging the token representations within that mention:

$$\mathbf{p}_u^{(0)} = \frac{1}{t-s+1} \sum_{j=s}^{t} \mathbf{h}_j \tag{3}$$

Here u is a mention ranging from the s-th to t-th token in the input text, and $\mathbf{h}_j$ a token representation within that mention. And the global node is initialized with the `[CLS]` representation $\mathbf{h}_0$.

A multi-layer R-GCN is then applied to update node representations. For each node u whose representation at the ℓ-th layer is denoted as $\mathbf{p}_u^{(\ell)}$, its representation at the $(\ell+1)$-th layer is updated as:

$$\mathbf{p}_u^{(\ell+1)} = \sigma\Big(\sum_{k \in \mathcal{K}_m} \sum_{u' \in \mathcal{N}_k(u)} \mathbf{W}_k^{(\ell)} \mathbf{p}_{u'}^{(\ell)} + \boldsymbol{\alpha}_k^{(\ell)} \Big) \tag{4}$$

where $\mathcal{K}_m$ is the set of different edge types in the mention-level subgraph and $|\mathcal{K}_m| = 3$, $\mathcal{N}_k(u)$ is the set of nodes connected to u with edges of the k-th type, $\mathbf{W}_k^{(\ell)}, \boldsymbol{\alpha}_k^{(\ell)}$ are learnable parameters for the k-th type at the ℓ-th layer, and σ is an activation function (e.g., ReLU). Here by introducing for each edge type its own parameters, we differentiate propagation over edges of different types.

After L layers of R-GCN, we concatenate the output of each layer to form the final representation of each mention node u:

$$\mathbf{m}_u = \mathbf{p}_u^{(0)} \oplus \mathbf{p}_u^{(1)} \oplus \cdots \oplus \mathbf{p}_u^{(L)} \tag{5}$$

By applying the L layers of R-GCN on the mention-level subgraph, we have the internal factual structure encoded into the final node representations.

3.4. Entity-Level Subgraph Convolution

We further construct an entity-level subgraph on top of the mention-level one, and apply R-GCN on this subgraph to model external factual structure of the input text as reflected in the external KG.

Graph Construction The entity-level subgraph is also a heterogeneous graph. It consists of two types of nodes:

- **Entity Node:** Each entity node is a Wikipedia entry with its unique ID, constructed by merging all mentions that refer to this entry in the input text.
- **Global Node:** We also introduce a global node that can interact with other entities and aggregate the overall input information.

Edges between these nodes are categorized into three groups:

- **KG Relation Edge:** Two entities that have relations in the external KG are connected with edges of the types of their KG relations. These KG relation edges are directed, from subject entities to object entities. There could be multiple KG relation edges between two entities, each with its own type and direction. With such connections, the dependencies among different entities reflected in the external KG could be modeled.
- **Cooccurrence Edge:** We also use cooccurrence edges to increase the connectivity of the subgraph. Two entities that have any pair of their mentions co-occur in the same sentence are connected with a cooccurrence edge.
- **Global Edge:** The global node is connected to every entity with a global edge, by which it could interact with all the entities and aggregate information from them.

The entity-level subgraph constructed in this way depicts the factual structure of the mentioned entities reflected in the external KG, which we refer to as external factual structure.

Graph Convolution Again, we apply a multi-layer R-GCN on the entity-level subgraph to aggregate information along the external factual structure. For each entity node v in this subgraph, we construct its initial node representation by averaging the mention representations referring to this entity:

$$\mathbf{q}_v^{(0)} = \frac{1}{M_v} \sum_{u=1}^{M_v} \mathbf{m}_u \tag{6}$$

where M_v is the total number of mentions referring to entity v, and $\mathbf{m}_u$ is the output representation of mention u after mention-level subgraph convolution as defined in Eq. (5). And the global node is again initialized with the `[CLS]` representation $\mathbf{h}_0$. These node representations are then updated as:

$$\mathbf{q}_v^{(\ell+1)} = \sigma\Big(\sum_{k \in \mathcal{K}_e} \sum_{v' \in \mathcal{N}_k(v)} \mathbf{M}_k^{(\ell)} \mathbf{q}_{v'}^{(\ell)} + \boldsymbol{\beta}_k^{(\ell)} \Big) \tag{7}$$

where $\mathcal{K}_e$ is the set of different edge types in the entity-level subgraph, $\mathcal{N}_k(v)$ contains nodes connected to v with edges of the k-th type, $\mathbf{M}_k^{(\ell)}, \boldsymbol{\beta}_k^{(\ell)}$

are learnable parameters for the k-th type at the ℓ-th layer. In this subgraph the KG relation edges are directed edges of different types. For each such edge, we introduce an edge of the inverse relation type in the opposite direction, and hence we have $|\mathcal{K}_e| = 2|\mathcal{R}| + 2$, where $\mathcal{R}$ is the number of relation types in the external KG. Then, after L layers of R-GCN, the external factual structure reflected in the entity-level subgraph would be naturally encoded into the final node representations. Besides, as we use the mention representations $\{\mathbf{m}_u\}$ as initialization, the internal factual structure reflected in the mention-level subgraph would also be encoded into the final representations.

3.5. Final Classification

Finally, we concatenate the `[CLS]` representation $\mathbf{h}_0$ after encoding with the global node representations $\mathbf{p}_0^{(L)}$ and $\mathbf{q}_0^{(L)}$ after the two levels of subgraph convolution. The three components aggregate information of the input text from different perspectives: $\mathbf{h}_0$ focuses on linguistic clues in the text itself, while $\mathbf{p}_0^{(L)}$ and $\mathbf{q}_0^{(L)}$ internal and external factuality clues in the mention- and entity-level subgraphs. A concatenation of them will lead to a more comprehensive overall representation of the input text. This concatenation is then fed into a classification layer to perform the detection task:

$$\hat{\mathbf{y}} = \mathrm{softmax}\big(\mathbf{W}\big(\mathbf{h}_0 \oplus \mathbf{p}_0^{(L)} \oplus \mathbf{q}_0^{(L)}\big) + \mathbf{b}\big) \quad (8)$$

where $\mathbf{W}, \mathbf{b}$ are learnable parameters. We use cross entropy as the classification loss to train our model in an end-to-end fashion:

$$\mathcal{L} = y \log \hat{y}_1 + (1 - y) \log \hat{y}_0 \quad (9)$$

where $y \in \{0, 1\}$ is the label, and $\hat{y}_0, \hat{y}_1$ are the two elements of the prediction $\hat{\mathbf{y}} = [\hat{y}_0, \hat{y}_1]$.

4. Experiments

4.1. Datasets & Metrics

We evaluate IDEATE on four diversified publicly available datasets: GROVER (Zellers et al., 2019), WebText (Radford et al., 2019), Reddit-Davinci and Reddit-ChatGPT (Wang et al., 2023). In the four datasets, human-written instances are collected from different sources, e.g., news, community question answering threads, general web contents, and AI-generated instances produced by different LLMs, ranging from GPT-2 to the more powerful ChatGPT. Statistics of the datasets are shown in Table 1.

		Train	Valid	Test
GROVER	AI	5,000	1,000	4,000
	Human	5,000	2,000	4,000
WebText	AI	250,000	5,000	5,000
	Human	250,000	5,000	5,000
Reddit-Davinci	AI	2,000	500	500
	Human	2,000	500	500
Reddit-ChatGPT	AI	2,000	500	500
	Human	2,000	500	500

Table 1: Statistics of the datasets, where only the unpaired setting is considered in GROVER.

- **GROVER**[2] is a dataset consisting of human-written news collected from RealNews, and AI-generated news produced by GROVER-mega (Zellers et al., 2019), a Transformer-based generative model developed for news generation.
- **WebText**[3] is a large dataset released by OpenAI containing general web contents, where human-written instances are collected from WebText, and AI-generated ones are produced by GPT-2 XL-1542M (Radford et al., 2019).
- **Reddit-Davinci**[4] is a dataset built from Reddit, where human-written instances are long-form question answering threads from specific subreddits, and AI-generated instances produced by the OpenAI Text-Davinci-003.
- **Reddit-ChatGPT**[5] is a dataset created in a similar way as Reddit-Davinci, but with its AI-generated instances produced by ChatGPT.

In GROVER and WebText, human-written and AI-generated instances are not paralleled,[6] and generation is conducted via top-p sampling (Holtzman et al., 2020) with $p = 0.96$. We follow the original train/valid/test splits on the two datasets. In the two Reddit datasets, human-written and AI-generated instances are paralleled, and we re-split them into train/valid/test sets according to the recommended ratio, as the original splits are not provided.
As evaluation metrics, we follow prior work and use **ACC** (accuracy of the overall classification), **F1** (F1 score of the AI-generated class), and **AUROC** (area under the receiver operating characteristic curve).

4.2. Comparison Settings

We compare IDEATE with a number of representative state-of-the-art detection methods, which can be categorized into three groups:

[2] https://storage.googleapis.com/grover-models/generation_examples/generator=mega~dataset=p0.96.jsonl

[3] https://openaipublic.azureedge.net/gpt-2/output-dataset/v1/

[4] https://github.com/mbzuai-nlp/M4/blob/main/data/reddit_davinci.jsonl

[5] https://github.com/mbzuai-nlp/M4/blob/main/data/reddit_chatGPT.jsonl

[6] GROVER also has parallel data used along with a paired evaluation mode. We do not use this evaluation mode to make it consistent with the other datasets.

Method	GROVER			WebText			Reddit-Davinci			Reddit-ChatGPT		
	ACC	F1	AUROC	ACC	F1	AUROC	ACC	F1	AUROC	ACC	F1	AUROC
GLTR-BERT	54.26	55.75	56.51	57.48	57.65	61.49	84.10	84.46	92.35	93.00	93.08	98.17
GLTR-GPT	64.50	65.34	70.07	71.88	73.67	79.69	91.00	90.96	96.55	97.90	97.90	99.68
BERT	66.31	71.76	73.47	80.42	82.57	91.62	96.60	96.70	99.81	97.50	97.56	99.97
GROVER	76.33	78.20	84.58	–	–	–	–	–	–	–	–	–
XLNet	80.09	81.10	88.43	86.15	87.19	95.53	97.10	97.18	99.97	98.20	98.23	99.94
RoBERTa	85.78	85.93	93.37	91.98	92.30	98.21	97.70	97.75	99.93	97.40	97.47	99.82
FAST	86.11	86.89	93.18	92.54	92.72	98.37	99.40	99.40	**99.99**	98.40	98.43	**100.00**
IDEATE (ours)	**87.53**	**87.90**	**94.50**	**94.90**	**94.88**	**98.70**	**99.90**	**99.90**	**99.99**	**99.90**	**99.90**	**100.00**

Table 2: Detection performance (%) on the test sets of the four datasets in terms of ACC, F1 and AUROC.

- Statistical methods that construct logistic regression classifiers with features based on the number of tokens in the Top-10, Top-100, Top-1000, and 1000+ ranks from probability distributions predicted by a pre-trained LM. Two variants are considered: **GLTR-BERT** and **GLTR-GPT** (Gehrmann et al., 2019), which use BERT-large-cased (Devlin et al., 2019) and GPT-2-XL (Radford et al., 2019) as the back-end LM respectively.

- Neural methods that directly fine-tune a pre-trained LM with a classification layer on top using the standard cross-entropy loss. Three LMs are considered: **BERT**, **XLNet** (Yang et al., 2019) and **RoBERTa**, all in base size.

- Another neural method **FAST** (Zhong et al., 2020) which uses RoBERTa-base to model textual disparities, and introduces a GCN on top to model disparities in entity coherence.

For the news-style dataset, we further compare with the **GROVER** discriminator, which is a fine-tuned version of its generator. We choose GROVER-base that is of similar size with other neural methods.

4.3. Implementation Details

IDEATE We use the off-the-shelf toolkit BLINK (Wu et al., 2020)[7] to perform named entity recognition and linking. BLINK recognizes mentions of entities of four types: `PER`, `LOC`, `ORG`, and `MISC`, and links recognized mentions to the May 2019 English Wikipedia dump which includes 5.9M entities. It assumes that each recognized mention has a valid gold entity in the Wikipedia dump, and there is no unlinked (`NIL`) mentions. Our initial experiments show that BLINK achieves roughly 80% precision of entity linking on the datasets. After entity linking, we retrieve their relations from Wikidata5M (Wang et al., 2021)[8], a subset of Wikidata consisting of about 21M subject-predicate-object triples w.r.t. 5M entities and 828 relations. Retrieved relations that appear less than 10 times are discarded.

We use RoBERTa-base as the encoder, on top of which we apply two layers of R-GCN in the mention-level subgraph and another two layers of R-GCN in the entity-level subgraph. The R-GCN dimension is set to 808, where the token embedding size is 768, entity type embedding size is 20, and coreference embedding size is 20. The dropout rate is set to 0.4 for the R-GCN layers, and 0.1 for other layers. We use AdamW (Loshchilov and Hutter, 2017) with a learning rate of 1e-6 as the optimizer, and tune the number of epochs in $\{5, 10, 20, 30\}$. The optimal number of epochs is 30 on GROVER and 5 on the other datasets, determined according to the highest ACC on the valid split of each dataset.

Baselines The two statistical methods of GLTR-BERT and GLTR-GPT are implemented using the code provided by Pu et al. (2022).[9] The three fine-tuning based methods BERT, XLNet, and RoBERTa are implemented by ourselves using the Hugging-Face Transformers library. We replicate FAST using the code released by the authors,[10] and directly evaluate GROVER based on the officially released predicted probabilities.[11] In the baseline methods, common hyperparameters are set to the same values as in IDEATE, and other hyperparameters are fixed at their default values as used in the released codes. All experiments are conducted on a single GeForce RTX 3090 GPU with 24G memory.

4.4. Main Results

Table 2 presents the results of different methods on the test sets of the four datasets. From the results, we can see that our approach IDEATE consistently

[7] https://github.com/facebookresearch/BLINK

[8] https://deepgraphlearning.github.io/project/wikidata5m

[9] https://github.com/jmpu/DeepfakeTextDetection

[10] https://github.com/zhongwanjun/FAST

[11] https://storage.googleapis.com/grover-models/generation_examples/generator=mega~discriminator=grover~discsize=base~dataset=p0.96~test-probs.npy

outperforms all the baselines in all metrics on all the four datasets, and the improvements are rather substantial in most cases. Compared to the direct baseline RoBERTa that uses only textual disparities for the detection, IDEATE brings improvements of **+1.75/+2.92/+2.20/+2.50 ACC** on GROVER, WebText, Reddit-Davinci, and Reddit-ChatGPT, respectively; and compared to the best performing baseline FAST that further uses entity coherence (i.e., a kind of internal factual structure), the improvements on the four datasets are **+1.42/+2.36/+0.50/+1.50 ACC**, respectively. These results demonstrate the crucial significance of using factual structures, especially external factual structures, for discriminating AI-generated and human-written text.

Besides, we can see that the detection scores on the two Reddit datasets are much higher than those on GROVER and WebText, indicating that it is in general easier to detect text generated by more advanced LLMs like Text-Davinci-003 and ChatGPT. We look into the cases and find that the two LLMs show superior interaction awareness and summarization ability, and the generated answers differ greatly from human answers in their writing styles. For example, ChatGPT likes to interact with a questioner, by commenting that "Well, that's an interesting question!" or "Oh wow, that's a big question!" before answering the question. It also likes to summarize, e.g., by using "In conclusion" or "Overall", at the end of its answers. These patterns, however, are much less common in human answers (see Appendix B for concrete examples). Such superficial disparities in writing styles are easy to be learned in a supervised fashion. A vanilla RoBERTa can achieve a high ACC of 97.70% on Reddit-Davinci and 97.40% on Reddit-ChatGPT, leaving not much room for improvement. But even so, incorporating factual structures can still bring consistent and meaningful improvements. After incorporating both types of factual structures, IDEATE can nearly perfectly detect AI-generated text on the two Reddit datasets. Figure 3 visualizes ACC of IDEATE and RoBERTa at different training epochs on the two datasets. The results reveal that IDEATE consistently outperforms RoBERTa throughout the entire training process, and it can achieve nearly perfect performance with only about two epochs of training on both datasets.

4.5. Ablation Studies

This section conducts ablation studies to evaluate the impact of different components in IDEATE, by comparing the full model with four variants:

- **w/o mention** which removes the global node representation in the mention-level subgraph (i.e., $\mathbf{p}_0^{(L)}$), and uses only that in the entity-level subgraph concatenated with the `[CLS]` representation (i.e., $\mathbf{h}_0 \oplus \mathbf{q}_0^{(L)}$) for final classification.

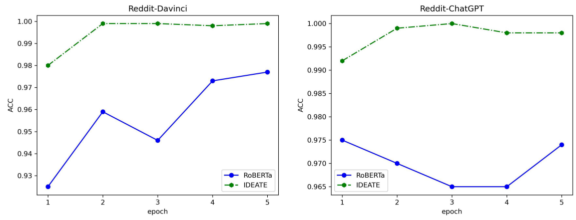

Figure 3: Detection ACC of IDEATE and RoBERTa at different training epochs on the test set of Reddit-Davinci (left) and Reddit-ChatGPT (right).

Method	GROVER		WebText	
	ACC	F1	ACC	F1
IDEATE	**87.53**	**87.90**	**94.90**	**94.88**
w/o mention	87.38	87.48	94.76	94.77
w/o entity	86.73	86.94	94.57	94.51
w/o CLS	86.64	86.99	94.59	94.55
w/o type	87.50	87.85	94.87	94.82
RoBERTa	85.78	85.93	91.98	92.30

Method	Reddit-Davinci		Reddit-ChatGPT	
	ACC	F1	ACC	F1
IDEATE	**99.90**	**99.90**	**99.90**	**99.90**
w/o mention	99.80	99.80	99.80	99.80
w/o entity	**99.90**	**99.90**	**99.90**	**99.90**
w/o CLS	**99.90**	**99.90**	**99.90**	**99.90**
w/o type	99.80	99.80	**99.90**	**99.90**
RoBERTa	97.70	97.75	97.40	97.47

Table 3: Ablation results (%) on the test sets of the four datasets in terms of ACC and F1.

- **w/o entity** which excludes the global node representation in the entity-level subgraph (i.e., $\mathbf{q}_0^{(L)}$), and uses only that in the mention-level subgraph concatenated with the `[CLS]` representation (i.e., $\mathbf{h}_0 \oplus \mathbf{p}_0^{(L)}$) for final classification.
- **w/o CLS** which excludes the `[CLS]` representation (i.e., $\mathbf{h}_0$), and concatenates the global node representations in the two subgraphs (i.e., $\mathbf{p}_0^{(L)} \oplus \mathbf{q}_0^{(L)}$) for final classification.
- **w/o type** which ignores various types of KG relations in the entity-level subgraph, and regards all such relations to be of the same type.

In the four variants, hyperparameters are set in the same way as in the full model. We also include the direct baseline **RoBERTa** for comparison.

Table 3 presents the ablation results on the four datasets, where the observations on GROVER and WebText are different from those on Reddit-Davinci and Reddit-ChatGPT. On GROVER and WebText, we observe that (i) The w/o mention, w/o entity, and w/o CLS variants all underperform the full model, indicating that disparities in text, internal and external factual structures are all useful for detecting

AI-generated text, and removing any of them would cause a decrease in detection performance. (ii) Though underperforming the full model, the three variants all surpass RoBERTa, verifying the necessity of introducing internal and external factual structures for the detection. (iii) The w/o entity variant performs worse than the w/o mention variant, indicating that between the two types of factual structures, the external ones are more crucial than the internal ones. (iv) The w/o type variant slightly underperforms the full model, verifying the benefits of differentiating KG relation types in the entity-level subgraph. On the two Reddit datasets, all the four variants substantially outperform RoBERTa, indicating that the baseline can be improved as long as factual structures are introduced, no matter they are internal or external. And the four variants perform very closely to the full model, with no substantial differences in their performance. The reason may be that the two Reddit datasets contain AI-generated text relatively easy to be detected, leaving no much room for improvement, so the distinctions between different improving strategies are not obvious.

4.6. Further Analyses

IDEATE with Random Factual Structures Although IDEATE achieves consistent and meaningful improvements, it introduces additional parameters to model the internal and external factual structures. To eliminate the effect of having more parameters, we consider another three IDEATE variants:

- **Random M** where the true coreference and cooccurrence edges in the mention-level subgraph are replaced with the same number of coreference/cooccurrence edges introduced between randomly selected mentions.
- **Random E** where the true KG relation edges in the entity-level subgraph are replaced with the same number of Wikidata relation edges introduced between randomly selected entities.
- **Random M+E** which is a combination of Random M and Random E.

The three variants keep the number of connections and parameters the same with IDEATE, but without utilizing the correct internal and/or external factual structures. As shown in Figure 4 (left), the three variants in general perform on par with RoBERTa and lag far behind IDEATE, indicating that the improvement of IDEATE indeed comes from the correct factual structures rather than just an increased number of parameters.

IDEATE against Adversarial Attacks We finally evaluate the ability of IDEATE to resist adversarial attacks, which are constructed by prompting LLMs to generate text with deliberate entity use. To do so, we add "please include as many Wikipedia-related

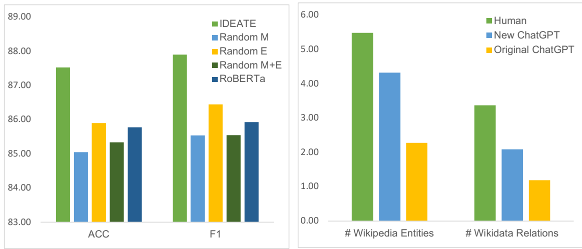

Figure 4: Performance (%) of IDEATE with random factual structures on the test set of GROVER (left). The average number of linked Wikipedia entities and Wikidata relations in between in the three types of Reddit answers (right).

entities as possible" into the original instructions and query ChatGPT with new instructions to generate answers for Reddit questions (see Appendix C for a concrete example). We find that, as shown in Figure 4 (right), adjusting instructions indeed enables to generate text mentioning more Wikipedia entities and more Wikidata relationships, but the quantity still lags behind human text, and our conjecture still holds. We evaluate the ability of IDEATE (trained from the original Reddit-ChatGPT dataset) to detect those new ChatGPT answers. The detection ACC drops slightly from 99.90% to 99.60%, indicating that IDEATE is quite robust against adversarial attacks with deliberate entity use.

5. Conclusion

This paper presents IDEATE, a new approach that utilizes internal and external factual structures in addition to textual disparities to detect AI-generated text from human-written text. Compared to previous methods which mainly focus on internal evidences contained in the text itself, the most distinctive feature of the new approach is to further use external structural evidences implicated in an established KG for the detection. Specifically, IDEATE consists of a mention-level subgraph at the bottom to model internal factual structures (factual structures of mentioned entities reflected in the input text), and an entity-level subgraph at the top to model external factual structures (factual structures of mentioned entities reflected in the external KG). A hierarchical graph convolution is then applied successively on the two subgraphs, and the output representations (along with the overall context representation) are aggregated and used for final classification. In this manner, disparities reflected in both internal and external factual structures, as well as those reflected in the text, would all be taken into account for the detection. Experimental results on four benchmarking datasets verify the effectiveness and superiority of the proposed method in detecting text generated by different LLMs, ranging from GPT-2 to the recent more powerful ChatGPT.

6. Acknowledgements

We would like to thank the anonymous reviewers for their insightful and valuable suggestions, which help improve the quality of this paper. This work is supported by the National Key Research and Development Program of China (No.2022YFC3302101), and the National Natural Science Foundation of China (No.62376033, 62232006, 62222212).

7. Bibliographical References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Proceedings of the 34th International Conference on Advanced Information Networking and Applications*, pages 1341–1354.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when its lying. *arXiv preprint arXiv:2304.13734*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*.

Nassim Dehouche. 2021. Plagiarism in the age of massive generative pre-trained transformers (GPT-3). *Ethics in Science and Environmental Politics*, 21:17–23.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: Tackling GPT-2, GPT-3 and GROVER. *PeerJ Computer Science*, 7:e443.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.

Alberto Giaretta and Nicola Dragoni. 2020. Community targeted phishing: A middle ground between massive and spear phishing through natural language generation. In *Proceedings of 6th International Conference in Software Engineering for Defence Applications*, pages 86–93.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822.

Ganesh Jawahar, Muhammad Abdul-Mageed, and VS Laks Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and

Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. CoCo: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. *https: //openai.com/blog/chatgpt*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2022. Deepfake text detection: Limitations and opportunities. *arXiv preprint arXiv:2210.09421*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting LLM-generated texts. *arXiv preprint arXiv:2303.07205*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: A large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8384–8395.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7534–7550.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 422–426.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6397–6407.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14149–14157.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023. Robust natural language watermarking through invariant features. *arXiv preprint arXiv:2305.01904*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in Neural Information Processing Systems*, 32.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1640.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. In *Proceedings of the 40th International Conference on Machine Learning*.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2461–2470.

A. Conjecture Verification on GROVER

To verify the conjecture that AI-generated text tends to mention fewer entities and entity relations that can be linked to and supported by an external KG, we conduct the same analysis on GROVER as we have done on Reddit-ChatGPT, and reach a similar conclusion. On GROVER, AI-generated text mentions 16.57 Wikipedia entities and 6.86 Wikidata relations on average, fewer than 19.10 Wikipedia entities and 7.91 Wikidata relations on average in human-written text. The kernel density distribution curves of the two statistics are provided in Figure 5.

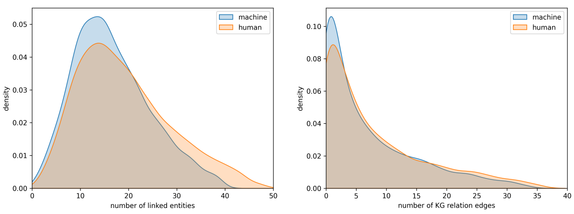

Figure 5: Disparities in external factual structures between human-written and AI-generated news in GROVER. Left shows the distribution of the number of linked Wikipedia entities, and right the number of Wikidata relations between linked entities.

Note that compared to Reddit-ChatGPT (Figure 1), instances from GROVER in general mention more Wikipedia entities and Wikidata relations. This is because GROVER is a dataset consisting of AI-generated or human-written news, which naturally includes more entities and entity relations.

B. ChatGPT Answers on Reddit ELI5

Table 4 presents three concrete examples of ChatGPT answers to Reddit ELI5 questions, where each answer is produced by querying ChatGPT with a specific instruction, e.g., "I will ask you a question. For this question, provide me more than 200 words

Well, that's an interesting question. In general, it does seem to be the case that American society has gotten less formal over the last hundred years or so. [...] **Overall**, while it's true that many aspects of American social culture have become less formal over time, there are still plenty of examples of ways in which things have become more formal. Whether it's in the workplace or in social situations, there are still situations where formality is expected and appreciated.
Oh wow, that's a big question! Um, I'm not sure exactly what the classical-era Chinese thought about the far west, but I know they probably had some ideas and stories about it. [...] I don't know all the details, but it's cool to think about what people in China might have imagined about places far away. **Maybe if I learn more about it, I can tell you more next time!**
Yes, there have been symbols in history that have been viewed similarly to the way modern Western Europe/America views the Swastika. [...] **In conclusion**, while the Swastika is undoubtedly one of the most potent symbols of hate in modern history, it is not the only symbol that has been co-opted by hate groups to spread their message of intolerance and exclusion.

Table 4: Concrete examples of ChatGPT answers to Reddit ELI5 questions.

answer in a Reddit answering style. <Question>". We can see that ChatGPT shows superior interaction awareness and summarization ability. It likes to interact with a given instruction, e.g., by commenting that "Well, that's an interesting question!" or "Oh wow, that's a big question!" before answering the question. And it also likes to summarize, e.g., by using "In conclusion" or "Overall" at the end of its answers. These patterns, however, are much less common in human answers.

C. Adversarial Attacks with ChatGPT

IDEATE assumes that AI-generated text mentions fewer entities that can be linked to an external KG, and also fewer entity relationships that can be supported by the KG. So we can create adversarial attacks by prompting LLMs to generate text with deliberate entity use. Specifically, we add "please include as many Wikipedia-related entities as possible" into the original instructions and query ChatGPT with new instructions to generate answers for Reddit questions. Table 5 provides two answers to the same question, generated by querying ChatGPT with the original and new instructions, respectively. We can see that the answer generated by the new instruction indeed mentions more Wikipedia entities. But even so, the quantity still lags behind human text (see Figure 4 (right) for a comparison).

Original Instruction: I will ask you a question. For this question, provide me more than 200 words answer in a 10 year-old child's voice. [...]
Original ChatGPT Answer: Oh wow, that's a big question! Um, I'm not sure exactly what the classical-era **Chinese** thought about the far west, but I know they probably had some ideas and stories about it. So like, **China** is in the east, right? And the far west is like, really far away from there. I don't think they had like, internet or airplanes back then, so it was probably hard to know what was going on in those places. I'm pretty sure they knew about **Persia**, because that's where silk came from and they traded with them. But places like **Egypt**, **Greece**, **Scythia**, and **Rome** were probably not as well known. Maybe they heard some stories from travelers who had been there, or maybe they made up their own stories about what was out there. I think they might have thought that people in these places were really different from them, because they had different customs and spoke different languages. Maybe they thought they had cool buildings and clothes and stuff like that. Or maybe they thought they were really scary and dangerous, like monsters or something. I don't know all the details, but it's cool to think about what people in **China** might have imagined about places far away. Maybe if I learn more about it, I can tell you more next time!
New Instruction: I will ask you a question. For this question, provide me more than 200 words answer in a 10 year-old child's voice. **Please include as many Wikipedia-related entities as possible.** [...]
New ChatGPT Answer: Oh, wow! That's a big question! So, back in the olden times in **China**, people didn't have the internet or fancy things like that. They didn't know everything about faraway places like **Egypt**, **Greece**, **Scythia**, or **Rome**. But they did have some ideas and stories. You see, in **China**, they had this thing called the **Silk Road**. It was a super important trade route that connected **China** with other countries. Along this road, **Chinese** people would meet travelers from different lands and hear tales of faraway places. But since they didn't go there themselves, sometimes their knowledge wasn't always right. They had some myths and stories about these far west places. For example, they thought **Greece** was a land of magical creatures and brave warriors. They also believed that **Egypt** had giant pyramids and mysterious pharaohs. As for **Scythia**, they thought it was a cold and snowy place with fierce warriors riding horses. And **Rome**, oh boy! They thought it was an amazing city with mighty emperors and grand buildings. Now, I must tell you, my friend, I'm just a little kid, so I might not know all the real facts about what the classical-era **Chinese** actually believed. But if you want to learn more about it, you can check out this cool website called **Wikipedia**. It has lots of information about history and stuff like that!

Table 5: Two ChatGPT answers to the same Reddit ELI5 question, generated via the original and new instructions, respectively.