Understanding Privacy Risks of Embeddings Induced by Large Language Models

Zhihao Zhu, Ninglu Shao, Defu Lian, Chenwang Wu, Zheng Liu, Yi Yang, Enhong Chen

ABSTRACT

Large language models (LLMs) show early signs of artificial general intelligence but struggle with hallucinations. One promising solution to mitigate these hallucinations is to store external knowledge as embeddings, aiding LLMs in retrieval-augmented generation. However, such a solution risks compromising privacy, as recent studies experimentally showed that the original text can be partially reconstructed from text embeddings by pre-trained language models. The significant advantage of LLMs over traditional pre-trained models may exacerbate these concerns. To this end, we investigate the effectiveness of reconstructing original knowledge and predicting entity attributes from these embeddings when LLMs are employed. Empirical findings indicate that LLMs significantly improve the accuracy of two evaluated tasks over those from pre-trained models, regardless of whether the texts are in-distribution or out-of-distribution. This underscores a heightened potential for LLMs to jeopardize user privacy, highlighting the negative consequences of their widespread use. We further discuss preliminary strategies to mitigate this risk.

ACM Reference Format:
Zhihao Zhu, Ninglu Shao, Defu Lian, Chenwang Wu, Zheng Liu, Yi Yang, Enhong Chen. 2024. Understanding Privacy Risks of Embeddings Induced by Large Language Models. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Large language models [10, 27] have garnered significant attention for their exceptional capabilities across a wide range of tasks like natural language generation [7, 37], question answering [35, 55], and sentiment analysis [5, 52].

Nonetheless, it's observed that large language models can confidently assert non-existent facts during their reasoning process. For example, Bard, Google's AI chatbot, concocted information in the first demo that the James Webb Space Telescope had taken the first pictures of a planet beyond our solar system [12]. Such a hallucination problem [31, 54] of large language models is a significant barrier to artificial general intelligence [22, 44]. A primary strategy for tackling the issue of hallucinations is to embed external knowledge in the form of embeddings into a vector database [19, 23], making them accessible for retrieval augmented generation by large language models [6, 13].

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

An embedding model [43, 49] encodes the original objects' broad semantic information by transforming the raw objects (e.g., text, image, user profile) into real-valued vectors of hundreds of dimensions. The advancement of large language models enhances their ability to capture and represent complex semantics more effectively, such that an increasing number of businesses (e.g., OpenAI [40] and Cohere [1]) have launched their embedding APIs based on large language models. Since embeddings are simply real-valued vectors, it is widely believed that it is challenging to decipher the semantic information they contain. Consequently, embeddings are often viewed as secure and private, as noted by [50], leading data owners to be less concerned about safeguarding the privacy of embeddings compared to raw external knowledge. However, in recent years, multiple studies [30, 38, 50] have highlighted the risk of embeddings compromising privacy. More specifically, the pre-trained LSTM (Long Short-Term Memory) networks [25] or other language models can recover parts of the original texts and author information from text embeddings, which are generated by open-source embedding models. Although current studies have exposed security weaknesses in embeddings, the effects of large language models on the privacy of these embeddings have not been fully explored. A pressing issue is whether LLMs' emergent capabilities enable attackers to more effectively decipher sensitive information from text embeddings. This issue is driven not only by the proliferation of large language models but also by the availability of the embedding APIs based on LLMs, which permits attackers to gather numerous text-embedding pairs to build their attack models.

To this end, we establish a comprehensive framework that leverages a large language model (LLM) to gauge the potential privacy leakage from text embeddings produced by the open-sourced embedding model. From a security and privacy perspective, LLM serves as the attacker, and the embedding model acts as the target, while the goal is to employ the attack model to retrieve sensitive and confidential information from the target model. Specifically, our approach begins with fine-tuning attack models to enable text reconstruction from the outputs of the target model. Following this, we assess the privacy risks of embeddings via two types of attack scenarios. On the one hand, we recover the texts from their embeddings in both in-distribution and out-of-distribution scenarios. On the other hand, we identify certain private attributes of various entities in the original text (such as birthdays, nationalities, criminal charges, etc.) and predict these attributes from the text embeddings. This prediction is determined by the attribute that exhibits the highest cosine similarity between the text embedding and the corresponding attribute embedding. Consequently, this method does not necessitate training with supervised data. Should the target embedding model decline to generate embeddings for attributes with extremely brief texts described (1-2 words) out of embedding stealing concerns, we introduce an external embedding model that acts as a proxy to project the original text and the attribute value

in the same embedding space. Specifically, this external model is tasked with embedding the attribute and the text reconstructed by the attack model, the latter being derived from text embeddings produced by the target embedding model.

The evaluation of text reconstruction reveals that 1) a larger attack language model, when fine-tuned with a sufficient amount of training data, is capable of more accurately reconstructing texts from their embeddings in terms of metrics like BLEU [41], regardless of whether the texts are in-distribution or out-of-distribution; 2) in-distributed texts are more readily reconstructed than out-of-distributed texts, with the reconstruction accuracy for in-distributed texts improving as the attack model undergoes training with more data; 3) the attack model can improve the reconstruction accuracy as the expressiveness of the target embedding models increases.

The evaluation of attribute prediction demonstrates that 1) attributes can be predicted with high accuracy across various domains, including encyclopedias, news, medical, and legislation. This means the attacker is capable of inferring details like a patient's health condition, a suspect's criminal charges, and an individual's birthday from a set of seemingly irrelevant digital vectors, highlighting a significant risk of privacy leakage; 2) generally speaking, enlarging the scale of the external/target embedding model substantially enhances the accuracy of attribute prediction; 3) when the target model denies embedding services for very short texts, the most effective approach using reconstructed text by the attack model can achieve comparable performance to using original text, when the target model and the external embedding model are configured to be the same.

From the experiments conducted, we find that knowledge representations merely through numerical vectors encompass abundant semantic information. The powerful generative capability of large language models can continuously decode this rich semantic information into natural language. If these numerical vectors contain sensitive private information, large language models are also capable of extracting such information. The development trend of large language models is set to increase these adverse effects, underscoring the need for our vigilance. Our research establishes a foundation for future studies focused on protecting the privacy of embeddings. For instance, the finding that accuracy in text reconstruction diminishes with increasing text length indicates that lengthening texts may offer a degree of privacy protection. Furthermore, the ability of the attack model to reconstruct out-of-distributed texts points towards halting the release of original texts associated with released embeddings as a precaution.

2 MAIN RESULTS

2.1 Fine-Tuning the Attack Model

Table 1: Sizes of attack models and embedding models

Attack Model	GPT2	GPT2-Large	GPT2-XL
#parameters	355M	744M	1.5B
Target Model	SimCSE	BGE-Large-en	E5-Large-v2
#parameters	110M	326M	335M

We employ pre-trained GPT2 [45] of varying sizes as the attacking model to decipher private information from embeddings produced by the target embedding models such as SimCSE [21], BGE-Large-en [53], and E5-Large-v2 [51]. We hypothesize that larger embedding models, due to their capacity to capture more information, are more likely to be exposed to a greater privacy risk. Consequently, all these models are designated as the target models, and their numbers of parameters are detailed in Table 1. It's important to note that we treat the target model as a black box, meaning we do not have access to or knowledge of its network architecture and parameters. Figure 1 showcases the fine-tuning process for the attack model. Initially, the example text "David is a doctor." is inputted into the target embedding model to generate its respective embedding. This embedding is then used as the input for the attack model, which aims to reconstruct the original text based solely on this embedding. An EOS (End-of-Sentence) token is appended to the text embedding to signal the end of the embedding input. The attacker's training goal is to predict the t-th token of the original text based on text embedding and the preceding (t-1) tokens of the original text. In the testing phase, the attacker employs beam search [18] to progressively generate tokens up to the occurrence of the EOS. Once the attack model has been fine-tuned, we evaluate the privacy risks of embeddings through two distinct attack scenarios: text reconstruction and attribute prediction.

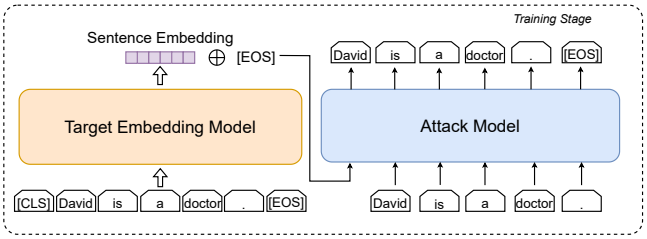

Figure 1: The fine-tuning of the foundation attack model. Initially, the attacker queries the target embedding model to convert the collected text into text embeddings. To signify the completion of the embedding input, an EOS (End-of-Sentence) token is appended to the text embedding. Next, the attacker selects the pre-trained GPT2 model as the attack model and uses the collected text and corresponding text embeddings as a dataset to train the attack model. When a text embedding is input, the attack model is trained to sequentially reconstruct the related original text.

2.2 Evaluation of Text Reconstruction Settings

For each text in the test set, we reconstruct it using the attack model based on its embedding generated by the target model. To evaluate the reconstruction accuracy, we employ two metrics: BLEU (Bilingual Evaluation Understudy) [41] and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [33]. Specifically, we utilize BLEU-1 and ROUGE-1, which are based solely on unigrams, as they yield better results compared to BLEU and ROUGE based on other n-grams [4]. These metrics gauge the similarity between the

Table 2: Reconstruction attack performance against different embedding models on the wiki dataset. The best results are highlighted in bold. $*$ represents that the advantage of the best-performed attack model over other models is statistically significant (p-value < 0.05).

Training data		wiki-small		Wiki-large		Wiki-xl	
Target model	Attack model	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1
SimCSE	GPT2	0.3184±0.0010	**0.3212±0.0010***	**0.4512±0.0010***	**0.4961±0.0011***	0.4699±0.0016	0.5256±0.0010
	GPT2_large	0.2996±0.0014	0.2913±0.0012	0.4349±0.0013	0.4678±0.0009	0.5293±0.0010	0.5930±0.0009
	GPT2_xl	**0.3196±0.0011***	0.3112±0.0013	0.4455±0.0015	0.4833±0.0010	**0.5331±0.0011***	**0.5987±0.0007***
BGE-Large-en	GPT2	**0.3327±0.0014***	**0.3288±0.0011***	0.4173±0.0016	0.4483±0.0009	0.4853±0.0011	0.5337±0.0016
	GPT2_large	0.2935±0.0013	0.2783±0.0011	0.4446±0.0011	0.4788±0.0006	0.5425±0.0012	0.5998±0.0010
	GPT2_xl	0.3058±0.0019	0.3043±0.0012	**0.4689±0.0011***	**0.5057±0.0007***	**0.5572±0.0008***	**0.6151±0.0007***
E5-Large-v2	GPT2	**0.3329±0.0013***	**0.3341±0.0012***	0.4838±0.0005	0.5210±0.0008	0.5068±0.0016	0.5522±0.0014
	GPT2_large	0.3093±0.0009	0.2875±0.0012	0.4700±0.0011	0.4990±0.0010	0.5679±0.0011	0.6220±0.0011
	GPT2_xl	0.3083±0.0013	0.3017±0.0013	**0.4993±0.0013***	**0.5274±0.0011***	**0.5787±0.0007***	**0.6378±0.0009***

original and reconstructed texts. Given that the temperature setting influences the variability of the text produced by GPT, a non-zero temperature allows for varied reconstructed texts given the same text embedding. We calculate the reconstruction accuracy across 10 trials to obtain mean and standard error for statistical analysis. Based on these outcomes, we compare the performance of various attack and target configurations using a two-sided unpaired t-test [16]. The evaluation is conducted on seven datasets, including wiki [47], wiki-bio [29], cc-news [24], pile-pubmed [20], triage [32], cjeu-terms [9], and us-crimes [9]. Details and statistics for these datasets are presented in Table 5.

Results of In-Distributed Texts

We create three subsets from the wiki dataset of varying sizes (i.e., wiki-small, wiki-large, and wiki-xl) and use these subsets to fine-tune the attack models, resulting in three distinct versions of the attack model. The performance of these attack models is then assessed using held-out texts from the wiki dataset. The experimental results presented in Table 2 illustrate that *the size of the training datasets and the models have a considerable influence on the reconstruction accuracy*. To elaborate, regardless of the attack model employed, *it's found that larger embedding models, such as BGE-Large-en and E5-Large-v2, enable more effective text reconstruction compared to others like SimCSE*. This is attributed to the strong expressivity of the large target embedding model, which allows it to retain more semantic information from the original text, proving beneficial for the embedding's application in subsequent tasks. Moreover, *provided that the attack model is adequately fine-tuned and the embedding model is expressive enough, the accuracy of text reconstruction improves as the size of the attack model increases*. This improvement is reflected in the table's last two columns, showing higher accuracy as the attack model progresses from GPT2 to GPT2_large, and finally to GPT2_xl, attributed to the improved generative capabilities of larger models. Additionally, *adequately fine-tuning the large language models is a prerequisite for their effectiveness in text reconstruction tasks. When the embeddings are less informative, fine-tuning the attack model demands a larger amount of training data*. This is highlighted by the lesser performance of GPT2_xl compared to GPT2 when fine-tuning with wiki-small, which reverses after fine-tuning with Wiki-xl. Moreover, GPT2_xl outperforms GPT2 in reconstructing text from SimCSE's embeddings as the fine-tuning dataset shifts from "Wiki-large" to "Wiki-xl".

To summarize, **a larger attack model, when fine-tuned with an increased amount of training data, is capable of more accurately reconstructing texts from the embeddings generated by target embedding models with higher expressivity**. Hence, a straightforward approach for safeguarding privacy involves not disclosing the original dataset when publishing its embedding database. Nonetheless, it remains to be investigated whether an attack model, fine-tuned on datasets with varying distributions, would be effective.

Results of Out-of-Distributed Texts

To address the unresolved question, we assume that the attacker model is trained on the Wiki dataset with more rigorous descriptions of world knowledge, yet the texts used for testing do not originate from this dataset. This implies that the distribution of the texts used for testing differs from that of the texts used for training. We assess the reconstruction capability of this attack model on sample texts from six other datasets: wiki_bio, cc_news, pile_pubmed, triage, us_crimes, and cjeu_terms. The results presented in Table 3 show that *the attack model retains the capability to accurately reconstruct texts from the embeddings, even with texts derived from different distributions than its training data*. In greater detail, the best reconstruction accuracy of the attack model on texts from 5 out of 6 datasets is equal to or even exceeds that of a model fine-tuned on wiki-small, a relatively small subset of Wikipedia. As a result, if we release embeddings for the wiki_bio, cc_news, pile_pubmed, us_crimes, and cjeu_terms datasets, an attack model fine-tuned on the Wiki-xl dataset can extract semantic information from them with relatively high confidence. This suggests that simply withholding the original raw data does not prevent an attacker from reconstructing the original text information from their released embeddings.

To understand which kind of text data is more easily recovered from text embedding based on the attack model fine-tuned with Wiki-xl, we also analyze the similarity between the six evaluation datasets and Wiki based on previous works [28]. The results reported in figure 2 show that *texts from evaluation datasets with higher similarity to the training data are reconstructed more accurately*. To elaborate, Wiki-bio, which compiles biographies from Wikipedia, shares the same origin as the training dataset. Despite covering different content, the language style is very similar. Consequently, the quality of the attack's text reconstruction for this

Table 3: Reconstruction attack performance on different datasets. The best results are highlighted in bold. $*$ represents that the advantage of the best-performed attack model over other models is statistically significant (p-value < 0.05).

Test dataset		wiki_bio		cc_news		pile_pubmed	
Target model	Attack model	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1
SimCSE	GPT2	0.5428±0.0025	0.5596±0.0022	0.3881±0.0012	0.4487±0.0011	0.3623±0.0015	0.3976±0.0014
	GPT2_large	0.5859±0.0011	0.6272±0.0015	0.4314±0.0015	0.5020±0.0014	0.4054±0.0009	0.4427±0.0011
	GPT2_xl	**0.5878±0.0012***	**0.6329±0.0016***	**0.4355±0.0010***	**0.5084±0.0008***	**0.4133±0.0013***	**0.4505±0.0010***
BGE-Large-en	GPT2	0.4773±0.0025	0.5327±0.0020	0.3906±0.0014	0.4314±0.0013	0.3297±0.0013	0.3581±0.0011
	GPT2_large	0.5497±0.0018	0.6015±0.0009	0.4339±0.0009	0.4867±0.0012	0.3819±0.0013	0.4074±0.0014
	GPT2_xl	**0.5652±0.0030***	**0.6200±0.0015***	**0.4480±0.0008***	**0.5038±0.0006***	**0.3955±0.0019***	**0.4218±0.0017***
E5-Large-v2	GPT2	0.5312±0.0015	0.5532±0.0014	0.4065±0.0009	0.4428±0.0010	0.3673±0.0012	0.3995±0.0009
	GPT2_large	0.5695±0.0014	0.6206±0.0017	0.4521±0.0013	0.5006±0.0013	0.4174±0.0006	0.4523±0.0007
	GPT2_xl	**0.5823±0.0012***	**0.6354±0.0017***	**0.4645±0.0014***	**0.5173±0.0015***	**0.4316±0.0009***	**0.4683±0.0009***
Test dataset		triage		us_crimes		cjeu_terms	
Target model	Attack model	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1	BLEU-1	ROUGE-1
SimCSE	GPT2	0.0932±0.0007	0.1555±0.0010	0.3092±0.0006	0.3238±0.0008	0.3485±0.0010	0.3646±0.0009
	GPT2_large	0.1188±0.0006	0.1756±0.0006	**0.3268±0.0006***	0.3406±0.0003	0.3755±0.0012	**0.3954±0.0008***
	GPT2_xl	**0.1299±0.0006***	**0.2004±0.0010***	0.3226±0.0005	**0.3429±0.0006***	**0.3739±0.0007***	0.3914±0.0006
BGE-Large-en	GPT2	0.0828±0.0007	0.1072±0.0008	0.2705±0.0008	0.2834±0.0010	0.3271±0.0008	0.3378±0.0009
	GPT2_large	**0.1376±0.0008***	**0.1659±0.0009***	0.2755±0.0008	0.3155±0.0004	**0.3639±0.0016***	**0.3785±0.0014***
	GPT2_xl	0.1232±0.0008	0.1640±0.0007	**0.2775±0.0008***	**0.3207±0.0006***	0.3522±0.0012	0.3794±0.0012
E5-Large-v2	GPT2	0.1451±0.0007	0.2265±0.0008	0.2825±0.0009	0.2938±0.0010	0.3418±0.0018	0.3668±0.0016
	GPT2_large	**0.2272±0.0004***	**0.3172±0.0007***	0.3066±0.0007	0.3308±0.0006	0.3621±0.0013	0.3980±0.0012
	GPT2_xl	0.2230±0.0008	0.3177±0.0007	**0.3164±0.0007***	**0.3459±0.0005***	**0.3688±0.0009***	**0.4101±0.0011***

dataset is the highest. In other words, simply withholding the original text associated with embeddings does not adequately safeguard sensitive information from being extracted, as fine-tuning the attack model with datasets that are similar in terms of language style or content can elevate the risk of privacy breaches.

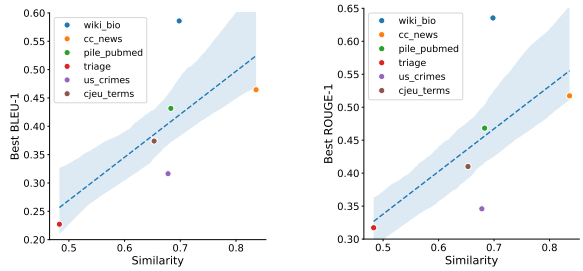

Figure 2: The similarity between the evaluation datasets and the Wiki dataset v.s. the best reconstruction performance.

The disclosure of even a small collection of original texts significantly amplifies the risk, as illustrated in figure 3 which presents results from the pile-pubmed dataset where the attack model undergoes further fine-tuning with these original texts. The availability of more original texts linked to the embedding directly correlates with an increased risk of sensitive information leakage. Specifically, the BLEU-1 score for GPT2-xl against the BGE-large-en embedding model sees a 2% increase when the attack model is supplemented with 10k original texts. Notably, even with the use of a limited amount of target data (1K samples), the improvements in BLEU-1 score are considerable.

In summary, **concealing the datasets of published embeddings does not effectively prevent information leakage from these embeddings. This is because their underlying information can still be extracted by an attack model that has been fine-tuned on datasets of similar style or content. Furthermore, revealing even a small number of samples can significantly improve the extraction accuracy.**

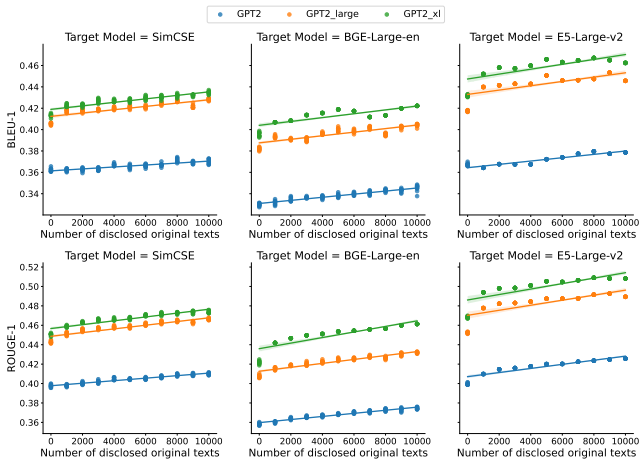

Figure 3: Impact of disclosed original texts volume on text reconstruction accuracy. Each column represents a different target embedding model. The first and second rows represent the reconstruction performance concerning the BLEU-1 and ROUGE-1 metrics, respectively.

Results of Varying Text Lengths

Given GPT's capability to generate outputs of varying lengths with consistent meanings, exploring how text length impacts reconstruction quality becomes pertinent. With the average text length in the Wiki dataset being 21.32, as noted in Table 5, we selected three subsets: Wiki-base, Wiki-medium, and Wiki-long, each with 5,000

samples but average lengths of 20, 40, and 80 words, respectively. Among them, Wiki-medium and Wiki-long are created by extending texts in Wiki-base via the GPT4 API [3] accessible by OpenAI.

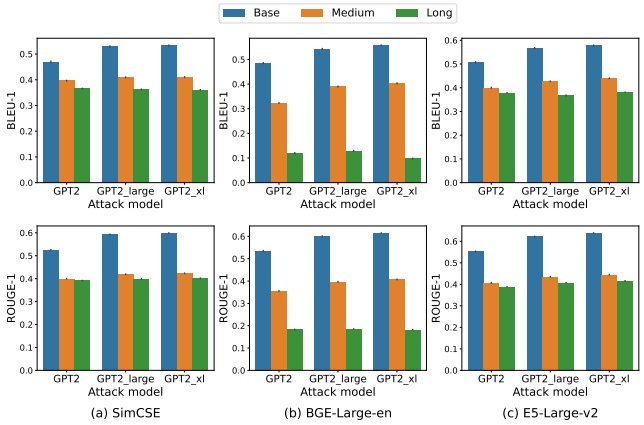

Figure 4: Influence of the text length. Error bars represent the mean reconstruction accuracy with 95% confidence intervals obtained from 10 independent trials, and each column corresponds to a different target embedding model.

The results of text reconstruction are depicted in figure 4. It's evident that embeddings from shorter texts are more susceptible to being decoded, posing a greater risk to privacy. For example, the ROUGE-1 score of GPT2-xl on BGE-Large-en fell by over 43.3% as the test text length increased from Wiki-base to Wiki-long. This decline can be attributed to the fixed length of text embeddings, which remain constant regardless of the original text's length. Consequently, embeddings of longer texts, which encapsulate more information within the same embedding size, make accurate text recovery more formidable. Therefore, **extending the original texts could potentially fortify the security of released embeddings**.

2.3 Evaluation of Sensitive Attribute Prediction

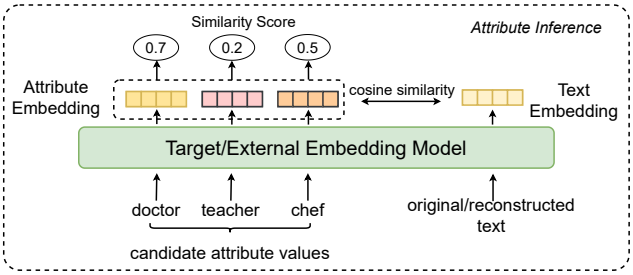

Figure 5: The inference framework of sensitive attributes. The attacker employs the same embedding model to convert original text and candidate attributes into embeddings. The attacker then identifies the attribute that exhibits the highest cosine similarity between its embedding and text embedding as sensitive information of the original text.

Settings

In contrast to text reconstruction, our primary concern is whether the attacker can extract specific sensitive information from the text embeddings. The task of predicting sensitive attributes more clearly illustrates the issue of privacy leakage through embeddings, compared to the task of reconstructing text. Initially, we pinpoint sensitive or crucial data within the datasets: wiki-bio, triage, cjeu-terms, and us-crimes. In particular, we examined patient dispositions and blood pressure readings in the triage dataset, and in the wiki-bio dataset, we focused on individuals' nationality, birthdate, and profession. Additionally, we looked into criminal charges in the us_crimes dataset and considered legal terminology as important information in the cjeu_terms dataset. Given the extensive variety of attributes and the scarcity of labeled data for each, it's impractical for the privacy attacker to train a dedicated model for each sensitive attribute, unlike the approach taken in previous studies [50]. Therefore, we predicted the sensitive information from text embedding by selecting the attribute that exhibits the highest cosine similarity between text embedding and its embedding. This approach is effective across various attributes and does not necessitate training with supervised data. However, a challenge arises because the text describing the attribute is often very brief (sometimes just a single word), and the target embedding model may refuse to produce embeddings for such short texts due to concerns about embedding theft [17, 34]. As a result, it becomes difficult to represent the original text and the attribute value within the same embedding space. To overcome this, we introduce an external embedding model to serve as an intermediary. This external model is responsible for embedding both the attribute and the reconstructed text, which has been derived from text embeddings by the attack model. Consequently, texts and attributes are embedded within the same space via reconstructing texts from the text embeddings, allowing for accurate similarity measurement. The overall process for inferring attributes is depicted in figure 5. The outcomes of the attribute inference attack using various methods are presented in Table 4. The last row of this table, which relies on the similarity between attributes and original texts, lacks randomness in its measurement due to the direct comparison method employed. In contrast, the preceding rows, which are based on reconstructed texts, introduce randomness into the similarity measurement. This variability stems from employing a non-zero temperature setting, allowing for multiple independent text generations to produce diverse outputs.

Results

The findings presented in Table 4 reveal that *sensitive information can be accurately deduced from text embeddings without the necessity for any training data, highlighting a significant risk of privacy leakage through embeddings.* Specifically, attributes such as nationality and occupation can be inferred with high precision (an accuracy of 0.94) even when using texts that have been reconstructed. This level of accuracy is attributed to the embeddings' ability to capture the rich semantic details of texts coupled with the attack model's strong generative capabilities. Remarkably, the accuracy of predictions made using an external embedding model on reconstructed texts is on par with those made using the target embedding model on original texts in several cases. For an equitable comparison, the

Table 4: Attribute inference attack performance (accuracy) when using bge-large-en as the embedding model. The best results are highlighted in bold. $*$ denotes that the advantage of the best-performed attack model over other models is statistically significant (p-value < 0.05). The last row provides experimental results where the attacker has unrestricted access to the target embedding model and compares the embeddings of original text and candidate attributes. Other results are based on the attacker's usage of an external embedding model to calculate the similarities between reconstructed text and candidate attributes.

Dataset		wiki-bio			triage		cjeu_terms	us_crimes
Similarity Model	Attack Model	nationality	birth_date	occupation	disposition	blood_pressure	legal_term	criminal_charge
SimCSE	GPT2	0.875±0.005	0.546±0.008	0.869±0.005	0.506±0.001	0.135±0.004	0.395±0.010	0.377±0.003
	GPT2_large	0.882±0.003	0.579±0.011	**0.878±0.004**	0.506±0.001	0.119±0.006	0.405±0.006	0.396±0.004
	GPT2_xl	**0.886±0.003***	**0.596±0.009***	**0.878±0.005**	**0.514±0.002***	**0.137±0.003**	**0.428±0.008***	**0.407±0.004***
BGE-Large-en	GPT2	0.927±0.005	0.525±0.009	0.913±0.006	**0.505±0.002**	0.195±0.004	0.496±0.008	0.470±0.005
	GPT2_large	0.937±0.005	0.560±0.009	0.919±0.003	0.504±0.001	0.238±0.005	0.538±0.004	0.510±0.005
	GPT2_xl	**0.941±0.003***	**0.581±0.008***	**0.922±0.005**	0.504±0.001	**0.254±0.007***	**0.551±0.006***	**0.527±0.003***
E5-Large-v2	GPT2	0.932±0.004	0.670±0.008	0.927±0.005	0.514±0.002	0.206±0.005	0.492±0.010	0.465±0.004
	GPT2_large	0.940±0.003	0.729±0.008	0.938±0.003	**0.521±0.001**	0.229±0.003	0.544±0.007	0.506±0.006
	GPT2_xl	**0.941±0.003**	**0.756±0.008***	**0.940±0.005**	**0.521±0.002**	**0.230±0.004**	**0.545±0.006**	**0.524±0.006***
BGE-Large-en	None	0.953	0.742	0.950	0.538	0.431	0.764	0.716

external embedding model was set to be identical to the target embedding model, although, in practice, the specifics of the target embedding model might not be known. The inference accuracy improves when employing a larger attack model for text reconstruction and a more expressive embedding model. This outcome aligns with observations from text reconstruction tasks. Although the accuracy of attribute prediction with reconstructed text falls short in some instances compared to using original texts, the ongoing advancement in large language models is rapidly closing this gap. Hence, the continuous evolution of these models is likely to escalate the risks associated with privacy breaches, emphasizing the critical need for increased awareness and caution in this domain.

3 DISCUSSIONS AND LIMITATIONS

This study delves into the implications of large language models on embedding privacy, focusing on text reconstruction and sensitive information prediction tasks. Our investigation shows that as the capabilities of both the sophisticated attack foundation model and the target embedding model increase, so does the risk of sensitive information leakage through knowledge embeddings. Furthermore, the risk intensifies when the attack model undergoes fine-tuning with data mirroring the distribution of texts linked to the released embeddings.

To protect the privacy of knowledge embeddings, we propose several strategies based on our experimental findings:

- **Cease the disclosure of original texts tied to released embeddings**: Preventing the attack model from being fine-tuned with similar datasets can be achieved by introducing imperceptible noise into the texts or embeddings. This aims to widen the gap between the dataset of original or reconstructed texts and other analogous datasets.
- **Extend the length of short texts before embedding**: Enhancing short texts into longer versions while preserving their semantic integrity can be accomplished using GPT-4 or other large language models with similar generative capacities.
- **Innovate new privacy-preserving embedding models**: Develop embedding models capable of producing high-quality text embeddings that are challenging to reverse-engineer into the original text. This entails training models to minimize the cloze task loss while maximizing the reconstruction loss.

However, our study is not without limitations. Firstly, due to substantial training expenses, we did not employ an attack model exceeding 10 billion parameters, though we anticipate similar outcomes with larger models. Secondly, while we have quantified the impact of various factors such as model size, text length, and training volume on embedding privacy, and outlined necessary guidelines for its protection, we have not formulated a concrete defense mechanism against potential embedding reconstruction attacks. Currently, effective safeguards primarily rely on perturbation techniques and encryption methods. Perturbation strategies, while protective, can compromise the embedding's utility in subsequent applications, necessitating a balance between security and performance. Encryption methods, though secure, often entail considerable computational demands. Future work will explore additional factors influencing embedding privacy breaches and seek methods for privacy-preserving embeddings without sacrificing their utility or incurring excessive computational costs.

4 METHODS

4.1 Preliminary

Prior to delving into the attack strategies, we will commence with the introduction of the language models and evaluation datasets.

Language Model

A language model is a technique capable of assessing the probability of a sequence of words forming a coherent sentence. Traditional language models, such as statistical language models [8, 46] and grammar rule language models [26, 48], rely on heuristic methods to predict word sequences. While these conventional approaches may achieve high predictive accuracy for limited or straightforward sentences within small corpora, they often struggle to provide precise assessments for the majority of other word combinations.

Table 5: Statistics of datasets

Dataset	Domain	#sentences	avg. sentence len
wiki	General	4,010,000	21.32
wiki_bio	General	1,480	22.19
cc_news	News	5,000	21.39
pile_pubmed	Medical	5,000	21.93
triage	Medical	4668	54.10
cjeu_terms	Legal	2127	118.96
us_crimes	Legal	4518	181.28

With increasing demands for the precision of language model predictions, numerous researchers have advocated for neural network-based language models [15, 36] trained on extensive datasets. The performance of neural network-based language models steadily increases as model parameters are increased and sufficient training data is received. Upon reaching a certain threshold of parameter magnitude, the language model transcends previous paradigms to become a Large Language Model (LLM). The substantial parameter count within LLMs facilitates the acquisition of extensive implicit knowledge from corpora, thereby fostering the emergence of novel, powerful capabilities to handle more complex language tasks, such as arithmetic operations [39] and multi-step reasoning [42]. These capabilities have enabled large language models to comprehend and resolve issues like humans, leading to their rising popularity in many aspects of society. However, the significant inference capacity of large language models may be leveraged by attackers to reconstruct private information from text embeddings, escalating the risk of privacy leakage in embeddings.

Datasets for Evaluation

In this paper, we assess the risk of embedding privacy leaks on seven datasets, including wiki [47], wiki-bio [29], cc-news [24], pile-pubmed [20], triage [32], cjeu-terms [9], and us-crimes [9]. Wiki collects a large amount of text from the Wikipedia website. Since it has been vetted by the public, the text of the wiki is both trustworthy and high-quality. Wiki-bio contains Wikipedia biographies that include the initial paragraph of the biography as well as the tabular infobox. CC-News (CommonCrawl News dataset) is a dataset containing news articles from international news websites. It contains 708241 English-language news articles published between January 2017 and December 2019. Pile-PubMed is a compilation of published medical literature from Pubmed, a free biomedical literature retrieval system developed by the National Center for Biotechnology Information (NCBI). It has housed over 4000 biomedical journals from more than 70 countries and regions since 1966. Triage records the triage notes, the demographic information, and the documented symptoms of the patients during the triage phase in the emergency center. Cjeu-term and us-crimes are two datasets from the legal domain. Cjeu-term collects some legal terminologies of the European Union, while us-crimes gathers transcripts of criminal cases in the United States.

For wiki, cc_news, and pile_pubmed datasets, we randomly sample data from the original sets instead of using the entire dataset because the original datasets are enormous. To collect the sentence texts for reconstruction, we utilize en_core_web_trf [2], an open-source tool, to segment the raw data into sentences. Then, we cleaned the data and filtered out sentences that were too long or too short. The statistical characteristics of the processed datasets are shown in Table 5.

Ethics Statement

For possible safety hazards, we abstained from conducting attacks on commercial embedding systems, instead employing open-source embedding models. Additionally, the datasets we utilized are all publicly accessible and anonymized, ensuring no user identities are involved. To reduce the possibility of privacy leakage, we opt to recover more general privacy attributes like occupation and nationality in the attribute prediction evaluation rather than attributes that could be connected to a specific individual, such as phone number and address. The intent of our research is to highlight the increased danger of privacy leakage posed by large language models in embeddings, suggest viable routes to safeguard embedding privacy through experimental analysis, and stimulate the community to develop more secure embedding models.

4.2 Threat Model

Attack Goal

This paper primarily investigates the extraction of private data from text embeddings. Given that such private data often includes extremely sensitive information such as phone numbers, addresses, and occupations, attackers have ample motivation to carry out these attacks. For instance, they might engage in illegal activities such as telecommunications fraud or unauthorized selling of personal data for economic gain. These practices pose significant threats to individual privacy rights, potentially lead to economic losses, increase social instability, and undermine trust mechanisms.

Attack Knowledge

To extract private information from text embeddings, attackers require some understanding of the models responsible for generating these embeddings. Intuitively, the more detailed the attacker's knowledge of the target embedding model, including its internal parameters, training specifics, etc., the more potent the attack's efficacy. However, in real-world scenarios, to safeguard their intellectual property and commercial interests, target models often keep their internal information confidential, providing access to users solely through a query interface O. Based on the above considerations, this study assumes that attackers only possess query permissions to the target embedding model, which responds with the corresponding text embedding based on the text inputted by the attacker. This query process does not reveal any internal information about the model. Furthermore, with the increasing popularity of large language models, numerous companies are opting to release their anonymized datasets for academic research purposes. Therefore, this study also assumes that attackers have the capability to gather specific open-source data (e.g., Wikipedia in our evaluation) and leverage interactions with target models to acquire associated text embeddings. Clearly, such low-knowledge attack settings help simulate real attack scenarios and more accurately assess the risks of embedding privacy leaks.

4.3 Attack Methodology

Attack Model Construction

Attackers reconstructed original text from text embedding by training an attack model. However, they lack prior knowledge about which architecture should be used for the attack model. When using neural networks as the attack model, it is challenging to decide which neural network architecture should be employed. If the architecture of the attack model is not as expressive and complex as the embedding model, then it is difficult to ensure that it can extract private information.

Considering the exceptional performance of large language models across various domains, particularly in text comprehension [11] and information extraction [14], employing them as attack models could be an appropriate choice. Based on these considerations, this study utilizes GPT-2 models of varying sizes as attack models, training them to reconstruct the original text from the embeddings.

Training set generation. We start by retrieving the text embeddings from the collected open-source data D using the query interface O of the target embedding model.

$$e_w = O(w) \tag{1}$$

Then, we construct the training dataset D_{train} for the attack model based on these embeddings and corresponding texts.

$$D_{train} = \{(d_w, w) | w \in D, d_w = (e_w, < EOS >)\} \tag{2}$$

where EOS(End-of-Sentence) token is a special token appended to the text embedding to signify the end of the embedding input.

The attack model performs the opposite operation compared to the embedding model: for each sample $(d_w, w) \in D_{train}$, the attack model strives to recover the original text w from the embedding e_w.

Attack model training. For each embedding input d_w, the attack model is trained to output the first token of the original text. Subsequently, when the attacker provides d_w along with the first $i-1$ tokens of the original text, the attack model is tasked with predicting the $i-th$ token of the original text with utmost accuracy. Formally, for a sample (d_w, w), the text reconstruction loss of the attack model is as follows:

$$L_\theta(d_w, w) = -\sum_{i=1}^{l} \log P(x_i | d_w, w_{<i}, \theta) \tag{3}$$

where $w_{<i} = (x_1, ..., x_{i-1})$ represents the first $i-1$ tokens of the original text w. l denotes the length of w and θ represents the parameter of the attack model. Therefore, the training loss for the attack models is the sum of the text reconstruction loss across all samples in the training dataset:

$$L_\theta = -\sum_{(d_w, w) \in D_{train}} L_\theta(d_w, w) \tag{4}$$

To evaluate the effectiveness of the attack models, we employ two tasks: text reconstruction and attribute prediction.

Text Reconstruction Task

Reconstructed text generation. In the text reconstruction task, the attack models aim to generate reconstructed text that closely resembles the original text. When generating a reconstructed text of length l, the attacker aims for the generated text to have the highest likelihood among all candidate texts, which is formalized as follows:

$$w^* = \underset{w=\{x_1,...,x_l\}}{\arg\max} P(x_1, ..., x_l | d_w, \theta) \tag{5}$$

However, on the one hand, the number of candidate tokens for x_i in the reconstructed text w is large, often exceeding $10,000$ in our experiments. On the other hand, the number of candidate texts exponentially increases as the text length l grows. As a result, it becomes infeasible to iterate through all candidate texts of length l and select the one with the highest likelihood as the output. A viable solution is greedy selection, which involves choosing the candidate token with the highest likelihood while progressively constructing the reconstructed text.

$$x_i^* = \underset{x_i}{\arg\max} P(x_i | d_w, \theta, w_{<i}^*) \tag{6}$$

$$w^* = (x_1^*, ..., x_l^*) \tag{7}$$

However, this approach may easily lead the generation process into local optima. To enhance the quality of the reconstructed text and improve generation efficiency, we employ beam search in the text reconstruction task. In beam search, if the number of beams is k, the algorithm maintains k candidates with the highest generation probability at each step. Specifically, in the initial state, for a given input text embedding e_w, the attack model first records k initial tokens with the highest generation likelihood (ignoring eos token) as candidate texts of length 1 (C_1^*).

$$C_1^* = \underset{C_1 \subset \mathcal{X}, |C_1|=k}{\arg\max} \sum_{x \in C_1} P(x | d_w, \theta) \tag{8}$$

where $\mathcal{X}$ represents the set of all tokens in the attack model.

Subsequently, these k initial tokens are combined with any token in $\mathcal{X}$ to create a text set of length 2. The attack model then iterates through these texts of length 2 and selects k texts with the highest generation likelihood as candidate texts of length 2 (C_2^*).

$$C_2^* = \underset{|C_2|=k}{\arg\max} \sum_{(x_1,x_2) \in C_2} P(x_1, x_2 | d_w, \theta) \tag{9}$$

$$C_2 \subset \{(x_1, x_2) | x_1 \in C_1^*, x_2 \in \mathcal{X}\} \tag{10}$$

This process continues, incrementing the text length until the model generates an EOS token signaling the end of the generation process.

Evaluation metric. We adopt BLEU-1 and ROUGE-1 to evaluate the reconstruction performance of the attack model, measuring how similar the reconstructed text w' is to the original text w. The formulas for these two metrics are as follows:

$$\text{BLEU-1} = BP \cdot \frac{\sum_{x \in \text{set}(w')} \min(\text{count}(x, w), \text{count}(x, w'))}{\sum_{x \in \text{set}(w')} \text{count}(x, w')} \tag{11}$$

$$\text{ROUGE-1} = \frac{\sum_{x \in \text{set}(w)} \min(\text{count}(x, w), \text{count}(x, w'))}{\sum_{x \in \text{set}(w)} \text{count}(x, w)} \tag{12}$$

where $\text{set}(w)$ and $\text{set}(w')$ are the sets of all tokens in w and w'. $\text{Count}(x, w)$ and $\text{count}(x, w')$ are the number of times x appears in w and w', respectively. The brevity penalty (BP) is used to prevent short sentences from getting an excessively high BLEU-1 score. BLEU-1 primarily assesses the similarity between the reconstructed text and the original text, whereas ROUGE-1 places greater emphasis on the completeness of the reconstruction results and whether

the reconstructed text can encompass all the information present in the original text.

Dataset similarity calculation. We assess the similarity between the evaluation datasets and the wiki dataset based on a simple character n-gram comparison [28]. Specifically, we employed the 5000 commonly used 4-gram characters in English as the feature set $\mathcal{F}$ of the dataset. Each dataset is then represented as a 5000-dimensional feature vector.

$$\overrightarrow{F_D} = [\text{count}(f_1, D), ..., \text{count}(f_{5000}, D)] \tag{13}$$

where $f_i \in \mathcal{F}$ is a 4-gram character, and $\text{count}(f_i, D)$ is the number of times f_i appears in dataset D. Finally, we calculate the Spearman correlation coefficient between the feature vectors of the two datasets to quantify their similarity.

$$\text{Sim}(D_1, D_2) = \text{Spearman}(\overrightarrow{F_1}, \overrightarrow{F_2}) \tag{14}$$

where $\overrightarrow{F_1}$ and $\overrightarrow{F_2}$ are feature vectors of D_1 and D_2, respectively. The Spearman coefficient ranges from -1 to 1, where a higher value indicates a greater similarity between the two corresponding datasets.

Attribute Prediction Task

In the attribute prediction task, this study focuses on the attacker's ability to extract private information from the original text. We chose several private attributes from four datasets and evaluated the attack model's ability to infer the precise values of these private attributes from the released text embedding. For example, in the wiki-bio dataset, occupation is chosen as a private attribute. The attacker attempts to ascertain that the original text contains the private message "doctor" by using the embedding of the sentence "David is a doctor."

Instead of training the attack model to perform the attribute prediction task, this study utilizes the embedding similarity between the text and the attribute value to determine the suggested attribute value of the original text. Its rationality is that the text contains relevant information about sensitive attributes, so their embeddings should be similar. The ideal approach would be to determine based on the similarity between embeddings of the original text and sensitive attribute embeddings. However, this poses challenges: (1) The original text is unknown. (2) Privacy attributes are often short texts, and in most cases, consist of only one word; such frequent anomalous (short) inputs might be considered malicious attempts and rejected. Therefore, this study (1) uses reconstructed text instead of the original text, and (2) employs an open-source external embedding model as a proxy to obtain embeddings instead of using the target embedding model. It's worth noting that this study did not directly search for privacy attributes in the reconstructed text due to potential inaccuracies in reconstructing privacy attributes, such as missing tokens or reconstructing synonyms of the attributes.

Specifically, the attacker initially acquires embeddings of the reconstructed text and sensitive attribute from their proxy embedding model, subsequently computing the cosine similarity between them, and ultimately selecting the attribute with the highest similarity as the prediction result. Formally, the attacker infers the sensitive attribute w_v as follows:

$$w_v = \arg\max_{v \in \mathbb{C}_v} \frac{e_{w'} \cdot e_v}{|e_{w'}||e_v|} \tag{15}$$

where $\mathbb{C}_v$ is the set of candidate attribute values, w_v is the predicted attribute value of the original text w. $e_{w'}$ and e_v are the embedding vectors of reconstructed text w' and attribute value v with the aid of the external embedding model, respectively. We employ accuracy as the metric to evaluate the performance of the attack model on the attribute prediction task.

REFERENCES

[1] [n. d.]. The Cohere Platform. https://docs.cohere.com/docs/.
[2] [n. d.]. English transformer pipeline. https://huggingface.co/spacy/en_core_web_trf/.
[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[4] Raed Z Al-Abdallah and Ahmad T Al-Taani. 2017. Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Computer Science* 117 (2017), 30–37.
[5] Barakat AlBadani, Ronghua Shi, and Jian Dong. 2022. A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM. *Applied System Innovation* 5, 1 (2022), 13.
[6] Konstantinos Andriopoulos and Johan Pouwelse. 2023. Augmenting LLMs with Knowledge: A survey on hallucination prevention. *arXiv preprint arXiv:2309.16459* (2023).
[7] Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. *arXiv preprint arXiv:2307.07312* (2023).
[8] Jerome R Bellegarda. 2004. Statistical language model adaptation: review and perspectives. *Speech communication* 42, 1 (2004), 93–108.
[9] Ilias Chalkidis, Nicolas Garneau, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. 2023. LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development. *arXiv preprint arXiv:2305.07507* (2023).
[10] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109* (2023).
[11] Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09530* (2023).
[12] Martin Coulter and Greg Bensinger. 2023. Alphabet shares dive after Google AI chatbot Bard flubs answer in ad. *Reuters* (2023).
[13] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092* (2023).
[14] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications* 15, 1 (2024), 1418.
[15] Wim De Mulder, Steven Bethard, and Marie-Francine Moens. 2015. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language* 30, 1 (2015), 61–98.
[16] David B Duncan. 1975. T tests and intervals for comparisons suggested by the data. *Biometrics* (1975), 339–359.
[17] Oluwaseyi Feyisetan and Shiva Kasiviswanathan. 2021. Private release of text embedding vectors. In *Proceedings of the First Workshop on Trustworthy Natural Language Processing*. 15–27.
[18] Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806* (2017).
[19] Tal Friedman and Guy Broeck. 2020. Symbolic querying of vector spaces: Probabilistic databases meets relational embeddings. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 1268–1277.
[20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027* (2020).
[21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.
[22] Ben Goertzel. 2014. Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence* 5, 1 (2014), 1.
[23] Rentong Guo, Xiaofan Luan, Long Xiang, Xiao Yan, Xiaomeng Yi, Jigao Luo, Qianya Cheng, Weizhi Xu, Jiarui Luo, Frank Liu, et al. 2022. Manu: a cloud native vector database management system. *arXiv preprint arXiv:2206.13843* (2022).
[24] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A Generic News Crawler and Extractor. In *Proceedings of the 15th International Symposium of Information Science* (Berlin). 218–223. https://doi.org/10.5281/zenodo.4120316

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[26] Daniel Jurafsky, Chuck Wooters, Jonathan Segal, Andreas Stolcke, Eric Fosler, Gary Tajchaman, and Nelson Morgan. 1995. Using a stochastic context-free grammar as a language model for speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. IEEE, 189–192.

[27] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* (2023).

[28] Adam Kilgarriff. 2001. Comparing corpora. *International journal of corpus linguistics* 6, 1 (2001), 97–133.

[29] Rémi Lebret, David Grangier, and Michael Auli. 2016. Generating Text from Structured Data with Application to the Biography Domain. *CoRR* abs/1603.07771 (2016). arXiv:1603.07771 http://arxiv.org/abs/1603.07771

[30] Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence Embedding Leaks More Information than You Expect: Generative Embedding Inversion Attack to Recover the Whole Sentence. *arXiv preprint arXiv:2305.03010* (2023).

[31] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 6449–6464.

[32] Yutong Li, Ruoqing Zhu, Annie Qu, Han Ye, and Zhankun Sun. 2021. Topic modeling on triage notes with semiorthogonal nonnegative matrix factorization. *J. Amer. Statist. Assoc.* 116, 536 (2021), 1609–1624.

[33] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[34] Xuebo Liu, Derek F Wong, Yang Liu, Lidia S Chao, Tong Xiao, and Jingbo Zhu. 2019. Shared-private bilingual word embeddings for neural machine translation. *arXiv preprint arXiv:1906.03100* (2019).

[35] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems* 35 (2022), 2507–2521.

[36] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5528–5531.

[37] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *Comput. Surveys* 56, 2 (2023), 1–40.

[38] John X Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. *arXiv preprint arXiv:2310.06816* (2023).

[39] Matteo Muffo, Aldo Cocco, and Enrico Bertino. 2023. Evaluating transformer language models on arithmetic operations using number decomposition. *arXiv preprint arXiv:2304.10977* (2023).

[40] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005* (2022).

[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[42] Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014* (2023).

[43] Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. 2023. A Survey of Text Representation and Embedding Techniques in NLP. *IEEE Access* (2023).

[44] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572, 7767 (2019), 106–111.

[45] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[46] Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proc. IEEE* 88, 8 (2000), 1270–1278.

[47] Diego Saez-Trumper and Miriam Redi. 2020. Wikimedia Public (Research) Resources. In *Companion Proceedings of the Web Conference 2020*. 311–312.

[48] Hassan Sawaf, Kai Schütz, and Hermann Ney. 2000. On the use of grammar based language models for statistical machine translation. In *Proceedings of the Sixth International Workshop on Parsing Technologies*. 231–241.

[49] S Selva Birunda and R Kanniga Devi. 2021. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020* (2021), 267–281.

[50] Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 377–390.

[51] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[52] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* 55, 7 (2022), 5731–5780.

[53] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[54] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* (2023).

[55] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).