Reproduction of 'Text Embeddings Reveal (Almost) As Much As Text'

Ryan Ott
University of Amsterdam
Amsterdam, The Netherlands

Jesse Wonnink
University of Amsterdam
Amsterdam, The Netherlands

Richter van Emmerik
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

This study reproduces the findings of the original '*Text Embeddings Reveal (Almost) As Much As Text*'[10], which investigates privacy risks associated with text embeddings, by showing that it is possible to reconstruct an original text from its embedding. By implementing an iterative correction mechanism and using a transformer-based encoder-decoder architecture, Vec2Text achieves significant accuracy in embedding inversion, evidenced by high reconstruction BLEU and Token F1 scores in both in-domain and out-of-domain evaluations. This reproduction validates the claims of the original paper regarding the performance on in-domain datasets, and questions some of the generalisability claims. The reproduction also analyses critical performance factors, such as beam width and sequence length. Additionally, this study extends the analysis by evaluating computational trade-offs and performance on different length sequences than the one for which the corrector model was originally trained. Limitations, including a discussion on valid metrics, and token-length constraints, highlight areas for improvement for the development of vector-to-text models. These findings reinforce the importance of balancing privacy risks, computational efficiency, and performance in text embedding systems.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols, Privacy protections**; • **Information systems → Information retrieval**; • **Computing Methodologies → Natural language processing**.

KEYWORDS

Text Embeddings, Embedding Inversion, Privacy Risks in NLP, Beam Search Optimization , Sequence Reconstruction ,Controlled Generation, Reproducibility

ACM Reference Format:
Ryan Ott, Jesse Wonnink, and Richter van Emmerik. . Reproduction of 'Text Embeddings Reveal (Almost) As Much As Text'. In *Proceedings of Reproduction (UvA IR2).* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

Text embedding vectors are dense, high-dimensional numerical representations that capture semantic meaning, enabling machines to perform various tasks more effectively in areas such as personalised recommendations, search engines, and information retrieval (IR). Embeddings are seeing increased use with the rise of vector databases [13], which are integral to architectures like retrieval-augmented generation in large language models, which leverage external knowledge to enhance their accuracy and reduce hallucinations. Users of these databases maintain a level of privacy by not sending directly interpretable raw text, rather querying the database using embeddings.

However, a growing concern is that of embedding inversion: recreating the original input information from its embedding. This process has already been successfully demonstrated in the context of image embeddings [4]. The problem of reconstructing text sequences, where changes between two sequences are discrete words, rather than, for example, continuous pixel values, has proved to be more difficult.

In [10], the authors claim to have solved this problem. The paper introduces **Vec2Text**, a novel multistep method that:

(1) Iteratively corrects and re-embeds text to reconstruct the original.
(2) Uses controlled generation techniques to optimise the similarity between reconstructed text embeddings and the original embeddings.

The original work outlines that Vec2Text is capable of accurately reconstructing text sequences in sensitive domains, such as medical patient records, potentially posing privacy risks. To investigate and build upon these findings, this paper is structured as follows: First, we discuss (**Related Work**) in section 1.1. We discuss the model in-depth in 2 (**Background**. In Section 3 (**Claims and Scope**), we detail the claims of the original paper and define the scope of our reproduction. In Section 4 (**Reproductions**), we outline the methodology and model settings we employ to replicate and extend the results. In Section 5 (**Results**), we present our reproduction results. In Section 6 (**Extension Results**), we discuss results from additional experiments that go beyond the scope of the original work. Finally, in Section 7 (**Conclusion**), we present our conclusions, and discuss limitations and future directions for research.

1.1 Related Work

The concept is derived from the image inversion settings [9][4]. This domain is considered more tractable since images exist in a 'continuous' domain, whereas the 'discrete' nature of text presents greater challenges for training.

Vector-to-text black box attacks have been attempted on several occasions, including an earlier work [18] that demonstrated a proof-of-concept with Bag of Words text inversion. Subsequently, [6] conducted an intriguing study exploring the intersection of copyrighted text and vector-to-text conversion by training an inverse model of BERT [3]. Their approach does not incorporate iterative refinement and does not conduct out-of-domain evaluations. Furthermore, [7] introduced a transformer-based architecture lacking iterative refinement, resulting in outcomes that were comparable

UvA IR2, Dec 22, Amsterdam, NL
2024.

to the baseline but inferior to those reported in the original paper of this reproduction work.

A study [22], employing a method analogous to that of the original paper, utilizes a GPT-2 [15]-based attacker model. This model is trained on a combination of embeddings and input sentences to predict the responses of a target model. The authors make a persuasive argument for focusing metrics on entity attribute inversion outcomes rather than whole-sentence conclusions, positing that these are likely the more significant components of the extracted information for hypothetical attackers.

In a subsequent development of the original study, [5] investigated a more challenging scenario, characterized by the availability of a limited training dataset and an unknown underlying model. Their proposed method comprises two main phases; initially, they emulate the unknown embedder model and subsequently generate additional samples to train the attack model to effectively target the unknown model. This advancement is potentially concerning, given its positive correlation with the scale of the embedder model, leading to increased efficacy when targeting models with higher parameters.

A general framework to which vector-to-text models should conform is proposed in [2]. This framework considers several desirable attributes:

(1) Universality: The embedding space should represent all possible sentence meanings.
(2) Diversity: The model should generate diverse outputs from a single embedding.
(3) Fluency: Generated text should be fluent and grammatically correct.
(4) Semantic Structure: Similar embeddings should correspond to similar meanings.

A challenge to the burgeoning vector-to-text-based research activity is presented in [8], where a defence mechanism against inversion attacks is proposed. We deem this a logical evolution and a precursor to the inherent cycle of development in the domain of inversion attacks, analogous to the field of cryptography research [14].

2 BACKGROUND

2.1 Vec2Text Method

The authors frame embedding inversion as a **controlled generation problem**, for which their approach leverages an iterative correction mechanism and controlled generation techniques to refine the reconstructed text. Given a text embedding $\mathbf{e} \in \mathbb{R}^d$ generated by an encoder ϕ, the goal is to recover the text sequence x so that the cosine similarity between the embedding of the reconstructed text $\phi(x)$ and the ground-truth embedding $\mathbf{e}$ is maximised:

$$x^* = \arg\max_x \cos(\phi(x), \mathbf{e})$$

This optimisation problem is computationally infeasible if approached directly due to the vast space of possible text sequences. Vec2Text circumvents this by learning a conditional language model to generate text hypotheses and iteratively refine them. The method starts by training a base model $p(x^{(0)}|\mathbf{e})$ to generate an initial hypothesis $x^{(0)}$ from the embedding $\mathbf{e}$. This is achieved by maximising the conditional likelihood:

$$\theta = \arg\max_{\hat{\theta}} \mathbb{E}_{x\sim\mathcal{D}}[p(x|\phi(x); \hat{\theta})]$$

where θ represents the parameters of the language model and $\mathcal{D}$ is the training dataset.

To improve the accuracy of reconstruction, Vec2Text employs an iterative correction mechanism. At each step t, a new text hypothesis $x^{(t+1)}$ is generated by conditioning on the previous hypothesis $x^{(t)}$, its new embedding $\phi(x^{(t)})$, and the difference between the new and ground-truth embeddings $\mathbf{e} - \phi(x^{(t)})$:

$$p(x^{(t+1)}|\mathbf{e}) = \sum_{x^{(t)}} p(x^{(t)}|\mathbf{e}) p(x^{(t+1)}|\mathbf{e}, x^{(t)}, \phi(x^{(t)}))$$

Whereby the refinement process is initialised with the base model $p(x^{(0)}|\mathbf{e}) = p(x^{(0)}|\mathbf{e}, \emptyset, \phi(\emptyset))$.

The model itself is an encoder-decoder transformer [20], designed to process input sequences of fixed dimensions. To accommodate embeddings of varying sizes, the authors employ a learnt projection mechanism that converts embeddings into a fixed-dimensional sequence. The input to the encoder is constructed by concatenating and projecting the following components:

(1) Ground-truth embedding $\mathbf{e}$
(2) Hypothesis embedding $\phi(x^{(t)})$
(3) Embedding difference $\mathbf{e} - \phi(x^{(t)})$
(4) Token embeddings of the current hypothesis $x^{(t)}$

The entire method is visualized in 1

During inference, Vec2Text uses a beam search to explore multiple candidate corrections efficiently. At each step t, it:

(1) Generates b top hypotheses $x_1^{(t+1)}, \ldots, x_b^{(t+1)}$
(2) Computes their embeddings $\phi(x_i^{(t+1)})$
(3) Selects the top b unique hypotheses based on their cosine similarity to $\mathbf{e}$

This ensures that the iterative process converges to a text hypothesis x^* with a high-quality reconstruction. The entire model, including the encoder-decoder transformer and projection layers, is trained end-to-end using a standard cross-entropy loss. An overview can be seen in Figure 1.

2.2 Backbones

The authors use 3 models in their work, including, on one hand, the base version of the **G**eneralizable **T**5-based dense **R**etrievers (GTR) as suggested in[11][16]. The model was fine-tuned to generate 32-token long sequences on the Natural Questions (NQ) dataset, one of the Benchmarking-IR datasets (BEIR)[19]. They also fine-tune two models based on the text–embedding–ada–002 OpenAI embedding model on the multifaceted corpus of MSMARCO[1], one for 32 and once on 128-token long sequences. In unison with the fine-tuned embedders, the original author trained the Vec2Text corrector model. This is also a T5-based architecture [16], with 12 layers of each encoder and decoders, trained on conditional generation, as discussed previously.

3 CLAIMS AND SCOPE

The key claims made in the paper by the authors are:

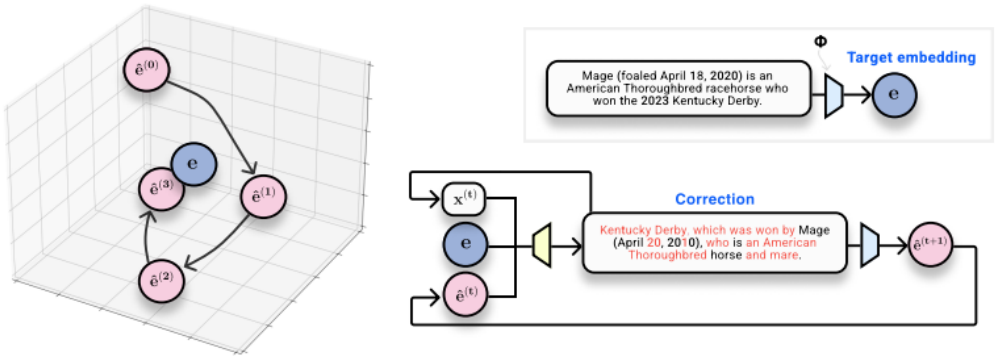

Figure 1: Overview of the Vec2Text iterative correction process. Given a target embedding e (Bleu), an initial text hypothesis is generated and repeatedly refined. Each iteration involves comparing the hypothesis's embedding $\hat{e}^{(t)}$ with the target e, and then adjusting the text to better align its embedding with the target. [10]

(1) Vec2Text outperforms current baselines in accuracy for text reconstruction from embeddings
(2) The method adapts well to out-of-domain data and can accurately reconstruct text across various domains.
(3) Reconstruction performance is inversely related to the length of the sequence. Longer sequences are harder to reconstruct.

In addition to the main claims, the paper performs an in-depth case study using a clinical dataset to evaluate the effectiveness of the model within a sensitive data domain, particularly with regard to individual names and other sensitive information in clinical records. This examination aims to highlight the potential privacy risks associated with text embeddings and underscore the imperative to protect them.

Furthermore, the paper investigates the effect of adding Gaussian noise to embeddings, which could potentially dampen reconstruction accuracy while preserving utility in retrieval tasks, offering a potential defence mechanism against inversion attacks.

The scope of this reproduction is to verify the core claims presented in the paper. We will not include the case study that involves the clinical dataset, as it is not the central focus of the paper and demonstrates results consistent with the primary claims. Likewise, we shall not replicate the Gaussian noise analysis at this time, since it has already been addressed by a separate reproduction of the original paper [23].

This reproduction focused solely on the authors' claims regarding their GTR-based model, as the text–embedding–ada–002 model fine-tuned on 32-token sequences was unavailable, and both the 32 and 128 sequence length models require paid access to the OpenAI API. This does impact the reproducibility of their out-of-domain claims, as detailed in 5.2.

We created a fork of the original codebase with added comments, fixed dependencies, and set up a reproduction folder with all necessary files to run the reproduction. A reproductionREADME[1] is provided, which includes explanations of how to run our reproduction and extension experiments.

[1]https://github.com/ryan-ott/vec2text

3.1 Extensions

Our extensions focus on investigating the computational requirements of Vec2Text, specifically examining how the number of correction steps and the beam search width affect the model's performance. Understanding the resource demands is crucial for evaluating the method's practicality and scalability, given its computationally intensive nature.

We also examine how the model performs on 'out-of-domain' data with lengths exceeding the token lengths it was trained on. Preliminary tests show that at least one of the underlying models is sensitive to the token length it was trained on. Evaluating the model's ability to handle longer texts provides insights into its adaptability. This is especially relevant in real-world scenarios, where the length of the original text or the specifics of the embedding-generating model may not be known.

4 REPRODUCTIONS

To replicate the results of the original paper, we conducted two sets of experiments designed to assess text reconstruction performance, corresponding to the evaluations presented in their Tables 1 and 2, respectively: one focused on in-domain evaluation and the other on out-of-domain evaluation. Both experiment sets aimed to assess the performance of the GTR model under their respective conditions.

4.1 Experimental setup

We used the same parameters as the author suggested and implemented. This means that for Table 1, we did a similar investigation of steps, while further exploring the effect of decoding beam-width (see 6.1). The hyperparameters or underlying model that were used for Table 2 were not reported. Preliminary experiments and communication with the author indicated that the Ada 128 model was used with a beam width of 8 and a maximum of 50 steps. We applied these parameters to the GTR-32 model for our Table 2. As with the Table1 experiments, we used a batch size of 16. To reduce computational overhead, we evaluated only 1,000 samples per dataset, using a fixed random seed (42) to ensure reproducibility in our

experiments. The original paper did not specify their random seed implementation or sample size.

Datasets. Mostly the same BEIR[19] datasets were used as in the original paper. However, 4 out of the datasets (Signal-1M, TREC-News, Robust04 and BioASQ) used in the original paper have since been made private. We only managed to get access to BioAsq, so the others ones are missing from the 2 results.

Metrics The primary metrics used are Bilingual Evaluation Understudy (BLEU) score and F1 token score. BLEU scores are commonly used in machine translation to evaluate the quality of generated text. They measure the overlap of n-grams (sequences of words) between the machine translation and the reference, in this case, the original input text and the reconstructed text. A higher BLEU score indicates closer alignment with the reference translations. Poignantly, BLEU scores are not perfect, as they do not account for semantic meaning or fluency, solely for word overlap. F1 token score is a metric that measures the similarity between the original text and the reconstructed text. It is calculated as the harmonic mean of precision and recall, where the True Positives (TP) are the number of tokens that are correctly reconstructed, and the False Positives (FP) and False Negatives (FN) are the number of tokens that are incorrectly reconstructed and not reconstructed, respectively. A higher F1 token score indicates a higher similarity between the original and reconstructed text. Other metrics are also used, such as cosine similarity, or exact match, but less emphasis is put on these by the original author.

Hardware The experiments were conducted on the Snellius cluster[17] with an NVIDIA A100-SXM4-40GB GPU[12].

5 RESULTS

5.1 In-Domain Results

The results for reproducing in-domain performance on the Natural Questions dataset using the pre-trained GTR model are shown in Table1. Our experimental results closely align with those reported by [10].

In the Table 1 progressive improvements are illustrated in various configurations. Starting from the Base model at 0 steps compared to the Vec2Text with a substantial boost in all metrics. With more training steps, performance continues to rise, reaching BLEU: 83% and tf1 of 95% at 20 steps, then BLEU of 84% and tf1 of 95% at 50 steps. Introducing beam search further enhances results, culminating in a BLEU score of 92% and a tf1 score of 98%. These findings confirm that the pre-trained GTR model's performance can be reliably reproduced on the Natural Questions dataset. Running the most compute-intensive configuration took just under 4 hours on our A100 GPU for 1000 samples, with peak GPU memory usage of around 8.1 GB. While the in-domain results are highly reproducible and almost exactly match those of [10], variation in the exact scores is observed. We are uncertain what led to this. Possible reasons could be that the sample size used was significantly smaller than the original, which allowed for more variance. The Exact score being a stringent metric likely means it suffers more from random variation on a smaller sample.

5.2 Out-of-Domain

Table 2 shows that performance varies notably depending on the dataset domain. For instance, scores for hotpotqa and NQ are relatively strong, achieving both a BLEU score of 89%. In contrast, datasets like bioasq, cqadupstack, and webis-touche2020 show lower BLEU and Token F1 scores, indicating greater difficulty in these domains. We note that the average runtime per dataset for 1000 samples was just over 4 hours as well, with also a similar peak memory usage of 8.4 GB.

These results unfortunately can neither confirm nor deny the original authors' Table 2 results. The only thing we can state is that up until the respective token length the model was trained on, it performs relatively well, but there is a big variety. We further wanted to investigate where this discrepancy in scores came from. The initial hypothesis was that the outcomes were likely related to the data the model was trained on. recreating domain-specific words with a model fine-tuned on a different domain is likely going to be difficult. if this is a general problem, it would hamper the usage of vector-to-texts models by bad actors, since they would need to specialize their model for specific domains.

Similarity investigation The above hypothesis was tested by taking 1000 random samples from each dataset in Table 2 and comparing them against the NQ dataset the GTR model was trained on. We decided to use as comparison metric the average Bertscore of all sentences [21] as this is argued to be a valid proxy for comparing datasets. We then take the cosine similarity between the embeddings.

As is visible in Figure 2, there is a strong positive correlation with the similarity of the dataset to the NQ dataset and the BLEU score. We would argue that this is evidence in favour of the strong in-domain capabilities of the Vec2Text model, but shows a lack of its generalisability. Similarly, it can be argued that the results of the original paper's Table 2 are less convincing as many of the tested datasets actually have very high similarity to the in-domain datasets. In general, it would be interesting to see how well this method generalizes. Does it need to be fine-tuned to even perform, or would a more varied training dataset also allow the model to perform better on all subdomains.

6 EXTENSION RESULTS

6.1 Beam Width Trade-Offs

As shown in Table 1 and in the original paper, increasing the beam-width of the decoder allows for enhanced reconstruction performance. We wanted to investigate what *the best trade-off point is for beam width, computational cost and accuracy* In order to investigate this we set up an experiment, varying the beam-width with a constant 50 correction steps on the GTR-32 model on the NQ validation set with 1000 samples.

Results. Figure 3 examines the balance between performance and computational efficiency in beam search. Increasing the beam width enhances embedding inversion by enabling broader exploration of the search space. However, the improvements diminish significantly beyond a width of 4. Wider beam widths also introduce substantial computational costs, with runtime escalating exponentially. This presents notable challenges for tasks constrained by

Table 1: In-domain reproducibility results using the GTR model finetuned by [10] on the Natural Questions dataset for sequences of 32 tokens. Values marked as NR indicate results not reported by the original paper. Reproduced results are italicised, and any results more than 5% off are bold.

Method	Pred Tokens		BLEU		TF1		Exact		Cos	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours	Original	Ours
Base [0 steps]	32	*31*	31.9	*31.6*	67	*67*	0.0	*1.4*	0.91	*0.90*
Vec2Text [1 step]	32	*31*	50.7	*49.5*	80	*80*	0.0	***8.4***	0.96	*0.95*
[20 steps]	32	*31*	83.9	*83.3*	96	*95*	40.2	***57.8***	0.99	*0.99*
[50 steps]	32	*31*	85.4	*84.2*	97	*96*	40.6	***58.6***	0.99	*0.99*
[50 steps + 2 sbeam]	NR	*31*	NR	*87.1*	NR	*96.5*	NR	*67.2*	NR	*99.2*
[50 steps + 4 sbeam]	NR	*31*	NR	*90.0*	NR	*97.1*	NR	*75.0*	NR	*99.5*
[50 steps + 8 sbeam]	32	*31*	97.3	***92.3***	99	*98*	92.0	***79.4***	0.99	*0.99*

Table 2: Evaluation on out-of-domain BEIR datasets using the GTR model trained to output 32-token sequences. Generation using 50 correction steps, 8 decoding beams on 1000 samples.

Dataset	Tokens	BLEU	tf1
hotpotqa	30.39	89.43	96.66
nq	31.02	88.79	96.56
dbpedia-entity	31.65	84.25	93.32
nfcorpus	31.90	81.77	92.00
climate-fever	27.73	80.74	91.78
fever	27.73	80.74	91.48
scifact	32.00	80.58	92.07
scidocs	31.03	78.24	91.63
arguana	32.00	78.10	92.90
msmarco	32.00	63.41	85.23
quora	15.64	57.57	78.95
trec-covid	23.52	56.16	66.30
fiqa	31.78	51.38	78.72
webis-touche2020	29.01	48.30	76.43
cquadupstack	31.69	46.00	78.17
bioasq	32.00	39.31	71.40

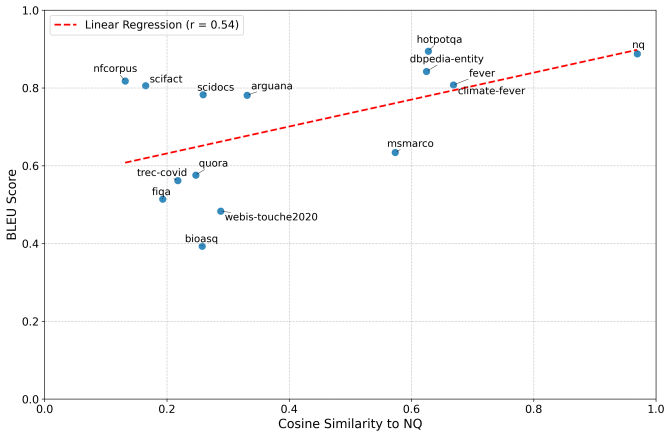

Figure 2: Cosine similarity of datasets to NQ versus the BLEU scores.

computational resources or time-sensitive applications. This underscores the necessity of tuning beam width to meet task-specific goals, ensuring a balance between accuracy and efficiency in practical applications. Moreover, we conducted the effect of beam width on peak memory usage, which showed expected scaling of GPU memory usage with doubling of beam-width as shown in Figure 5.

6.2 Impact of Sequence Length

Given the constraints of the 32-token pretrained GTR model, we aimed to evaluate the model's performance by truncating data to specific token lengths. Ideally, we would allow the model to still predict varying token lengths. It turns out the token length a model can predict is hardcoded to be the maximum of its trained length (and padded if lower), even though it seems variable. Thus, this experiment analyses *how well a model performs out of its sequence length domain*. It was chosen to perform this experiment on the 3 BEIR datasets: DBPedia, TREC-COVID, and FiQA-2018 who we respectively classified as *similar, average, non-similar* to the NQ dataset, based on the results from Table 2. The reasoning was that we would get a clear view of the relation between type of inference data and token length. In the end, it's relevant to know the practical application of this vector-to-text method. We evaluated on 500 samples with 50 steps and beam width of 4. These were chosen to be a good trade-off between computational cost and accuracy. We searched the space from 8 tokens to 52, with 4 token intervals. Preliminary results showed this was the most interesting range, as the graph flattens outside of it. Similarly, BLEU score starts bugging at lower token lengths because it looks for trigrams, which might not be appearing consistently.

Results Figure 4 illustrates a consistent downward trajectory in BLEU scores as token length increases, reflecting the challenges inherent in inverting high-quality long-sequence text. The decline suggests that as sequences grow longer, the model struggles to maintain both syntactic coherence and semantic accuracy, likely

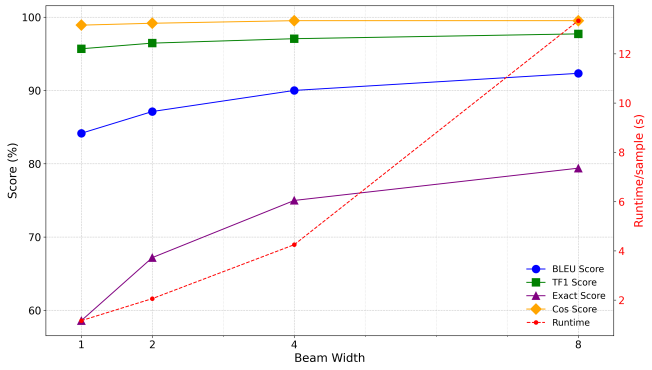

Figure 3: Impact of varying beam width against multiple metrics and runtime on 50 steps and 1,000 samples.

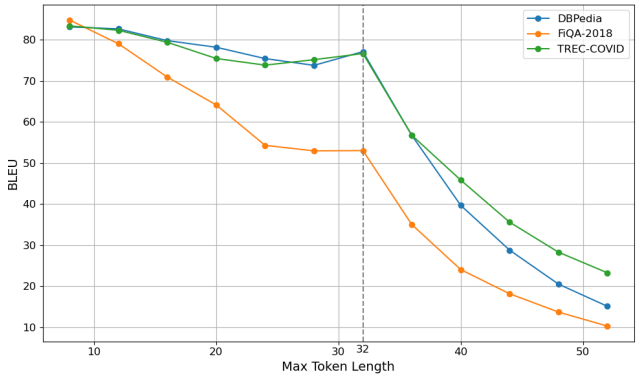

Figure 4: BLEU score progression over varying token lengths for 500 samples with DBPedia, TREC-COVID, and FiQA-2018.

due to limitations in the fixed token-length architecture. Interesting is the score for its specific trained token length; the BLEU score peaks at exactly 32, and becomes lower 4 tokens higher and 4 tokens lower, only to go back up for the shorter and shorter lengths.

Notably, the steepness of the decline accelerates beyond approximately 30 tokens, indicating that the model's capacity to manage longer contexts diminishes more drastically at this threshold. This trend highlights a critical bottleneck where the model transitions from manageable to significantly degraded performance. The relatively stable scores at shorter token lengths suggest that the model performs well when context size remains within its design constraints. All results point towards Vec2Text excelling at its predestined token length, disqualifying the claim 3 that the scores will be better for lower token models per definition.

7 CONCLUSION

In this paper, we conducted a reproduction study to [10]. In regards to the initial claims in Section 3:

(1) *Vec2Text outperforms current baselines* - The combination of both a baseline inverter model, combined with iterative refinement, seems to achieve higher scores compared to 1.1. We need to acknowledge that there are no direct comparisons made, since no other papers results have been done on the same datasets.
(2) *The method adapts well to out-of-domain data* - We can't conclusively state this at the current time. Our reproduction shows widely varying scores, but that can be due to the underlying GTR model not being expressive enough. Further investigation is required, possibly with different embedder models.
(3) *Reconstruction performance is inversely related to the length* - We would argue that at least for the GTR model, this fact does not stand. The model performs best at its trained token length, after which it dips and then slowly climbs, while at the higher side it crashes. This seems to be an overfitting on the training length.

While our reproduction successfully mirrored the original Table 1, Table 2 could not be established currently.

These findings open up several possibilities for future work. For example, it is crucial to acknowledge the limitations of the BLEU score as a primary metric for evaluating text reconstruction quality in the context of privacy. While BLEU effectively measures n-gram overlap, it may be overly strict when assessing the recovery of sensitive information. A reconstructed text that is semantically similar to the original but differs by even a single character can receive a significantly lower BLEU score. Similarly, a reconstruction that captures the core meaning and reveals private information, but deviates from the exact, wording may still receive a relatively low BLEU score. However, from a privacy perspective, even these "imperfect" reconstructions can pose a significant threat, as they may still reveal sensitive or personally identifiable information. This highlights a potential disconnect between the metric's emphasis on exact wording and the broader goal of assessing privacy risks associated with embedding inversion. Future work employing metrics based on derived entities from the text is likely to yield results that more accurately reflect the purported problem setting.

A simple but effective future development would be to train model with variable input length. Similarly, training a model without knowing the underlying embedder, for which preliminary work has been done in [5], likely is a promising avenue, as it gets closer to a generalizable method.

Finally, we want to comment that a more unified framework for this research area would benefit comparison. Currently, most research seems to use different datasets, methods, constraints and metrics, which makes it hard to clearly compare methods. It would be good for all actors to have a clear overview of what the good methods are, and especially which methods potentially protect versus which vector-to-text models. It is important to uncover both attacks but also protection to people's private data in the new ages of vector databases.

We want to acknowledge Yongkang Li for supervising our work, as well as the original main author Jack Morris for his answers to questions.

REFERENCES

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL] https://arxiv.org/abs/1611.09268

[2] Geoffrey Cideron, Sertan Girgin, Anton Raichuk, Olivier Pietquin, Olivier Bachem, and Léonard Hussenot. 2022. vec2text with Round-Trip Translations. arXiv:2209.06792 [cs.CL] https://arxiv.org/abs/2209.06792

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[4] Alexey Dosovitskiy and Thomas Brox. 2016. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4829–4837.

[5] Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. 2024. Transferable Embedding Inversion Attack: Uncovering Privacy Risks in Text Embeddings without Model Queries. (2024). arXiv:2406.10280 [cs.CR] https://arxiv.org/abs/2406.10280

[6] Kai Kugler, Simon Münker, Johannes Höhmann, and Achim Rettinger. 2024. InvBERT: Reconstructing Text from Contextualized Word Embeddings by inverting the BERT pipeline. (2024). https://doi.org/10.48694/JCLS.3572

[7] 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).

[8] Tiantian Liu, Hongwei Yao, Tong Wu, Zhan Qin, Feng Lin, Kui Ren, and Chun Chen. 2024. Mitigating Privacy Risks in LLM Embeddings from Embedding Inversion. (2024). arXiv:2411.05034 [cs.CR] https://arxiv.org/abs/2411.05034

[9] Aravindh Mahendran and Andrea Vedaldi. 2014. Understanding Deep Image Representations by Inverting Them. (2014). arXiv:1412.0035 [cs.CV] https://arxiv.org/abs/1412.0035

[10] 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).

[11] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899* (2021).

[12] NVIDIA Corporation. 2020. NVIDIA A100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/a100/ Data Center GPU.

[13] J.J. Pan, J. Wang, and G. Li. 2024. Survey of vector database management systems. *The VLDB Journal* 33 (2024), 1591–1615. https://doi.org/10.1007/s00778-024-00864-x

[14] Lauren Pearce. 2020. *A Whirlwind History of Cryptography*. Technical Report. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States). https://doi.org/10.2172/1671059

[15] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[17] Snellius Computing Infrastructure. 2024. Snellius High-Performance Computing Cluster. https://www.snellius.example.com Computational resource utilized for experiments..

[18] Congzheng Song and Ananth Raghunathan. 2020. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 377–390. https://doi.org/10.1145/3372297.3417270

[19] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv:2104.08663 [cs.IR] https://arxiv.org/abs/2104.08663

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

[21] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs.CL] https://arxiv.org/abs/1904.09675

[22] Zhihao Zhu, Ninglu Shao, Defu Lian, Chenwang Wu, Zheng Liu, Yi Yang, and Enhong Chen. 2024. Understanding Privacy Risks of Embeddings Induced by Large Language Models. *arXiv preprint* (2024). arXiv:2404.16587 https://arxiv.org/abs/2404.16587v1

[23] Shengyao Zhuang, Bevan Koopman, Xiaoran Chu, and Guido Zuccon. 2024. Understanding and Mitigating the Threat of Vec2Text to Dense Retrieval Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Washington, DC, USA, 2391–2401. https://doi.org/writeonitfromacopyrightprotectionperspective, note={AnalyzesandmitigatesVec2Textattacksonretrievalsystems. RelevanttothesecurityofinformationretrievalwhentexttoembeddingsisUsed.}, acmid={4634352},isbn={978-1-4503-9838-1}

A APPENDIX

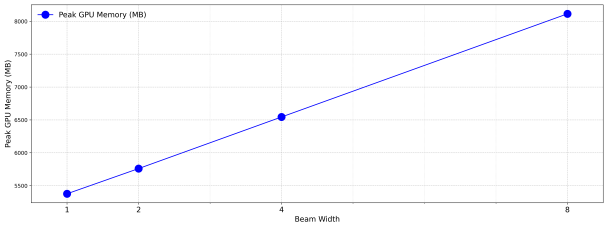

Figure 5: Effect of beam width on peak memory usage using the GTR-32 backbone