Textomics: A Dataset for Genomics Data Summary Generation

Mu-Chun Wang[1,*], **Zixuan Liu**[2,*], **Sheng Wang**[2]
[1]University of Science and Technology of China
[2]Paul G. Allen Scholl of Computer Science and Engineering, University of Washington
amoswang2000@mail.ustc.edu.cn
{zucksliu, swang}@cs.washington.edu

Abstract

Summarizing biomedical discovery from genomics data using natural languages is an essential step in biomedical research but is mostly done manually. Here, we introduce Textomics, a novel dataset of genomics data description, which contains 22,273 pairs of genomics data matrices and their summaries. Each summary is written by the researchers who generated the data and associated with a scientific paper. Based on this dataset, we study two novel tasks: generating textual summary from a genomics data matrix and vice versa. Inspired by the successful applications of k nearest neighbors in modeling genomics data, we propose a kNN-Vec2Text model to address these tasks and observe substantial improvement on our dataset. We further illustrate how Textomics can be used to advance other applications, including evaluating scientific paper embeddings and generating masked templates for scientific paper understanding. Textomics serves as the first benchmark for generating textual summaries for genomics data and we envision it will be broadly applied to other biomedical and natural language processing applications.[1]

1 Introduction

Modern genomics research has become increasingly automated through being roughly divided into three sequential steps: next-generation sequencing technology produces a massive amount of genomics data, which are in turn processed by bioinformatics tools to identify key variants and genes, and, ultimately, analyzed by biologists to summarize the discovery (Goodwin et al., 2016; Kanehisa and Bork, 2003). In contrast to the first two steps that have been automated by new technologies and software, the last step of summarizing discovery is still largely performed manually, substantially slowing down the progress of scientific discovery (Hwang et al., 2018). A plausible solution is to automatically summarize the discovery from genomics data using neural text generation, which has been successfully applied to radiology report generation (Wang et al., 2021; Yuan et al., 2019) and clinical notes generation (Melamud and Shivade, 2019; Lee, 2018; Miura et al., 2021).

In this paper, we study this novel task of generating sentences to summarize a genomics data matrix. Several excisting approaches demonstrate encouraging results in generating short phrases to describe functions of a set of genes (Wang et al., 2018; Zhang et al., 2020; Kramer et al., 2014). However, our task is fundamentally different from these: the input of our task is a matrix that contains tens of thousands of genes, which could be noisier than a set of selected genes; the outputs of our task are sentences instead of short phrases or controlled vocabularies.

To study this task, we curate a novel dataset, Textomics, by integrating data from PMC, PubMed, and Gene Expression Omnibus (GEO) (Edgar et al., 2002) (**Figure** 1). GEO is the default database repository for researchers to upload their genomics data matrices, such as gene expression matrices and mutation matrices. Each genomics data matrix in GEO is a sample by feature matrices, where samples are from often humans or mice that are sequenced together to study a specific biological problem, and features are genes or variants. Each matrix is also associated with a few sentences that are written by researchers to summarize this data matrix. After pre-processing, we obtain 22,273 matrix summary pairs, spanning 9 sequencing technology platforms. Each matrix has on average 2,475 samples and 22,796 features. Each summary has on average 46 words.

[*] Equal Contribution

[1] The link to access our code: https://github.com/amos814/Textomics

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4878 - 4891
May 22-27, 2022

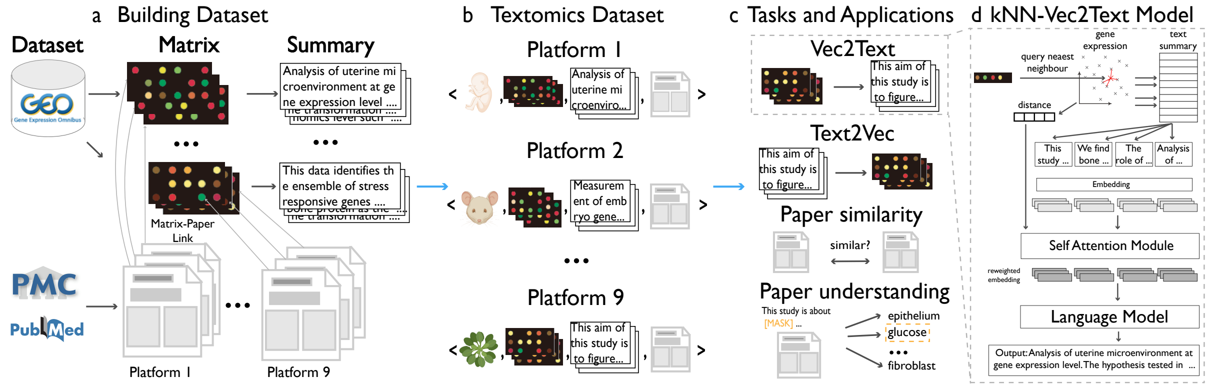

Figure 1: **Flow chart of Textomics.** a. Genomics data matrices and summaries are collected from GEO. Scientific papers are collected from PMC and PubMed. Each data matrix is associated with a unique summary and a unique scientific paper in Textomics. b. Textomics is divided into 9 sequencing platforms, spanning over various species. Data matrices in the same platforms share the same features and can therefore be used to train a machine learning model. c. Textomics can be used as the benchmark for a variety of tasks, including Vec2Text, Text2Vec, measuring paper similarity, and scientific paper understanding. d. kNN-Vec2Text is developed to address the task of Vec2Text, by first constructing a reference summary using similar genomics data matrices and then unifying these summaries to generate a new summary.

We further propose a novel approach to automatically generate a summary from a genomics data matrix, which is highly noisy and high-dimensional. k nearest neighbor (kNN) approaches have obtained great success in genomics data by capturing the hidden modules within it (Levine et al., 2015; Baran et al., 2019). The key idea of our method is to find k nearest summaries according to the genomics data similarity and then exploit the attention mechanism to convert these k nearest summaries to a new summary. Our method obtained substantial improvement in comparison to baseline approaches. We further illustrated how we can generate a genomics data matrix from a given summary, offering the possibility to simulate genomics data from textual description. We then introduced how Textomics can be used as a novel benchmark for measuring scientific paper similarity and evaluating scientific paper understanding. To the best of our knowledge, Textomics and kNN-Vec2Text together build up the first large-scale benchmark for genomics data summary generation, and can be broadly applied to a variety of natural language processing tasks.

Our paper is written as follows: We first introduce the Textomics dataset (section 2) and describe the Text2Vec and Vec2Text tasks (section 3). We then propose a baseline model and kNN-Vec2Text model for Vec2Text task (section 4.1) and the model for Text2Vec task. We then evaluate our method (section 5) and provide two applications (section 6) based on Textomics dataset. We then discussed the related works and the potential direction of future works (section 7 and 8).

2 Textomics Dataset

We collected genomics data matrices from Gene Expression Omnibus (GEO) (Edgar et al., 2002). The feature of each data matrix represents the expression level of a gene or other genomic measurements of a variant (typically real numbers). The sample of each matrix is an experimental subject, such as an experimental animal or a patient. Each data matrix is associated with an expert-written summary, describing this data matrix. We obtained in total 164,667 matrix-summary pairs, spanning 12,219 sequencing platforms.

Samples in different platforms have different features. However, data matrices belonging to the same sequencing platform are from the same species and share the same set of features, thus can be used together for model training. To further alleviate the missing feature problem, we kept the top-20000 features with a lower missing rate and filtered out the rest. We further selected 9 platforms with the average lowest rate of missing value and the largest amount of matrix-summary pairs to guarantee the quality and the scale of the dataset. After all, we imputed the resulted data matrices using averaging imputation across different features.

Data matrices belonging to the same platform have distinct samples (e.g., patient samples collected from two hospitals). To make them comparable and provide fixed-size features for machine learning models, we empirically used a five-number summary to represent each data matrix. In particular, we calculated the smallest, the first quartile, the median, the third quartile, and the largest value of each feature across samples in a specific data matrix. We then concatenated these values of all features, resulting in a 100k-dimensional feature vector for each data matrix. Compared with other statistics such as mean, median, and mode of the features, the five number statistics maintain the patterns hidden in the raw matrices better. This vector will be finally used as the input to the machine learning model.

All genomics data summaries we collected were written by the biologists who generate the corresponding genomics data matrices. Therefore, these summaries can properly reflect biologists' descriptions of their datasets. Since the summary is the first piece of information that one can learn about the dataset, authors often tend to clearly characterize their dataset in the summary. However, directly leveraging raw data of these summaries is questionable. On the syntactic level, the lengths of summary for each sample are different and comments are often used in genomics descriptions. In order to align our data and leverage the advanced Transformer model that requires fix-length sentences as well as simplifies the structure of the summary, we empirically removed the text in the brackets and truncated the summaries length to 64 words (the percentage of summaries with a length greater than 64 is 41%). On the semantic level, there could be non-informative summaries such as a simple sentence *'Please see our data below'* and some outliers that are substantially different from other summaries. In order to increase the quality of these genomics data summaries, we manually inspected and removed the non-informative summary and excluded the outliers based on the pairwise BLEU (Papineni et al., 2002) scores through a progressive automated procedure. Specifically, for every summary, we treated it as the query text and calculated the pairwise BLEU-1 scores with all other summaries, filtered out those median that is lower than 0.09, and then re-applied the procedure with a higher threshold of 0.13. Finally, each of the 9 platforms contains 471 matrix-summary pairs on average, presenting a desirable number of training samples to develop data summary generation models. We summarized the statistics of these 9 platforms in **Supplementary Table** S1.

Some of the data matrices are associated with a scientific paper, which describes how the authors generated and used the data. Therefore, the data matrix and the summary can be used to help embed these papers. We additionally retrieved these papers from PubMed and PMC databases according to the paper titles enclosed in GEO. We obtained the full text for those 7,691 freely accessible ones (**Supplementary Table** S1). We will introduce two applications that jointly use scientific papers and matrix-summary pairs in section 6.

3 Task Description

We aim to accelerate genomics discovery by generating a textual summary given the five-number summary-based vector of a genomics data matrix. We refer to the five-number summary-based vector as a gene feature vector for simplicity.

Specifically, consider textual summary domain $\mathcal{D}$ and gene feature vector domain $\mathcal{V}$, let $\mathbf{D} = \{\mathbf{D}_{\mathcal{D}}, \mathbf{D}_{\mathcal{V}}\} = \{(\mathbf{d}_i, \mathbf{v}_i)\}_{i=1}^{N} \overset{dist}{\sim} \mathbb{P}(\mathcal{D}, \mathcal{V})$ be a dataset containing N summary-vector pairs sampled from the joint distribution of these two domains, where $\mathbf{d}_i \triangleq \langle d_i^1, d_i^2, ..., d_i^{n_{d_i}} \rangle$ denotes a token sequence and $\mathbf{v}_i \in R^{l_v}$ denotes the gene feature vector. Here $d_i^j \in C$, C is the vocabulary. We now formally define two cross-domain generation tasks, Vec2Text and Text2Vec, based on our

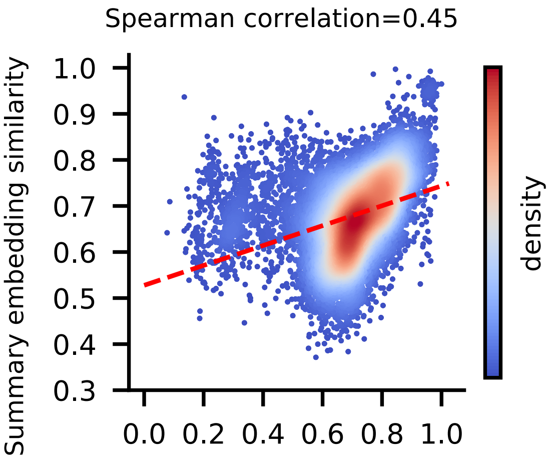

Figure 2: Density plot showing the Spearman correlation between text-based similarity (y-axis) and vector-based similarity (x-axis) on sequencing platform GPL6246. Each dot is a pair of data samples. A larger Spearman correlation indicates this Enc$_d$ is more accurate in embedding scientific papers.

dataset. Given a gene feature vector $\mathbf{v}_i$, Vec2Text aims to generate a summary $\mathbf{d}_i$ that could best describe this vector $\mathbf{v}_i$; given a textual summary $\mathbf{d}_i$, Text2Vec aims to generate the gene feature vector $\mathbf{v}_i$ that $\mathbf{d}_i$ describes. Since we are studying a novel task on a novel dataset, we first examined the feasibility of this task. To this end, we obtained the dense representation of each textual summary using the pre-trained SPECTER model (Cohan et al., 2020) and use these representations to calculate a summary-based similarity between each pair of summaries. We also calculated a vector-based similarity based on the gene feature vector using the cosine similarity. We found that these two similarity measurements show a substantial agreement (**Figure** 2, **Supplementary Table** S2). After filtering out the outliers, all 9 platforms achieved a Spearman correlation greater than 0.2, suggesting the possibility to generate textual summary from the gene feature vector and vice versa.

4 Methods

4.1 Vec2Text

We first introduce a baseline model that tries to encode gene feature vectors into the semantic embedding space and then decodes it to generate text. The baseline model contains a word embedding function Emb(.), a gene feature vector encoder $\text{Enc}_v(.)$ and a decoder $\text{Dec}_v(.)$. Given a gene feature vector $\mathbf{v}_i$, the encoder will first embed the data into a semantic representation space $\mathbf{s}_i^{(0)} = \text{Enc}_v(\mathbf{v}_i)$, and then the decoder will start from this representation for the text generation. The generation process is autoregressive. It generates j-th word $\hat{d}_i^{(j)}$ and its embedding $\mathbf{s}_i^{(j)}$ as:

$$P(\hat{d}_i^{(j)}|\mathbf{s}_i^{(<j)}) = \text{Dec}_v(\mathbf{s}_i^{(<j)}), j = 1, ..., n_{d_i}. \quad (1)$$

Then we sample the next word and obtain its embedding as:

$$\mathbf{s}_i^{(j)} = \text{Emb}(\hat{d}_i^{(j)}),\ \hat{d}_i^{(j)} \overset{sample}{\sim} P(\hat{d}_i^{(j)}|\mathbf{s}_i^{(<j)}). \quad (2)$$

This model is trained using the following loss function:

$$\mathcal{L}_{\text{baseline}} = -\frac{1}{|\mathbf{D}_\mathcal{V}|}\sum_{i=1}^{|\mathbf{D}_\mathcal{V}|}\sum_{j=1}^{n_{d_i}} \log P(\hat{d}_i^{(j)}|\mathbf{s}_i^{(<j)}). \quad (3)$$

4.1.1 kNN-Vec2Text Model

The baseline model attempts to learn an encoder that projects a gene feature vector to a semantic representation. However, the substantial noise and the high-dimensionality of the gene feature vector pose great challenges to effectively learn that projection. k-nearest neighbors models have been extensively used as the solution to overcome such issues in genomics data analysis (Levine et al., 2015; Baran et al., 2019). Therefore, one plausible solution is to explicitly leverage summaries from similar gene feature vectors to improve the generation. Inspired by the encouraging performance in using k-nearest neighbors (kNN) in seq2seq models (Khandelwal et al., 2020, 2021) and genomics data analysis (Levine et al., 2015; Baran et al., 2019), we propose to convert the Vec2Text problem to a Text2Text problem according to the k-nearest neighbor of each vector.

For a given gene feature vector $\mathbf{g}$, we use $e_i \in R$ to denote its Euclidean distance to another gene feature vectors $\mathbf{v}_i$ in $\mathbf{D}$. We then select the summaries of k samples that have the minimum Euclidean distances as the reference summary list $\tilde{\mathbf{t}} = [\mathbf{d}_{j_1}, ..., \mathbf{d}_{j_k}]$, where $j_m \in \{1, 2, ..., |\mathbf{D}|\}$ denotes the index of ordered summaries w.r.t the Euclidean distance, i.e, $e_{j_1} \leq e_{j_2} \leq ... \leq e_{j_{|\mathbf{D}|}}$.

In addition to alleviating the noise in genomics data using the reference summary list (Levine et al., 2015; Baran et al., 2019), our method explicitly converts the Vec2Text problem to a Text2Text problem, and can thus seamlessly incorporate many advanced pre-trained language models into our framework. The resulted problem we need to solve is a k sources to one target generation problem. One naive solution is to concatenate the k reference summaries together. However, this concatenation will make the source text much longer than the target text and how to order each summary during concatenation also remains unclear. Instead, we propose to transform this problem into k one-to-one generation problem and then use attention-based strategy to fuse them. Concretely, let $\mathbf{n_j} = \mathbf{max}\{n_{j_1}, ..., n_{j_k}\}$ be the maximum length among all the reference summaries. We first get the representation of each summary $\mathbf{x}_{j_m} = \text{Emb}(\mathbf{d}_{j_m}) = \langle \mathbf{x}_{j_m}^{(1)}, ..., \mathbf{x}_{j_m}^{(\mathbf{n}_j)} \rangle$ for $m = 1, ..., k$. Here $\mathbf{x}_{j_m}^{(i)}$ denotes the vector embedding of the i-th word in m-th summary. We construct fixed-length reference summaries by padding after the end of each summary with length less than $\mathbf{n_j}$. We then utilize self-attention module (SA) (Vaswani et al., 2017) to get the aggregated embedding of each reference with their embeddings as well as the gene feature vector distance e_i. Let Q_r, K_r, V_r be the query, key, value matrices of embedding sequence $\mathbf{r} = \langle \mathbf{r}^{(1)}, ..., \mathbf{r}^{(l_r)} \rangle$, we have:

$$\text{SA}(\mathbf{r}) = \text{Attention}(Q_r, K_r, V_r). \tag{4}$$

We then calculate the attention score as following:

$$\mathbf{a}_{j_m} = \text{SA}(\langle \mathbf{x}_{j_m}^{(1)}, ..., \mathbf{x}_{j_m}^{(n_{j_k})} \rangle), \tag{5}$$

$$\text{sc}_j = \text{SA}(\langle e_{j_1} \cdot \mathbf{a}_{j_1}, ..., e_{j_k} \cdot \mathbf{a}_{j_k} \rangle), \tag{6}$$

where $\text{sc}_j = [\text{sc}_{j_1}, ..., \text{sc}_{j_k}] \in R^k$. Here we used a 2-layer self attention scheme to first acquire the aggregated feature of each summary and then calculate the attention score based on that. The final score is then calculated based on the attention scores and temperature τ as:

$$w_{j_m} = \frac{\exp(\tau \cdot \text{sc}_{j_m})}{\sum_{l=1}^{k} \exp(\tau \cdot \text{sc}_{j_l})}. \tag{7}$$

Then, we aggregate embedding sequences by taking weighted averages:

$$\tilde{\mathbf{x}}_j^{(l)} = \sum_{m=1}^{k} w_{j_m} \mathbf{x}_{j_m}^{(l)}, l = 1, ..., \mathbf{n_j}. \tag{8}$$

Let $P_{<l,x}(\mathbf{d}) = P_{\theta_{LM}}(d^{(l)}|d^{(<l)}, \mathbf{x}), 0 < l < n_d$ be the probability distribution of $d^{(l)}$ output by the language model θ_{LM} conditioned on the sequences of the embedding vectors $\mathbf{x}$ and the first $l - 1$ sequence tokens. We feed the aggregated embedding sequences into the language model to reconstruct the summary $\mathbf{d}$ using an autoregressive-based loss function:

$$\mathcal{L}_{k\text{NN-Vec2Text}} = -\sum_{\mathbf{d} \in \mathbf{D}_\mathcal{D}} \sum_{l=1}^{n_d} \frac{\log P_{<l,\tilde{x}_j}(\mathbf{d})}{|\mathbf{D}_\mathcal{D}|}. \tag{9}$$

4.2 Text2Vec

We model the reverse problem of generating the gene feature vector $\mathbf{v}$ from a textual summary $\mathbf{d}$ as a regression problem. Our model is composed with a semantic encoder $\text{Enc}_d(.)$ and a readout head MLP(.). Specifically, the encoder will embed the textual summary into dense representation $\mathbf{x} = \text{Enc}_d(\mathbf{d})$, and the readout head will map the representation to the gene feature vector $\hat{\mathbf{v}} = \text{MLP}(\mathbf{x})$. Then we train this model by minimizing the rooted mean squared errors (RMSE):

$$\mathcal{L}_\text{v} = \sqrt{\frac{1}{|\mathbf{D}_\mathcal{V}|} \sum_{\mathbf{v}_i \in \mathbf{D}_\mathcal{V}} ||\hat{\mathbf{v}}_i - \mathbf{v}_i||_2^2}. \tag{10}$$

5 Results

5.1 Vec2Text

To evaluate the performance of kNN-Vec2Text on the task of Vec2Text, we compared it to the baseline models in 4.1. For the baseline models, we used a one layer MLP network as its encoder, and tested with different decoder structure, including canonical Transformer (decoder of T5) (Vaswani et al., 2017), GPT-2 (Radford et al., 2019), and Sent-VAE (Bowman et al., 2016). For kNN-Vec2Text, we directly used both the encoder and the decoder of T5 (Raffel et al., 2020), one of the state-of-the-art Transformer style models. we set $k = 4$ and $\tau = 0.1$ as this setting achieved the best empirical performance, though it is worth noting that our model is robust on the choices of k (from 1 to 4) and τ (from 0 to 1). For all 9 platforms, we reported the average performance under 5-fold cross validation to evaluate the robustness of our method. The results of BLEU-1 score (Papineni et al., 2002) are summarized in **Figure 3a**. We found that kNN-Vec2Text substantially outperformed other methods by a large margin. Specifically, kNN-Vec2Text obtained a 0.206 BLEU-1 score on average while none of the other three methods achieved an average BLEU-1 score greater than 0.150. The prominent performance of our method demonstrates the effectiveness of using a k-nearest-neighbor approach to convert the Vec2Text problem to a Text2Text problem.

To further understand the superior performance of the kNN-Vec2Text model, we presented a case study in **Table 1**. In this case study, the generated summary is highly accurate compared to the ground truth summary. By examining the summaries of the 4 nearest neighbors in the gene feature vector space, we found that the generated summary is composed of short spans from each individual neighbor, again indicating the advantage of using a k-nearest neighbor for this task. Our method leveraged an attention mechanism to unify these four neighbors, thus offering an accurate generation. We also observed consistent improvement of our method over comparison approaches on other metrics and summarized the results in **Supplementary Table** S3.

5.2 Text2Vec

We next used the Text2Vec task to illustrate how our dataset can be used to compare the performance of different pre-trained language models. In particular, we compared a recently proposed scientific paper embedding method SPECTER (Cohan et al., 2020), which has demonstrated prominent performance in a variety of scientific paper analysis tasks, with SciBERT (Beltagy et al., 2019), BioBERT (Lee

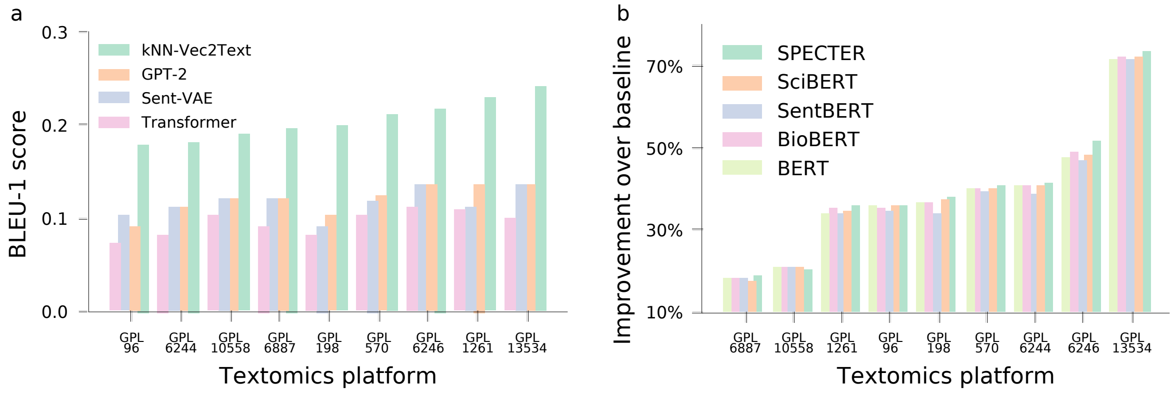

Figure 3: Performance on Vec2Text (a) and Text2Vec (b) using Textomics as the benchmark. a. Bar plot comparing our method kNN-Vec2Text with existing approaches on the ask of Vec2Text across 9 platforms in Textomics. b. Bar plot comparing the performance of different scientific paper embedding methods across 9 platforms in Textomics.

Table 1: A case study of the generated text by kNN-Vec2Text. Summaries of the four nearest neighbors in the input space are shown. The generated text is composed of short spans from the four different neighbors (colored in red). The BLEU-1 score for this example is 1 (prefect).

Neighbor 1:	Analysis of B16 tumor microenvironment at gene expression level. The hypothesis tested in the present study was that Tregs orchestrated the immune response triggered in presence of tumors.
Neighbor 2:	This study aims to look at gene expression profiles between wildtype and Bapx1 knockout cells of the gut in a E12.5 mouse embryo.
Neighbor 3:	The role of bone morphogenetic protein 2 in regulating transformation of the uterine stroma during embryo implantation in mice was investigated by the conditional ablation of Bmp2 in the uterus using the mouse.
Neighbor 4:	Measurement of specific gene expression in clinical samples is a promising approach for monitoring the recipient immune status to the graft in organ transplantation.
Generated:	Analysis of uterine microenvironment at gene expression level. The hypothesis tested in the present study was that Tregs orchestrated the immune response triggered in presence of embryo.
Truth:	Analysis of uterine microenvironment at gene expression level. The hypothesis tested in the present study was that Tregs orchestrated the immune response triggered in presence of embryo.

et al., 2020) and SentBERT (Wang and Kuo, 2020) and the vanilla BERT (Devlin et al., 2019). While the other language models directly take the token sequence as the input, SPECTER model needs to take both the abstract and the title. To make a fair comparison, we concatenated the title and the summary as the input for models other than SPECTER. For all 9 platforms, we reported the average performance under 5-fold cross validation. We further implemented a simple averaging baseline approach that predicts the vector for a test summary according to the average vectors of training samples. This baseline does not utilize any textual summary and can thus help us assess the effect of using textual summary information in this task. We used RMSE to evaluate the performance of all methods. We reported the RMSE improvement of each method over the averaging baseline model in **Figure 3b**. We found that all methods outperform the baseline approaches by gaining at least 15% improvement, indicating the importance of considering textual summary in this task. SPECTER achieved the best overall performance among all five methods, suggesting the advantage of separately modeling the title and the abstract when embedding scientific papers.

6 Applications

6.1 Evaluate paper embedding via Textomics

Embedding scientific papers is crucial to effectively identify emerging research topics and new knowledge from scientific literature. To this end, many machine learning models have been proposed to embed scientific papers into dense embeddings and then applied these embeddings for a variety of downstream applications (Cohan et al., 2020; Lee et al., 2020; Wang and Kuo, 2020; Beltagy et al., 2019; Devlin et al., 2019). However, there is currently limited golden standard that can measure the similarity between two papers. As a result, existing approaches use surrogate metrics such as citation relationship, keywords, and user activities to evaluate their paper embeddings (Cohan et al.,

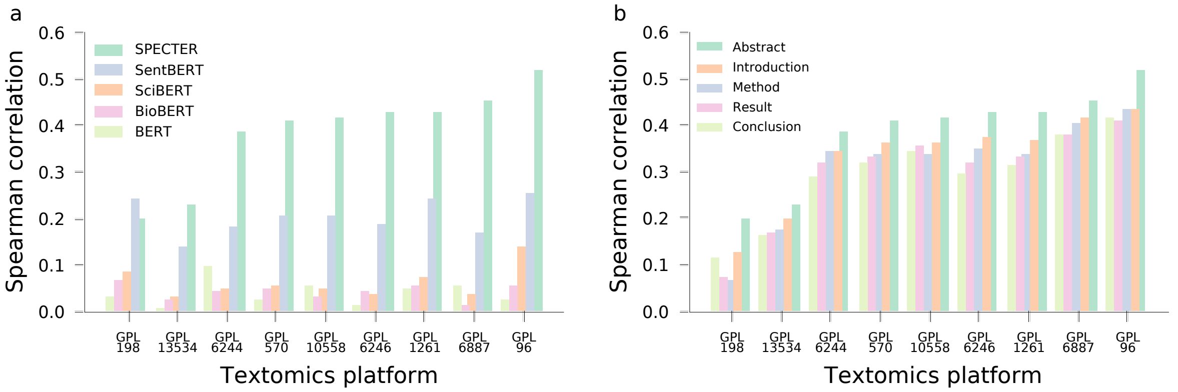

Figure 4: Performance on using Textomics as the benchmark to evaluate scientific paper embeddings. (A). Bar plot showing the comparison on embedding scientific papers using Textomics as the benchmark. (B). Bar plot showing the comparison on SPECTER embedding of different paper sections using Textomics as the benchmark.

2020; Chen et al., 2019; Wang et al., 2019).

Textomics can be used to measure these paper embedding approaches by examining the consistency between the embedding-based paper similarity and the embedding-based summary similarity since both the paper and the summary are written by the same authors. In particular, for a pair of summaries $\mathbf{d}_i, \mathbf{d}_j \in \mathbf{D}_{\mathcal{D}}$, let $\mathbf{t}_i, \mathbf{t}_j$ be the text (e.g., abstracts) extracted from their corresponding scientific papers. Let Enc_d be the encoder of the paper embedding method we want to evaluate. We first get their embeddings as:

$$\mathbf{s}_{d_i}, \mathbf{s}_{d_j} = \text{Enc}_d(\mathbf{d}_i), \text{Enc}_d(\mathbf{d}_j) \quad \in R^{l_s}, \qquad (11)$$

$$\mathbf{s}_{t_i}, \mathbf{s}_{t_j} = \text{Enc}_d(\mathbf{t}_i), \text{Enc}_d(\mathbf{t}_j) \quad \in R^{l_s}. \qquad (12)$$

We then compute the pairwise Euclidean distance between all pairs of summaries and all pairs of paper text as:

$$\mathbf{s}_{d_{i,j}} = \sqrt{\sum_{k=1}^{l_s} (s_{d_i}^{(k)} - s_{d_j}^{(k)})^2} \quad \in R, \qquad (13)$$

$$\mathbf{s}_{t_{i,j}} = \sqrt{\sum_{k=1}^{l_s} (s_{t_i}^{(k)} - s_{t_j}^{(k)})^2} \quad \in R. \qquad (14)$$

To evaluate the quality of the encoder Enc_d, we can calculate the Spearman correlation between the pairwise summary similarity and the pairwise text similarity. A larger Spearman correlation means the summary / textual contents of two samples in the pair are better aligned with each other, which indicates this Enc_d is more accurate in embedding scientific papers. As a proof-of-concept, we obtained the full text of 7,691 papers in our dataset from the freely accessible PubMed Central. We segmented each paper into five sections, which included abstract, introduction, method, result and conclusion. We first compared different paper embedding methods using the abstract of a paper. The five embedding methods we considered are introduced in section 5.1. Since SPECTER takes both the title and paragraph as the input we used the first sentence of the summary as a pseudo-title when encoding the summary. The results are summarized in **Figure 4a**. We found that SPECTER was substantially better than other methods on 8 out of the 9 platforms. SPECTER is specifically developed to embed scientific papers by processing the title and the abstract separately, whereas other pre-trained language models simply concatenated the title and the abstract. The superior performance of SPECTER suggests the importance of separately modeling paper title and abstract when embedding scientific papers. SentBERT obtained the best performance among four pre-trained language models, partially due to its prominent performance in sentence-level embedding. We further noticed that the relative performance among different methods is largely consistent with the previous work evaluated on other metrics (Cohan et al., 2020), demonstrating the high-quality of Textomics.

After observing the superior performance of SPECTER, we next investigated which section of the paper can be best used to assess paper similarity. Although existing paper embedding approaches often leverage the abstract for embedding, other sections, such as introduction and results might also be informative, especially for papers describing a specific dataset or method. We thus applied SPECTER to embed five different sections of each scientific paper and used Textomics to evaluate which section can best reflect paper similarity. We observed a consistent improvement of using the abstract section in comparison to other paper sections (**Figure** 4B), which is consistent with the intuition that the abstract represents a good summary of the scientific paper, again indicating the reliability of using Textomics to evaluate paper embedding methods.

6.2 Scientific paper understanding

Creating masked sentences and then filling in these masks can examine whether the machine learning model has properly understood a scientific paper (Yang et al., 2019; Guu et al., 2020; Ghazvininejad et al., 2019; Bao et al., 2020; Salazar et al., 2020). However, one challenge in such research is how to generate masked sentences that are relevant to a given paper while also ensuring the answer is enclosed in the paper. Our dataset could be used to automatically generate such masked sentences using the summary, which is highly relevant to the paper but also not overlapped with the paper. In particular, we can mask out keywords from the summary and then use this masked summary as the question and let a machine learning model to find the answer from the non-overlapping scientific paper. Let C_{bio} be a dictionary that contains biological keywords we want to mask out from the summary, $(\mathbf{d}_i, \mathbf{t}_i)$ be a pair of textual summary and paragraph text extracted from its corresponding scientific paper. If the j-th word $w_i = d_i^{(j)} \in C_{\text{bio}}$ in the summary belongs to C_{bio}, our proposed task is to predict which word in C_{bio} is the missing word in $\mathbf{d}_{\text{masked}}$ given $\mathbf{t}_i$. The masked summary $\mathbf{d}_{\text{masked}}$ is the same as $\mathbf{d}_i$ except its j-th word is substituted with [PAD]. For simplicity, we only mask at most one token in $\mathbf{d}_i$. We, therefore, form our task as a multi-class classification problem. Similar to section 6.1, we used the paper abstract as the paragraph text $\mathbf{t}_i$. To generate C_{bio}, we leveraged a recently developed biological terminology dataset Graphine (Liu et al., 2021), which provides the biological phrases spanning 227 categories. We selected 10 categories that can produce the largest number of masked sentences in Textomics. We manually filtered ambiguous words and stop words. On average, each category contains 317 keywords. We used a fully connected neural network to perform the multi-class classification task. The input feature is the concatenation of the masked summary embedding and the paragraph embedding. We used SPECTER to derive these embeddings as it has obtained the best performance in our previous analysis. The results are summarized in **Figure 5**. We observed improved accuracy on all ten categories, which are much better than the 0.4% accuracy by random guessing, indicating the usefulness of our benchmark in scientific paper understanding. Finally, we found that the performance of each category varied across different platforms, suggesting the possibility to further improve the performance by jointly learning from all platforms.

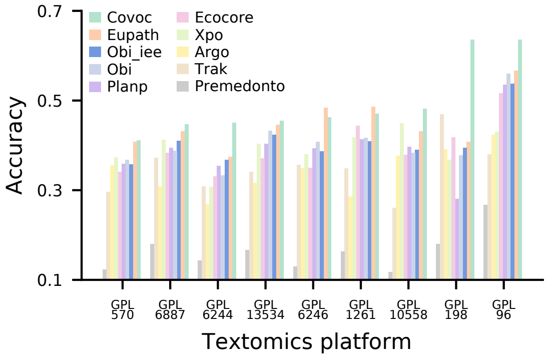

Figure 5: Bar plot showing the accuracy of filling the masked sentences of ten biomedical categories across 9 platforms using Textomics as the benchmark.

7 Related work

Our task is related to existing works that take structured data as the input and then generate the unstructured text. Different input data modalities and related datasets have been considered in the literature, including text triplets in RDF graphs (Gardent et al., 2017; Ribeiro et al., 2020; Song et al., 2020; Chen et al., 2020)), text-data tables (Lebret et al., 2016; Rebuffel et al., 2022; Dusek et al., 2020; Rebuffel et al., 2020; Puduppully and Lapata, 2021; Chen et al., 2020), electronic medical records (Lee, 2018; Guan et al., 2018), radiology reports (Wang et al., 2021; Yuan et al., 2019; Miura et al., 2021), and other continuous data modalities without explicit textual structures such as image (Lin et al., 2014; Cornia et al., 2020; Ke et al., 2019; Radford et al., 2021), audio (Drossos et al., 2020; Manco et al., 2021; Wu et al., 2021; Mei et al., 2021), and video (Li et al., 2021; Ging et al., 2020; Zhou et al., 2018; Li et al., 2020). Different from these structures, our dataset takes a high dimensional genomics feature matrix as input, which doesn't exhibit structure and is thus substantially different from other modalities. Moreover, our dataset is the first dataset that aims to convert genomics feature vector to textual summary. The substantial noise and high-dimensionality of genomics data matrices further pose unique challenges in text generation.

Our kNN-Vec2Text model is inspired by the recent success in applying kNN-based language models to machine translation (Khandelwal et al., 2021) and language models (Khandelwal et al., 2020; He et al., 2021; Ton et al., 2021). The main difference between our methods and their approaches is that while we try to leverage kNN in the genomics vector space to construct reference text, they use kNN in the text embedding space during the autoregressive generation process to help adjust the sample distribution. Some other methods can be used to generate text from vectors, such as (Bowman et al., 2016; Song et al., 2019; Miao and Blunsom, 2016; Montero et al., 2021; Zhang et al., 2019). Their inputs are latent vectors that need to be inferred from the data and do not have specific meanings, which are different from our gene feature vectors.

8 Conclusion and future work

In this paper, we have proposed a novel dataset Textomics, containing 22,273 pairs of genomics matrices and their corresponding textual summaries. We then introduce a novel task of Vec2Text based on our dataset. This task aims to generate the textual summary based on the gene feature vector. To address this task, we propose a novel method kNN-Vec2Text, which constructs the reference text using nearest neighbors in the gene feature vector space and then generates a new summary according to this reference text. We further introduce two applications that can be advanced using our dataset. One application aims at evaluating scientific paper similarity according to the similarity of its corresponding data summary, and the other application leverages our dataset to automatically generate masked sentences for scientific paper understanding.

To the best of our knowledge, Textomics and kNN-Vec2Text serve as the first large-scale genomics data description benchmark, and we envision it will be broadly applied to other natural language processing and biomedical tasks. On the biomedical side, we provide the benchmark to develop new NLP tools that can generate the description for a genomics data. Since each public genomics data needs a description, such tools will substantially accelerate this process. Also, descriptions generated from Textomics could contain new knowledge. While humans write the description almost solely based on that single dataset, description generation models jointly consider thousands of datasets, enabling the transfer of knowledge from other datasets. The generated description can guide biologists to write more informative descriptions, which ultimately leads to better and larger genomics description data. When biologists start to obtain the generated description from NLP tools, they will be able to write more informative descriptions with the assistance from these NLP tools. On the NLP side, the relationship between a summary and a dataset is analogous to the relationship between an abstract and a scientific paper. A high-quality summary ideally contains all perspectives of a study, including problems, methods, and discoveries. Moreover, our work will bridge the NLP and the genomics community and motivate people to analyze genomics data using NLP methods based on the multi-modality dataset introduced in this paper. Textomics could also be used to help scientific paper analysis tasks, such as paper recommendation (Bai et al., 2019), citation text generation (Luu et al., 2020), and citation prediction (Suzen et al., 2021).

Our method searches for the nearest neighbours by calculating the Euclidean distance between five-number summary vectors of the genomics feature matrices. However, this might lose useful information hidden in the original matrices. It's challenging and worth exploring end-to-end approaches that can learn embeddings from the genomics feature matrices instead of representing them as five-number summary vectors. On the Text2Vec side, one remaining challenge that could be the future direction of our work is to directly generate the whole genomics feature matrix instead of the five-number summary vector. Also, it would be interesting yet challenging to jointly learn the Text2Vec and the Vec2Text tasks, and one potential solution is to further decode the generated vector to reconstruct the embedding of summaries in Text2Vec, and leverage the resulted decoder to predict the embedding of text by using kNN method in the text embedding space. Also, it is interesting to jointly model data from multiple platforms, which might lead to beneficial results by transferring biological insights learned from different platforms.

References

Xiaomei Bai, Mengyang Wang, Ivan Lee, Zhuo Yang, Xiangjie Kong, and Feng Xia. 2019. Scientific paper recommendation: A survey. *IEEE Access*, 7:9324–9339.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*.

Yael Baran, Akhiad Bercovich, Arnau Sebe-Pedros, Yaniv Lubling, Amir Giladi, Elad Chomsky, Zohar Meir, Michael Hoichman, Aviezer Lifshitz, and Amos Tanay. 2019. Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome biology*, 20(1):1–19.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP (1)*, pages 3613–3618. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*, pages 10–21. ACL.

Liqun Chen, Guoyin Wang, Chenyang Tao, Dinghan Shen, Pengyu Cheng, Xinyuan Zhang, Wenlin Wang, Yizhe Zhang, and Lawrence Carin. 2019. Improving textual network embedding with global attention via optimal transport. In *ACL (1)*, pages 5193–5202. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: knowledge-grounded pretraining for data-to-text generation. In *EMNLP (1)*, pages 8635–8648. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *CVPR*, pages 10575–10584. Computer Vision Foundation / IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: an audio captioning dataset. In *ICASSP*, pages 736–740. IEEE.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Comput. Speech Lang.*, 59:123–156.

Ron Edgar, Michael Domrachev, and Alex E. Lash. 2002. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30 1.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*.

Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. COOT: cooperative hierarchical transformer for video-text representation learning. In *NeurIPS*.

Sara Goodwin, John D McPherson, and W Richard McCombie. 2016. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351.

Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. 2018. Generation of synthetic electronic medical record text. In *BIBM*, pages 374–380. IEEE Computer Society.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *EMNLP*.

Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. 2018. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):1–14.

Minoru Kanehisa and Peer Bork. 2003. Bioinformatics in the post-sequence era. *Nature genetics*, 33(3):305–310.

Lei Ke, Wenjie Pei, Ruiyu Li, Xiaoyong Shen, and Yu-Wing Tai. 2019. Reflective decoding network for image captioning. In *ICCV*, pages 8887–8896. IEEE.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *ICLR*. OpenReview.net.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *ICLR*. OpenReview.net.

Michael Kramer, Janusz Dutkowski, Michael Yu, Vineet Bafna, and Trey Ideker. 2014. Inferring gene ontologies from pairwise similarity data. *Bioinformatics*, 30(12):i34–i42.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *WMT@ACL*, pages 228–231. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213. The Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Scott Lee. 2018. Natural language generation for electronic health records. *CoRR*, abs/1806.01353.

Jacob H. Levine, Erin F. Simonds, Sean C. Bendall, Kara L. Davis, El ad D. Amir, Michelle D. Tadmor, Oren Litvin, Harris G. Fienberg, Astraea Jager, Eli R. Zunder, Rachel Finck, Amanda L. Gedman, Ina Radtke, James R. Downing, Dana Pe'er, and Garry P. Nolan. 2015. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. HERO: hierarchical encoder for video+language omni-representation pre-training. In *EMNLP (1)*, pages 2046–2065. Association for Computational Linguistics.

Yehao Li, Yingwei Pan, Jingwen Chen, Ting Yao, and Tao Mei. 2021. X-modaler: A versatile and high-performance codebase for cross-modal analytics. In *ACM Multimedia*, pages 3799–3802. ACM.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *ECCV (5)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Zequn Liu, Shukai Wang, Yiyang Gu, Ruiyi Zhang, Ming Zhang, and Sheng Wang. 2021. Graphine: A dataset for graph-aware terminology definition generation. In *EMNLP (1)*, pages 3453–3463. Association for Computational Linguistics.

Kelvin Luu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2020. Citation text generation. *ArXiv*, abs/2002.00317.

Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. 2021. Muscaps: Generating captions for music audio. In *IJCNN*, pages 1–8. IEEE.

Xinhao Mei, Qiushi Huang, Xubo Liu, Gengyun Chen, Jingqian Wu, Yusong Wu, Jinzheng Zhao, Shengchen Li, Tom Ko, H. Tang, Xi Shao, Mark D. Plumbley, and Wenwu Wang. 2021. An encoder-decoder based audio captioning system with transfer and reinforcement learning. In *DCASE*, pages 206–210.

Oren Melamud and Chaitanya Shivade. 2019. Towards automatic generation of shareable synthetic clinical notes using neural language models. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 35–45.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*.

Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. 2021. Improving factual completeness and consistency of image-to-text radiology report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Ivan Montero, Nikolaos Pappas, and Noah A. Smith. 2021. Sentence bottleneck autoencoders from transformer language models. In *EMNLP*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Trans. Assoc. Comput. Linguistics*, 9:510–527.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

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):9.

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):1–67.

Clément Rebuffel, Marco Roberti, Laure Soulier, Geoffrey Scoutheeten, Rossella Cancelliere, and Patrick Gallinari. 2022. Controlling hallucinations at word level in data-to-text generation. *Data Min. Knowl. Discov.*, 36(1):318–354.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In *ECIR (1)*, volume 12035 of *Lecture Notes in Computer Science*, pages 65–80. Springer.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Trans. Assoc. Comput. Linguistics*, 8:589–604.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *ACL*.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *ACL*, pages 7987–7998. Association for Computational Linguistics.

Tianbao Song, Jingbo Sun, Bo Chen, Weiming Peng, and Jihua Song. 2019. Latent space expanded variational autoencoder for sentence generation. *IEEE Access*, 7:144618–144627.

Neslihan Suzen, Alexander Gorban, Jeremy Levesley, and Evgeny Mirkes. 2021. Semantic analysis for automated evaluation of the potential impact of research articles.

Jean-Francois Ton, Walter A. Talbott, Shuangfei Zhai, and Joshua M. Susskind. 2021. Regularized training of nearest neighbor language models. *ArXiv*, abs/2109.08249.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Bin Wang and C.-C. Jay Kuo. 2020. SBERT-WK: A sentence embedding method by dissecting bert-based word models. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:2146–2157.

Sheng Wang, Jianzhu Ma, Michael Ku Yu, Fan Zheng, Edward W Huang, Jiawei Han, Jian Peng, and Trey Ideker. 2018. Annotating gene sets by mining large literature collections with protein networks. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium*, pages 602–613. World Scientific.

Wenlin Wang, Chenyang Tao, Zhe Gan, Guoyin Wang, Liqun Chen, Xinyuan Zhang, Ruiyi Zhang, Qian Yang, Ricardo Henao, and Lawrence Carin. 2019. Improving textual network learning with variational homophilic embeddings. In *NeurIPS*, pages 2074–2085.

Yixin Wang, Zihao Lin, Jiang Tian, Zhongchao Shi, Yang Zhang, Jianping Fan, and Zhiqiang He. 2021. Confidence-guided radiology report generation. *arXiv preprint arXiv:2106.10887*.

Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. 2021. Wav2clip: Learning robust audio representations from clip. *arXiv preprint arXiv:2110.11499*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Jianbo Yuan, Haofu Liao, Rui Luo, and Jiebo Luo. 2019. Automatic radiology report generation based on multi-view image fusion and medical concept enrichment. In *MICCAI (6)*, volume 11769 of *Lecture Notes in Computer Science*, pages 721–729. Springer.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019. Syntax-infused variational autoencoder for text generation. In *ACL (1)*, pages 2069–2078. Association for Computational Linguistics.

Yanjian Zhang, Qin Chen, Yiteng Zhang, Zhongyu Wei, Yixu Gao, Jiajie Peng, Zengfeng Huang, Weijian Sun, and Xuan-Jing Huang. 2020. Automatic term name generation for gene ontology: Task and dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4705–4710.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *CVPR*, pages 8739–8748. Computer Vision Foundation / IEEE Computer Society.

A Appendices

We provided more details here about our dataset and related experimental results here. In Table S1, we summarized the statistics information of 9 Textomics platforms. There are 3 different species

Platform	Species	#Matrix (All)	#Matrix (PMC)	# Matrix (Vec2Text)	#Feature	Missing rates
GPL96	Homo Sapiens	1,371	353	240	100K	0.19
GPL198	Arabidopsis Thaliana	1,081	194	250	100K	0.03
GPL570	Homo Sapiens	5,822	1,879	1,004	100K	0.12
GPL1261	Mus Musculus	4,563	1,326	1,059	100K	0.09
GPL6244	Homo Sapiens	1,831	659	307	100K	0.10
GPL6246	Homo Sapiens	2,366	850	388	100K	0.08
GPL6887	Mus Musculus	1,150	407	240	100K	0.09
GPL10558	Homo Sapiens	2,580	1,261	519	100K	0.11
GPL13534	Homo Sapiens	1,509	762	234	100K	0.26

Table S1: Statistics of the Textomics data. Each row is a sequencing platform in Textomics. All, PMC, Vec2Text represent number of samples without filtering, with associated PMC full text article, and after using automated filtering, respectively.

Textomics platform	GPL 96	GPL 198	GPL 570	GPL 1261	GPL 6244	GPL 6246	GPL 6887	GPL 10558	GPL 13534
Spearman correlation	0.36	0.20	0.24	0.34	0.44	0.45	0.22	0.38	0.30

Table S2: The result of Spearman correlation between gene data matrices and text summaries on 9 platforms.

Platform	BLEU-1	ROUGE-1	ROUGE-L	METEOR	NIST
GPL96	0.179	0.233	0.166	0.143	0.817
GPL198	0.198	0.257	0.192	0.168	0.889
GPL570	0.212	0.269	0.205	0.182	0.936
GPL1261	0.229	0.283	0.226	0.202	0.980
GPL6244	0.183	0.250	0.179	0.156	0.750
GPL6246	0.219	0.269	0.210	0.187	0.950
GPL6887	0.198	0.260	0.196	0.171	0.847
GPL10558	0.191	0.257	0.177	0.165	0.842
GPL13534	0.242	0.332	0.279	0.260	1.124

Table S3: More results on evaluating Vec2Text task on Textomics.

across 9 platforms, including Homo sapiens, Arabidopsis thailiana, and Mus musculus. #Sample (All) represents the entire number of samples for 9 platforms, #Sample (Vec2Text) represents the number of samples in the subset after BLEU filtering, and #Sample (PMC) represents the number of samples in the subset with full scientific articles.

We also represented the results of Spearman correlations between text-based similarity and vector-based simlarity across 9 platforms in Table S2. The Spearman correlations are all higher than 0.2 in every platform, which shows a substantial agreement between text-based similarity and vector-based similarity.

In Table S3, We represented the scores of different widely-used automatic metrics for word level sentence generation evaluation on Vec2Text task, including BLEU-1(Papineni et al., 2002), BLEU-2, ROUGE-1(Lin, 2004), ROUGE-L, METEOR(Lavie and Agarwal, 2007) and NIST(Doddington, 2002). The results indicated consistent improvement of our method over comparison approaches on different automatic metrics.