# Multimodal Review Generation with Privacy and Fairness Awareness

**Xuan-Son Vu**
Department of Computing Science
Umeå University, Sweden
sonvx@cs.umu.se

**Thanh-Son Nguyen**
Institute of High Performance Computing
A*STAR, Singapore
nguyents@ihpc.a-star.edu.sg

**Duc-Trong Le**
University of Engineering and Technology
Vietnam National University, Vietnam
trongld@vnu.edu.vn

**Lili Jiang**
Department of Computing Science
Umeå University, Sweden
lili.jiang@cs.umu.se

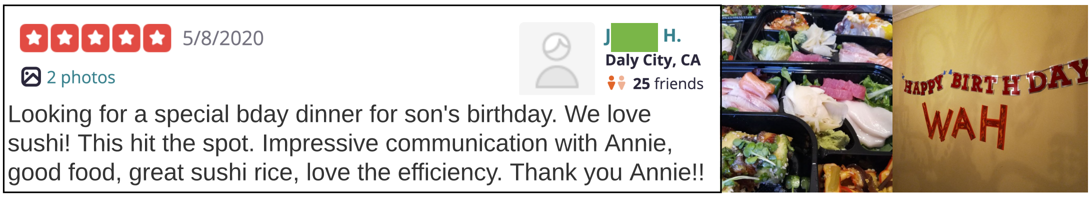
## Abstract

Users express their opinions towards entities (e.g., restaurants) via online reviews which can be in diverse forms such as text, ratings, and images. Modeling reviews are advantageous for user behavior understanding which, in turn, supports various user-oriented tasks such as recommendation, sentiment analysis, and review generation. In this paper, we propose MG-PriFair, a multimodal neural-based framework, which generates personalized reviews with privacy and fairness awareness. Motivated by the fact that reviews might contain personal information and sentiment bias, we propose a novel differentially private (dp)-embedding model for training privacy guaranteed embeddings and an evaluation approach for sentiment fairness in the food-review domain. Experiments on our novel review dataset show that MG-PriFair is capable of generating plausibly long reviews while controlling the amount of exploited user data and using the least sentiment-biased word embeddings. To the best of our knowledge, we are the first to bring user privacy and sentiment fairness into the review generation task. The dataset and source codes are available at https://github.com/ReML-AI/MG-PriFair.

## 1 Introduction

Users generate digital footprints when "traveling" on the internet. Modeling this behavioral data is useful to understand users' preferences. For example, Amazon infers users' preferences based on their views, add-to-card, or purchase actions. Likewise, online reviews explicitly manifest how users opine about business entities such as restaurants. Figure 1 shows an example of online reviews on Yelp.com, that expresses user's opinions about food and service of a sushi restaurant, along with images and rating score. Containing invaluable information of personal opinions, online reviews become an essential data source that is modeled in diverse tasks to comprehend users (Lackermair et al., 2013), e.g., sentiment analysis or review generation. In this paper, we study the task of review generation using multi modalities including image, user and entity information while taking into account user privacy and sentiment fairness. Specifically, we present a framework, namely MG-PriFair, which includes privacy and fairness controllers to preprocess data and a neural-based generation model to generate personalized reviews.

Reviews are user-generated contents that may contain personal information leading to privacy concerns. For example, the content and images of the review in Figure 1 signify sensitive information about the reviewer, i.e., *J. H.* in *Daly City* has a son named *Wah* who might be born on 8 May. This observes



★★★★★ 5/8/2020

2 photos

J[redacted] H.
Daly City, CA
25 friends

Looking for a special bday dinner for son's birthday. We love sushi! This hit the spot. Impressive communication with Annie, good food, great sushi rice, love the efficiency. Thank you Annie!!

Figure 1: An example of Online reviews on Yelp with personal information.

*Proceedings of the 28th International Conference on Computational Linguistics*, pages 414–425
Barcelona, Spain (Online), December 8-13, 2020

the problem of revealing personal information of an individual by observing outputs of the model trained using user-generated data (i.e. reviews) (Rocher and de Montjoye, 2019). To address this problem, we apply a privacy controller, which is a two-stage approach to minimize the use of personal information. First, we propose dpSENTI, a novel approach to learn differentially private word embeddings (dp-embedding), from which we infer user and entity representations (UERs). Next, we freeze these representations during the training process, to avoid further use of personal information. There are two levels of privacy protection in this task: individual privacy and model privacy. In the scope of this work, we focus on the former one.

Fairness is *"the absence of any bias"* (Ninareh Mehrabi, 2019). With the rapidly increasing number of machine learning applications in daily life, developing learning models that are *fair* with respect to sensitive attributes (e.g., gender, race) of the training data has become important. In the context of writing review, sentiment fairness is an issue raising at the individual level such as a restaurant, a product, or a dish. Sentiment bias can come from training reviews or external data. For example, when we use word embeddings trained using external data, some words describing a dish might be highly correlated to negative words causing the model to generate negative-sentiment reviews for the dish. In this paper, we focus on the bias causing by pretrained word embeddings models. Specifically, we propose an evaluation approach to measure sentiment bias for the food-review domain, thus assisting to select the least bias pretrained model.

Our contributions are three-fold. First, we propose a new dp-embedding (i.e., dpSENTI) approach for training privacy guaranteed word embeddings for the task of review generation. Secondly, we propose an evaluation approach for sentiment fairness in food-review domain. We also run the evaluation across multiple pretrained language models to evaluate their sentiment fairness for the domain. Thirdly, to the best of our knowledge, we are the first to introduce the notions of user privacy and sentiment fairness for the task of review generation. We evaluate extensively and present insights on multiple tasks ranging from dp-embeddings, sentiment fairness, to review generation. Additionally, the novel dataset is released with initial benchmark results for this task.

## 2 Related Work

We conduct a literature review in *text generation*, *review generation*, *user privacy* and *fairness* topics.

**Text Generation.** The closest tasks to review generation are *image captioning* and *review generation*. In image captioning, the objective is to automatically generate text to describe the content of an image via learning the correlation between vision and textual features (Xu et al., 2015). Xia et al. (2017) tackle the sequence generation problem, which applies neural machine translation and image captioning techniques with a new target-target attention mechanism on target sequences. In order to generate personalized captions, Chunseong Park et al. (2017) present Context Sequence Memory Network to take into account users' historical activities. Generally, review generation is different from image captioning since it requires additional input (i.e., user and entity information), and the target is not only to capture what is inside an image, but also to "express" opinions toward the entity being reviewed.

*Review generation* recently has received more attention. Nguyen et al. (2015) propose a graph-based approach to identify representative review snippets supporting to construct a review. Dong et al. (2017) propose an encoder-decoder network architecture that takes user/product attributes and ratings as input for personalized review generation. Ni and McAuley (2018) seek to learn aspect-aware user and item representations to generate reviews based on short phrases as input. In comparison with our proposed model, these reviewed works do not deal with visual input. Truong and Lauw (2019) introduce a multimodal review generation (MRG) to simultaneously predict ratings and generate reviews using information from users, items, and images. Their objective is to learn user preferences through predicting ratings and generate short reviews, whereas, we aim at generating relatively longer reviews. We compare our proposed model with MRG in Section 4.

**User Privacy**. Preserving user privacy has been studied for decades. The techniques of anonymization (Bayardo and Agrawal, 2005) and sanitization (Wang et al., 2009) have been widely applied. Differential privacy later emerged as the key privacy guarantee by providing rigorous, statistical guarantees against any inference from an adversary (Cynthia, 2006). Differential privacy has been applied in many research including text data (Abay et al., 2018). This motivates the use of differential privacy for review generation task. We propose to decrease the use of user information to reduce privacy leakage risk. There have been some works (McMahan et al., 2018; Vu et al., 2019) in learning differentially private language models. However, this paper aims at finding word representations for the review generation task, which has to preserve good sentiment. Therefore, we propose a different neural model to learn differentially private word embeddings for the sentiment classification task.

**Fairness.** There is an increasingly important concern as machine learning models are utilized to support decision making in high-stakes applications, e.g., mortgage lending, hiring, and prison sentencing (R. K. E., 2019). Kleinberg et al. (2018) present an empirical example for college admissions that the inclusion of variables (e.g., race) can increase both equity and efficiency. B Fish (2016) investigate algorithmic fairness and maintained the high accuracy of three learning algorithms while reducing the degree to reduce discrimination against individuals. ConceptNet Numberbatch 17.04 (Speer, 2017a) (hereafter ConceptNet) has been well known for having good semantic representation while addressing several word-embedding biases (e.g., gender bias and religious bias). However, it does not resolve sentiment bias for the food domain (Section 4). In this paper, we propose an evaluation approach that measures sentiment bias for the food-review domain to select a "less-bias" pretrained word embeddings model for the task of review generation. De-biasing sentiment bias is not in the scope of this work.

## 3 Multimodal Review Generation with Privacy and Fairness Awareness

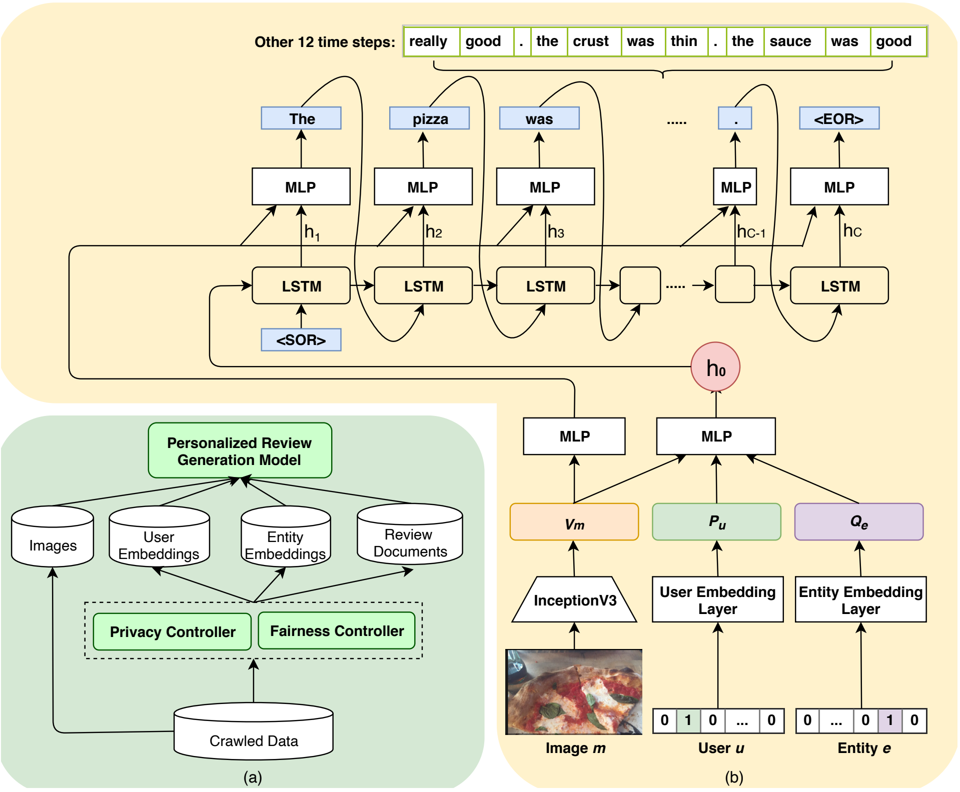


Figure 2: a) Architecture design of MG-PriFair which includes privacy and fairness controllers and a generation model. b) Our proposed personalized review generation model (PRGen).

In this section, we present our proposed multimodal review generation with privacy and fairness awareness framework, MG-PriFair. As shown in Figure 2a, MG-PriFair consists of three main components: privacy controller, fairness controller, and personalized review generation model (PRGen). *Privacy Controller* manages the use of personal information by injecting noise while learning differentially private representations for users and entities. The fairness controller measures sentiment bias of different word embeddings sets to select the least biased one. The preprocessed data is then passed to PRGen to train the generation model (Figure 2b). We formulate the problem of multimodal review generation as follows.

**Problem definition.** Let us denote the sets of users and entities as $\mathcal{U}$ and $\mathcal{E}$, respectively. The dataset $\mathcal{X}$ consists of tuples $(m, y, u, e)$, where image $m$ is associated with review document $y$ written by user $u \in \mathcal{U}$ for entity $e \in \mathcal{E}$. Each review document is a set of review sentences. Given dataset $\mathcal{X}$, the objective is to build a model that learns to generate a review document given an image $m$ and the associated information of user $u$ and entity $e$.

### 3.1 Privacy Controller

*Privacy Controller* controls the amount of user information that a learning algorithm can consume until the privacy budget is reached. The more data the model can consume, there is a higher risk of privacy leakage. To maintain the trade-off between user privacy and data utility for the review generation task, we introduce a privacy controller, namely dpSENTI, to act as a gateway protecting user privacy. Here, the controller is a differentially private neural model that learns to perform sentiment classification task based on user ratings. We assign 3 labels of NEG (rating 1 and 2), NEU (rating 3), and POS (rating 4 and 5). The training data for this task is similar to the training set of the review generation task except we have the labels of POS, NEU, NEG based on rating scores. We train a feed forward network consists of an embedding layer (hereafter dp-embedding), a pooling layer, and two linear layers. This architecture is simple yet efficient since it can capture semantic information of words in the dp-embedding layer for the review generation task. At the same time, it is also optimized for the sentiment classification to preserve sentiment for the review generation task. The whole model is trained with DP-SGD optimizer (Abadi et al., 2016) to protect user privacy. The dp-embedding layer is then used to extract user and entity representations for the text generation task. Intuitively, the dp-embedding layer is trained to prevent privacy leakage by injecting noise to the word vectors based on the differential privacy mechanism (Cynthia, 2006; Abadi et al., 2016). It is noted that, the dp-embedding layer is trained on the sentiment classification task; therefore, it preserves both user privacy and sentiment information, which are the main signals we need to feed into the review generation model. In fact, the dp-embedding layer is used to calculate dp-embeddings for both users and entities.

From the above problem definition, given a fixed dictionary $D$, a user $u \in \mathcal{U}$, $\mathcal{R}_u = \{r_{u_1}, ..., r_{u_M}\}$ denotes for the set of $M$ reviews written by user $u$. A dp-embedding $Emb_u$ for a given user $u$ is the averaging of all the embeddings of words written by $u$, i.e., $Emb_u = \frac{1}{M}\sum_{i=1}^{M}\sum_{w \in r_{u_i}} Emb_w$, where $w \in D$, $Emb_w$ is the word embedding of $w$. Since we use the Gausian mechanism implemented in DP-SGD of Abadi et al. (2016) to learn dp-embeddings at word-level, the average of these embeddings to constitute embeddings at user-level are also differentially private embeddings. Because the composition of a data-independent mapping $f$ with an $(\epsilon, \delta)$- differentially private algorithm $\mathcal{M}$ is also $(\epsilon, \delta)$-differentially private (Dwork and Roth, 2014).

### 3.2 Fairness Controller

*Fairness Controller* evaluates the sentiment bias (fairness) of word embeddings to be used in the generation model. Similar to Speer (2017b), we base on binary sentiment classification to measure the fairness of a pretrained word embedding set $Emb_X$ (e.g., GloVe (Pennington et al., 2014)). First, we train a binary sentiment classifier using two lists of positive ($L_1$) and negative ($L_2$) words from Hu and Liu (2004) as groundtruth, and $Emb_X$ as features. We split each list to 90% for training, and the rest 10% for testing. Using the trained classifier, we then test the sentiment bias of $Emb_X$ by extracting feature vectors for each testing word in a word list called Word Embedding Association Test (WEAT) (Caliskan et al., 2017) and compute the bias score.

WEAT is a list of words to measure how bias each word embedding set is for a certain bias category (e.g., ethnic and demographic). Due to the lack of WEAT list in sentiment-bias for food domain, we built our own WEAT list, namely R-WEAT, to measure sentiment-bias in our word embedding sets. We select 3 main food categories including (1) *Common food* (e.g., beef, chicken), (2) *Asian food* (e.g., rice, noodles), and (3) *Western food* (e.g., pasta, pizza). These terms are selected based on two criteria: they are either dish name or ingredients, and they must appear frequently in our dataset. In total, group (1), (2), (3) have 19, 49, 35 words, respectively. Based on these selected words, the trained classifier is used to predict sentiment. Then, we run a hypothesis testing using the Ordinary Least Squares (OLS) estimator implemented in (Seabold and Perktold, 2010) to get the F-statistic value (hereafter F-bias value) of the R-WEAT list in the trained sentiment classifier. F-bias value is the ratio of the variation between categories to the variation within categories. In other words, it represents the degree of sentiment fairness (the lower the better) of each word embedding set regarding the R-WEAT list.

### 3.3 Personalized Review Generation Model (PRGen)

Figure 2b shows our proposed generation model. PRGen receives as input an image $m$, user $u$ and entity $e$, and outputs a review document $y = \{y_1, y_2, ..., y_C\}$, where $y_t \in \mathbb{R}^K$, $K$ is the vocabulary size and $C$ is the length of the review document. To capture sequential information, recurrent neural networks (e.g., long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), ppRNNs (Tran et al., 2018)), can be applied. Here, LSTM is used to learn by generating one word at each time step. At time step $t$, the output $y_t$ is computed based on the vision features and the current hidden state $h_t$:

$$y_t = W_{o_2}\text{ReLU}(W_{o_1}(\tilde{V}_m \oplus h_t)) \tag{1}$$

where $h_t \in \mathbb{R}^{d_h}$ is the $d_h$-dimensional hidden state at time $t$, $\tilde{V}_m \in \mathbb{R}^{d_{\tilde{v}}}$ is the $d_{\tilde{v}}$-dimensional vector computed from vision features. ReLU is the Rectified Linear Unit function, $\oplus$ is the concatenation operator. $W_{o_1} \in \mathbb{R}^{d_o \times (d_{\tilde{v}} + d_h)}$ and $W_{o_2} \in \mathbb{R}^{K \times d_o}$ are parameters to be learnt during training. The probability of selecting word $i$ at time step $t$ is computed using the softmax function:

$$p(y_t^i) = \frac{\exp(y_t^i)}{\sum_{j \in K} \exp(y_t^j)} \tag{2}$$

where $y_t^i$ is the value of the $i^{\text{th}}-$element in vector $y_t$. $\tilde{V}_m$ is defined as $\tilde{V}_m = \text{tanh}(W_v V_m)$, where $V_m \in \mathbb{R}^{d_v}$ is the vision features for image $m$ that is extracted using a pretrained convolutional neural network (CNN)-based model. $W_v \in \mathbb{R}^{d_{\tilde{v}} \times d_v}$ are parameters to be learnt during training.

The hidden state $h_t$ at time $t$ is updated follows the formulation in (Zaremba et al., 2014). LSTM's parameters, $\Theta_{\text{LSTM}}$, are learnable during training. The initial hidden state $h_0$ is computed based on the features of the input image, user, and entity: $h_0 = tanh(W_0(V_m \oplus P_u \oplus Q_e))$. Here, $P_u = P\Pi_u, \in \mathbb{R}^{d_\mathcal{U}}$ and $Q_e = Q\Psi_e, \in \mathbb{R}^{d_\mathcal{E}}$ are the embeddings for user $u$ and entity $e$ respectively. $\Pi_u \in \mathbb{R}^{|\mathcal{U}|}$ and $\Psi_e \in \mathbb{R}^{|\mathcal{E}|}$ are one-hot vectors for $u$ and $e$. $P \in \mathbb{R}^{d_\mathcal{U} \times |\mathcal{U}|}$ and $Q \in \mathbb{R}^{d_\mathcal{E} \times |\mathcal{E}|}$ are the embedding matrices for user and entity, respectively. Embedding matrices $P$ and $Q$ can be initialized randomly or by pretrained user and entity embeddings, with/without fine-tuning. We compare the different strategies of using user and entity embeddings in Section 4.

**Training PRGen.** During training, PRGen takes as input a tuple $(m, y, u, e) \in \mathcal{X}_{\text{train}}$, where $\mathcal{X}_{\text{train}} \subset \mathcal{X}$ is the *train* set, and generates a review document $\hat{y}$. The model is trained with teacher-forcing and the objective is to minimize the cross entropy loss between the groundtruth $y$ and the generated $\hat{y}$:

$$\arg\min_{\Omega} \frac{1}{|y|} \sum_{t=1}^{|y|} \sum_{i=1}^{K} -y_t^i \log p(\hat{y}_t^i) \tag{3}$$

where $y_t \in \mathbb{R}^K$ is the corresponding one-hot vector for the word at position $t^{\text{th}}$, $p(\hat{y}_t^i)$ is computed using Equation 2, and $\Omega = \{\Theta_{\text{LSTM}}, P, Q, W_0, W_v, W_{o_1}, W_{o_2}\}$ are the trainable parameters which are learnt during the training process.

**Inference.** During inference (testing), PRGen is given only an image and information of the corresponding user and entity. The model generates one token at a time, starting with <*SOR*>(start-of-review) token. The generated token at a step will be the input token for the next step. The model stops when generating <*EOR*>(end-of-review) token or exceeding a predefined length constraint.

## 4 Experiments

### 4.1 Experimental Setup

**Dataset.** The task of multimodal review generation requires data of images with corresponding reviews, users, and entities information. Although there are existing review datasets, such as Amazon product data and Yelp Dataset Challenge, none of them provides enough information for our task. They either do not have review images or not provide enough information to map an image to its original review. Therefore, we construct a new dataset from Yelp.com that contains restaurant reviews for seven different English speaking cities. When posting a review on Yelp, users can choose to attach image(s) and opt to write captions for the images. For this task, we only keep the reviews that have images containing captions. Data from the cities are combined and used as a whole. Eventually, there are about 154K reviews, 69K users, 6K entities, and 237K images.

In order to form the groundtruth review documents for an image, we first remove all the irrelevant sentences in the corresponding review and group $n$ consecutive relevant-sentences as a groundtruth review document ($n = 3$). We assume that a sentence is relevant to an image if it is similar to the image's caption. The task of finding relevant sentences for an image becomes text matching problem. To capture the semantic similarity, we use Spacy (spacy.io) with pretrained word embeddings to calculate the similarity between an image caption and a review sentence. A threshold of 0.01 is used to determine relevant sentences. After matching images to review documents, the dataset collectively has more than one million groundtruth data tuples of *(image, review document, user, entity)*. We split the dataset into *train*, *validation*, and *test* sets. We keep about ten thousand images each for validation and test sets; the rest are used for training. On average, each image has about five groundtruth review documents.

**Vision.** Vision features of images are extracted using *Inception-v3* (Szegedy et al., 2016) with weights pretrained on ImageNet with the $include\_top$ parameter is set to False. Therefore, input size for this model is 299x299. Output vision dimension is 2048.

**User and entity representations.** Users have their own writing styles, and each entity normally receives a few major groups of opinions. This information is contained in the prior knowledge, i.e., the reviews they have written (users) or received (entities). We adopt *DocumentPoolEmbeddings* implemented in Flair toolkit (Akbik et al., 2018) to pretrain user and entity representations using prior knowledge in which, each user (or entity) is represented by the reviews that the user (or entity) has. To have enough training signal, we only learn embeddings for users and entities that have at least 10 reviews in the train set. We have 20,228 such users which covers 83.05%, 76.46% and 75.43% of data tuples in train, validation, and test sets respectively. For entity, the number is 5,126 covering more than 99% of data tuples for all the three sets. The rest of the users and entities will be treated as *unknown* users and entities, respectively.

**Settings.** We select the top 10,000 frequent tokens for the vocabulary. We use Adam optimizer with initial learning rate of $1e^{-4}$, decay rate of 0.9 after every 20 epochs, starting from epoch 30. LSTM hidden dimension is 256. Only reviews having length between 5 and 50 words are used for training. We use beam search (width = 3) to generate reviews during inference.

## 4.2 Evaluating the Generation Model

### 4.2.1 Ablation experiments

**Settings.** We evaluate different variations of PRGen, including: (1) Vision-only model (RGen); and (2) with personalized settings called PRGen. RGen has the same structure as PRGen (Figure 2b) but removed user and entity embedding layers. This is to evaluate the effect of using user and entity information for the task. PRGen with different manners of utilizing user and entity representations (UERs) where UERs are randomly initialized and finetune during training (PRGen-RY), or UERs are pretrained using prior knowledge (Section 4.1) and are fixed (PRGen-PN) or finetuned (PRGen-PY) during training.

For each variation, we report the results testing with three pretrained word embeddings models including GloVe (Pennington et al., 2014) (dim=300), BERT (Devlin et al., 2019) (dim=768), and RoBERTa (Liu et al., 2019) (dim=768). Since BERT and RoBERTa are contextual embeddings, for each word, we average all tokens of a word to get its vector.

**Evaluation metrics.** To evaluate the generation models, we use standard metrics for the text generation task, including Bleu (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE-L (Lin and Och, 2004), and CIDEr (Vedantam et al., 2015). We use COCO evaluator (Chen et al., 2015) to compute these metrics. The results using these metrics are reported in percentage (except for CIDEr).

Table 1: Ablation experiment results showing the impact of user and entity representations and prior knowledge on the task of review generation. Subscripts *G*, *B*, *R* denote for GloVe, BERT, and RoBERTa, respectively. The metric annotations B, MET, ROU, CID stand for Bleu, METEOR, ROUGE-L, and CIDEr, respectively. N/A stands for not applicable.

| Method | User/Entity Representations | | B-1 | B-2 | B-3 | B-4 | MET | ROU | CID |
|---|---|---|---|---|---|---|---|---|---|
| | Initialized with | Finetune | | | | | | | |
| RGen$_G$ | N/A | N/A | 35.83 | 15.26 | 6.73 | 3.02 | 7.56 | 18.31 | 1.96 |
| RGen$_B$ | | | 36.47 | 15.84 | 7.09 | 3.09 | 7.58 | 18.49 | 1.92 |
| RGen$_R$ | | | 34.16 | 14.88 | 6.53 | 2.90 | 7.39 | 18.46 | 1.76 |
| PRGen$_G$-RY | Random | Yes | 37.33 | 16.25 | 7.52 | 3.55 | 7.94 | 19.16 | 3.00 |
| PRGen$_B$-RY | | | 36.82 | 16.47 | 7.62 | 3.64 | 7.81 | 18.81 | 2.44 |
| PRGen$_R$-RY | | | 37.36 | 17.12 | 8.11 | 3.83 | 7.80 | 19.42 | 2.71 |
| PRGen$_G$-PN | Pretrained | No | 37.40 | 16.58 | 7.67 | 3.58 | 7.84 | 19.24 | 2.42 |
| PRGen$_B$-PN | | | 37.69 | 16.91 | 7.93 | 3.75 | 7.87 | 19.39 | 2.66 |
| PRGen$_R$-PN | | | 36.86 | 16.86 | 8.05 | 3.89 | 7.70 | **19.68** | 2.76 |
| PRGen$_G$-PY | Pretrained | Yes | 38.48 | 17.47 | 8.22 | 3.90 | 8.09 | 19.61 | 3.16 |
| PRGen$_B$-PY | | | **39.03** | **17.96** | 8.53 | 4.16 | **8.29** | 19.59 | 3.26 |
| PRGen$_R$-PY | | | 38.22 | 17.85 | **8.95** | **4.83** | 8.23 | 19.65 | **3.88** |

**Results.** Table 1 shows that UERs are useful for review generation since all the variances of PRGen outperform RGen regarding all the evaluation metrics. Even when the UERs are only initialized randomly and finetuned during training (PRGen-RY), the model seems to be able to encode useful user and entity information for the generation task. We further investigate the impact of prior knowledge on generating reviews. As shown in Table 1, UERs pretrained using prior knowledge are useful even when they are not finetuned. PRGen-PN are comparable to the one optimized for the generation task, i.e., PRGen-RY. The results are further improved when the pretrained UERs are finetuned during training the generation models. PRGen-PY outperforms PRGen-RY and PRGen-PN for different settings of word embeddings regarding almost all the metrics (except for ROUGE-L). PRGen$_B$-PY performs the best in Bleu-1, Bleu-2, and METEOR. For Bleu-3, Bleu-4, and CIDEr, using RoBERTa achieves the best. The results clearly show that prior knowledge contributes useful information for the review generation task.

### 4.2.2 Evaluating against text generation baselines

**Baselines.** We compare PRGen against text generation baselines in both image captioning and review generation: (1) ShowNTell (Vinyals et al., 2015): a well-known approach for the image captioning task that consists of a vision CNN-based followed by a language generator LSTM; and (2) MRG (Truong and Lauw, 2019): a multimodal review generation that simultaneously predict ratings and generate reviews.

**Settings.** To have fair comparison, all the models use GloVe embeddings. Our model uses the PRGen$_G$-PY setting. ShowNTell and MRG requires a lot of memory that could not feed our full training set to GPU, we only use 40% of the training set to train the models (including ours).

**Evaluation metrics.** We evaluate the models regarding the capability of generating reviews. A review should contain sentiment (subjectivity) and does not necessarily always describe only the content of an image. Therefore, in addition to Bleu-4, we also measure the readability of generated reviews including sentences' average length (number of words in a sentence), *sentiment polarity*, the *subjectivity* and the number of *grammar errors*. We use TextBlob (Loria, 2018) to analyse sentiment polarity and subjectivity of generated reviews. To measure the grammatical quality of generated reviews, we use LanguageTool (Naber, 2007). We ignore typographical and miscellaneous errors such as capitalization and white space before the full stop, due to the manner the reviews was constructed.

**Results.** Table 2 clearly shows that our model outperforms the baselines in terms of Blue_4, sentiment polarity and subjectivity. Among the three models, our model has the most capability of generating subjective reviews (with the lowest number of Zeros polarity and subjectivity). When it comes to sentence length, MRG tends to generate long sentences (on average of 50 words per sentence). ShowNTell gener-

Table 2: Comparison between our model (PRGen$_G$-PY) and the baselines in image caption (ShowNTell) and review generation (MRG) in terms of Bleu-4, sentiment polarity, subjectivity, grammar errors (GramErr) and sentences' length (AvgLen). The superscript $^*$ marks the metrics in which the lower value the better. POS, NEG, Avg and GT stand for positive, negative, average and groundtruth, respectively.

| Model | Bleu-4 | *Polarity* | | | *Subjectivity* | | GramErr$^*$ | AvgLen |
|---|---|---|---|---|---|---|---|---|
| | | POS | NEG$^*$ | #Zeros$^*$ | Avg | #Zeros$^*$ | | (GT:15) |
| ShowNTell | 2.58 | 0.44 | -0.22 | 1302 | 0.51 | 1157 | 3.86 | 31 |
| MRG | 1.07 | 0.41 | **-0.29** | 910 | **0.65** | 760 | **0.46** | 50 |
| PRGen$_G$-PY | **2.77** | **0.55** | -0.21 | **424** | **0.65** | **362** | 0.94 | 10 |

Table 3: Different settings of dpSENTI and the performances on a downstream task of sentiment classification on Hu and Liu (2004)'s dataset. DET denotes the use of a fixed 2% of training data.

| Setting | $(\epsilon, \delta)$-dp | seq_len | vocab_size | emb_dim | F-value bias | Sentiment Accuracy |
|---|---|---|---|---|---|---|
| DET-64 | $(4.97, 1e-05)$ | 256 | 10K | 64 | **0.171** | 68.53 |
| DET-300 | $(\mathbf{4.11}, 1e-05)$ | 200 | 5K | 300 | 1.293 | **70.44** |

ates shorter sentences (i.e., on average, 31 words per sentence) but still double that of groundtruth (i.e., 15 words per sentence). Our model generates reasonable-length sentences with 10 words per sentence, compared to that of the groundtruth sentences. Regarding grammatical test, ShowNTell has the most serious problem with the average count of 3.86 grammatical errors per review while in PRGen$_G$-PY and MRG, the values are less than 1.

## 4.3 Evaluating Privacy and Fairness Controllers

### 4.3.1 Privacy Controller

We design two settings of dpSENTI for training the word embedding layer in a deterministic way called DET-64 and DET-300. For both settings, a fixed number of 2% of training samples are selected for the training process. For DET-64, the word embedding layer has 64 dimensions, while the DET-300 has 300 dimensions. Table 3 details experimental results. It clearly shows that the DET-300 is a better option for the review generation task since it has higher sentiment accuracy and consumes less user privacy (i.e., the $\epsilon$ value is smaller). The F-bias value of DET-64 is smaller suggesting that it contains less sentiment information (i.e., lower sentiment accuracy) and hence, it posses less sentiment bias.

### 4.3.2 Fairness Controller

In this section, we examine the sentiment-bias (fairness) of different pretrained word embeddings models. In addition to GloVe, BERT, RoBERTa, and dpSENTI, we also include ConceptNet (Speer et al., 2017) and Word2Vec (Mikolov et al., 2013). The former is widely used since its word embeddings can capture both semantic relationship between words while possessing less biases for such as *gender* and *ethnic*. Therefore, ConceptNET is a potential out-of-the-box solution for sentiment fairness. Regarding Word2Vec, it is one of the most popular methods to learn word embeddings using shallow neural network. Here, we include Word2Vec to compare to other similar learning methods such as GloVe.

Figure 3 shows the sentiment predictions of each word embeddings model for the words in R-WEAT list. The sentiment score for a word is the subtraction of log probability of positive and negative predictions. F-bias value (F) and classification accuracy (A) for each word embeddings model are reported on the corresponding sub-figure in the form of F/A. ConceptNet achieves the best classification accuracy but its F-bias has the highest value of 14.61 (i.e., most biased). As mentioned in (Speer, 2017a), they apply the de-bias algorithm to protect pre-defined biases. Hence, it is reasonable that ConceptNet has a high fairness issue (i.e., high F-bias value) on our "unseen" R-WEAT list. Therefore, out-of-the-box solution is not easy to achieve sentiment fairness in this case. BERT, however, achieves the best fairness score even though it is not intended to deal with sentiment bias. Cummings et al. (2019) show that it is not easy to have both privacy-fairness guarantee with differential privacy but they can be adjusted. We find that the later point is valid in the food domain as dpSENTI achieves the runner up fairness result.

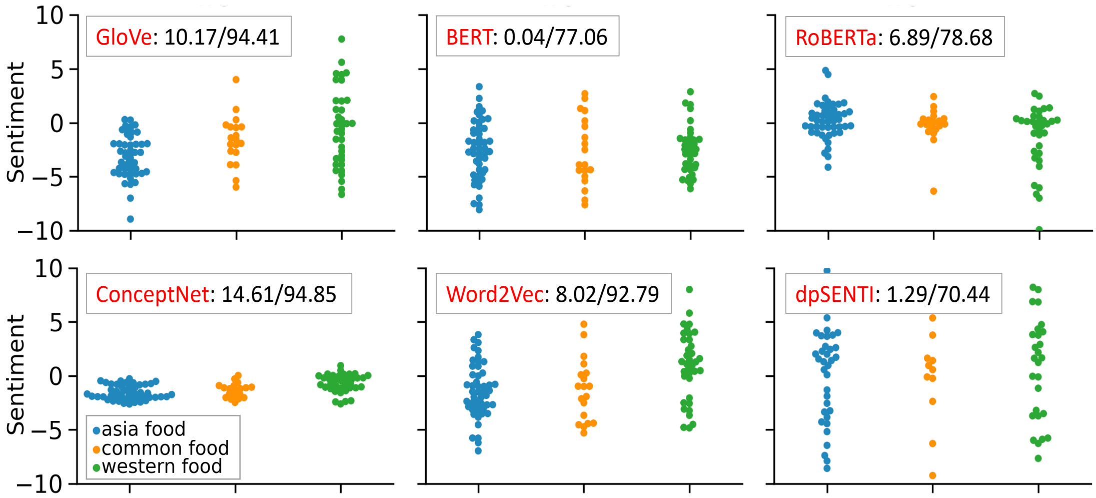


Figure 3: Fairness evaluation based on R-WEAT list for different word embeddings models. F-bias value (F) and classification accuracy (A) for each word embeddings model are reported in the form of F/A. Visualization method was inspired by Speer (2017b).

Table 4: Performance trade-off when taking into account fairness (Fair) and privacy (Priv) awareness. MG-PriFair uses BERT for word embeddings and dpSENTI for user and entity representations, which were fixed during the training process.

| Method | Fair | Priv | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | ROUGE_L | CIDEr |
|---|---|---|---|---|---|---|---|---|---|
| PRGen$_R$-PY | No | No | 38.22 | 17.85 | **8.95** | **4.83** | 8.23 | 19.65 | **3.88** |
| PRGen$_C$-PY | No | No | **39.99** | **18.54** | 8.80 | 4.18 | **8.30** | 19.94 | 3.37 |
| PRGen$_B$-PY | Yes | No | 39.03 | 17.96 | 8.53 | 4.16 | 8.29 | 19.59 | 3.26 |
| MG-PriFair | Yes | Yes | 37.85 | 16.86 | 7.37 | 3.05 | 7.75 | **20.45** | 1.64 |

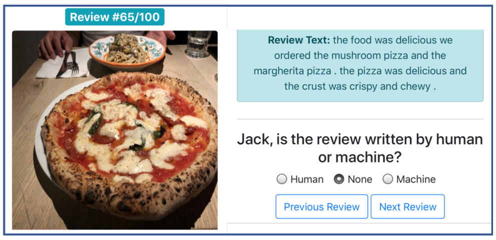
### 4.3.3 Evaluate our proposed framework: MG-PriFair

The main goal of our proposed framework, MG-PriFair, is to generate reviews with privacy and fairness awareness. We explore different word embedding sets regarding privacy and fairness criteria for the goal. Among the tested embedding sets, BERT achieves the best trade-off between model's performance and fairness. As shown in Figure 3, BERT achieves the best fairness score (lowest F-bias value) while obtaining the best Bleu-1, Bleu-2 and METEOR on the review generation task among GloVe, BERT, and RoBERTa as shown in Table 1. To further taking privacy into account, we use our newly trained dpSENTI embeddings to obtain the pretrained user and entity representations.

To evaluate the trade-off, we compare the models having different levels of controlling fairness and privacy: (1) without Fairness and Privacy (using RoBERTa (PRGen$_R$-PY) and ConceptNet (PRGen$_C$-PY)); (2) with Fairness, without Privacy (using BERT (PRGen$_B$-PY)); and (3) with Fairness and Privacy: MG-PriFair which uses BERT for word embeddings and dpSENTI for user and entity representations which are fixed at training time. Table 4 shows the trade-off of the performances when dealing with fairness and privacy. The models without fairness and privacy achieve the best performances in all the metrics. The performance slightly decreases when having fairness and continues decreasing when adding privacy control. These results are expected as more constraints are applied to deal with fairness and privacy, making it difficult to train generation model. Nevertheless, MG-PriFair's performance is comparable to the others' given the fact that user and entity representations are trained on a completely different task and are fixed during training the generation model.

## 4.4 Qualitative Results

To qualitatively evaluate the reviews generated by our proposed models, we conduct a user study shown in Figure 4. All the models use BERT embeddings. The images are randomly selected from the *test* set. For selecting groundtruth, we purposely choose reviews that have the length between 5 to 50 tokens (to be the same as the generated reviews' constraint). We recruited five participants for this study. Figure 4b



Review #65/100

**Review Text:** the food was delicious we ordered the mushroom pizza and the margherita pizza . the pizza was delicious and the crust was crispy and chewy .

Jack, is the review written by human or machine?

○ Human ● None ○ Machine

Previous Review | Next Review

(a) Graphic User Interface for User Study

| | | Predicted Result | |
|---|---|---|---|
| | | Human | Machine |
| Actual | Human | 62% | 38% |
| Result | Machine | 49% | 51% |

(b) Voting results

Figure 4: Qualitative evaluations design and results of generated reviews by different approaches. Here we have 100 images and reviews, in which half of reviews generated by our personalized review generation model. Each participant votes all images, to justify if a review was written by human or machine.

show the voting results, where the overall accuracy is 56%. On average, 49% of the machine-generated reviews were voted as human-generated. Moreover, only 62% of human-generated reviews were correctly voted. In addition, the average correlation among the participants is only 0.11. In other words, voting a review as human or machine generated is arguable among the participants. All the clues imply that differentiating our generated reviews from human-generated reviews is difficult.

## 5 Comparing with Text Generation Baselines

Figure 5 shows examples of reviews generated by our model (PRGen$_B$-PY) and the two baselines. ShowNTell has grammatical issues affecting the readability of the generated reviews and the contents are also not relevant to the input images. MRG is able to generate grammatically correct reviews, but tend to generate similar contents for different images. Our model, however, is able to generate relevant and grammatically correct reviews. Sometimes, generated reviews describe other aspects instead of only describing the image, e.g. the 1$^{st}$ image. With these results, our proposed review generation model demonstrates the efficiency on generating reviews with quality and readability.

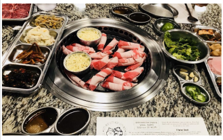
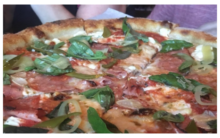
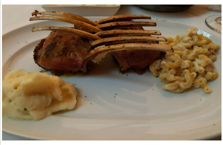
| Image | PRGen$_G$-PY | ShowNTell | MRG |
|---|---|---|---|
|  | i 've been here for a few times and i have never been disappointed . the service was great and the food was great . | to there have you for forfor personi can order on which crab ) , they ordered tuna shrimp soup it yum | and spicy tuna roll was in the world i 'm not sure what they 're in but the chicken was tender and the sauce was just a little bit salty . |
|  | my husband and i went here for a late night and it was a great experience . we ordered the margherita pizza and it was delicious . the pizza was good but the crust was thin and chewy . | was a good night i was n't you food was n't to be a i was a year minutefor i ! ! is i i food was good good it i , were open to open store , but i can see if you 're in for | and spicy tuna sandwich in vegas loved the atmosphere great the service was very friendly and attentive . |
|  | i 've been wanting to try this place for a while i 'm in vegas . i have to say this place is a great place to eat . | was very pictures order of to had than to was i was good good to other places i 've had other restaurants restaurants mac was very a highlight of meal of lobster meat opinion | and spicy tuna i 'm hoping it would be more but i 'm not sure if it 's a dish however the taste buds were the same thing but i just say it was n't bad . |

Figure 5: Reviews generated by our model (PRGen$_G$-PY) and the baselines (ShowNTell and MRG).

## 6 Conclusion

This paper proposed MG-PriFair, a multimodal neural-based framework, to automatically generate personalized reviews to understand user behaviors. MG-PriFair is aware of user privacy and sentiment fairness. Our extensive empirical experiments show the efficiency of the proposed framework in generating plausible reviews while taking into account user privacy and sentiment fairness. To the best of our knowledge, we are the first to raise the concerns of user privacy and sentiment bias for the review generation task. As a future work, the privacy of images can be concerned. For example, a taken photo of a restaurant may capture human faces. One potential solution to protect image-level privacy is to detect regions in images having sensitive-personal information and exclude those before sending to the model.

## References

M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, and L. Zhang. 2016. Deep Learning with Differential Privacy. *ArXiv e-prints*.

Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. 2018. Privacy preserving synthetic data release using deep learning. In *ECML PKDD*, pages 510–526. Springer.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING*, pages 1638–1649.

AD Lelkes B Fish, J Kun. 2016. A confidence-based approach for balancing fairness and accuracy. In *SDM*, pages 144–152.

R. J. Bayardo and Rakesh Agrawal. 2005. Data privacy through optimal k-anonymization. In *ICDE*, pages 217–228.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183–186.

X. Chen, H. Fang, TY Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server.

Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. 2017. Attend to you: Personalized image captioning with context sequence memory networks. In *CVPR*, pages 895–903.

Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. 2019. On the compatibility of privacy and fairness. In *UMAP'19 Adjunct*, pages 309–315. ACM.

Dwork Cynthia. 2006. Differential privacy. ICALP, pages 1–12.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *WMT*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *ACL*, pages 623–632.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, pages 211–407.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*, New York, NY, USA. ACM.

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. 2018. Algorithmic fairness. *AEA Papers and Proceedings*, 108:22–27, May.

Georg Lackermair, Daniel Kailer, and Kenan Kanmaz. 2013. Importance of online product reviews from a consumer's perspective. *Advances in economics and business*, 1(1):1–5.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, page 605.

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

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2:1–73.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private language models. *ICLR*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Daniel Naber. 2007. Languagetool. *https://www. languagetool. org*.

Thanh-Son Nguyen, Hady W Lauw, and Panayiotis Tsaparas. 2015. Review synthesis for micro-review summarization. In *WSDM*, pages 169–178.

Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *ACL*, pages 706–711.

Nripsuta Saxena Kristina Lerman Aram Galstyan Ninareh Mehrabi, Fred Morstatter. 2019. A survey on bias and fairness in machine learning. In *arXiv:1908.09635*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Bellamy; et al R. K. E. 2019. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of R&Development*, 63:4:1 – 4:15, July-Sept.

Hendrickx J.M. Rocher, L. and Y. de Montjoye. 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Natural Communications*, 10(3069).

Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Robyn Speer. 2017a. Conceptnet numberbatch 17.04: better, less-stereotyped word vectors. *ConceptNet blog. April*, 24. `http://blog.conceptnet.io/posts/2017/conceptnet-numberbatch-17-04-better-less-stereotyped-word-vectors/` (visited: 2020-04-22).

Robyn Speer. 2017b. How to make a racist ai without really trying. *ConceptNet blog. July*. `http://blog.conceptnet.io/posts/2017/how-to-make-a-racist-ai-without-really-trying/` (visited: 2020-04-22).

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826.

Son N. Tran, Qing Zhang, Anthony Nguyen, Xuan-Son Vu, and Son Ngo. 2018. Improving recurrent neural networks with predictive propagation for sequence labelling. In Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa, editors, *Neural Information Processing*, pages 452–462, Cham. Springer International Publishing.

Quoc-Tuan Truong and Hady Lauw. 2019. Multimodal review generation for recommender systems. In *WWW*, pages 1864–1874.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164.

Xuan-Son Vu, Son N. Tran, and Lili Jiang. 2019. dpugc: Learn differentially private representation for user generated contents. In *CICLing*.

Rui Wang, XiaoFeng Wang, Zhou Li, Haixu Tang, Michael K. Reiter, and Zheng Dong. 2009. Privacy-preserving genomic computation through program specialization. In *CCS*, pages 338–347.

Yingce Xia, Fei Tian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Sequence generation with target attention. In *ECML PKDD*, pages 816–831. Springer.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint:1409.2329*.