Spring 2023

Personalized Tweet Recommendation Using Users' Image Preferences

Shashwat Avinash Kadam
San Jose State University

Personalized Tweet Recommendation Using Users' Image Preferences

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Shashwat Avinash Kadam

May 2023

The Designated Project Committee Approves the Project Titled

Personalized Tweet Recommendation Using Users' Image Preferences

by

Shashwat Avinash Kadam

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2023

Dr. William Andreopoulos	Department of Computer Science
Dr. Katerina Potika	Department of Computer Science
Dr. Genya Ishigaki	Department of Computer Science

ABSTRACT

Personalized Tweet Recommendation Using Users' Image Preferences

by Shashwat Avinash Kadam

In the era of information explosion, the vast amount of data on social media platforms can overwhelm users. Not only does this information explosion contain irrelevant content, but also intentionally fabricated articles and images. As a result, personalized recommendation systems have become increasingly important to help users navigate and make sense of this data. We propose a novel technique to use users' image preferences to recommend tweets. We extract vital information by analyzing images liked by users and use it to recommend tweets from Twitter. As many images online have no descriptive metadata associated with them, in this framework, we also provide an opportunity for the users to annotate the images they liked with the hashtags of the recommended tweets.

Index terms--*Annotation, image captioning, personalization, recommendation system, twitter recommendation*

ACKNOWLEDGMENTS

I want to thank Professor William Andreopoulos for their unparalleled support throughout the semester. I would also like to thank Professor Andreopoulos for their help with the Twitter API and dataset required for the project. I would like to express my gratitude to the department of Computer Science for giving me access to the high performance computing facility critical for this project.

I would also like to thank my committee members Professor Katerina Potika and Professor Genya Ishigaki for their guidance with this project.

Lastly, I would like to thank my parents, friends, and my sister for their encouragement and support.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Due to easy accessibility and potent computing devices, our world has been firmly connected. Statistically speaking, approximately 2/3rd of the entire human population can access computers and the internet [1]. Everyone not only can read from the internet but also can write on it. While the former feature can be innocuous, the latter can overwhelm the world with the gigantic amount of information. Since the internet is available at the fingertips, millions of gigabytes of digital content are uploaded every second, causing an information explosion [2]. To be precise, according to a study conducted by Data Reportal [3], as of 2021, there are over 4 billion active users on various forms of social media, and the netizens generate over a trillion megabytes of data every day [4].

Because of this information explosion, we surround ourselves with irrelevant content. Today, obtaining information about rocket science is as easy as reading a fashion blog. However, only a tiny fraction of the human population is interested in delving deep into cryptic rocket science daily. So, a significant obstacle lies before us. A block that deals with filtering user-specific content on the internet.

We cannot reverse the effects of the information explosion. However, we can alleviate the impact. We can provide users with content with which they most relate. If users are interested in nature and outdoor activities, they will likely want information about these topics. Therefore, we should filter the content out of the massive haystack of data and provide relevant information to particular users.

The content published on the internet is of multiple types. People post text content in web articles, blogs, and forums. Users also publish texts on social networking sites like Facebook and microblogging websites like Twitter. According to a report published by SocialMediaToday, Twitter stands out among a plethora of similar

websites because it is 'real-time' [5]. There are over 450 million monthly active users as of December 2022, according to the Demand Sage article [6]. Twitter is also considered an influential platform. According to a Pew Research Center study [7], in May 2021, approximately 69% of American users relied on Twitter for their daily news.

Besides the text content, multimedia such as photos, videos, and audio are posted online. Images are a better avenue for information exchange as opposed to text. According to the article by the eLearning industry [8], human brains process visuals about 60,000 times faster than text, and within nanoseconds, our brain gains information from images. With the advent of social media and internet access on portable devices like smartphones, humans' attention span is also declining. According to Microsoft's research [9], the current attention span is reduced to 8 seconds. Therefore, it makes sense to gather information from images rather than reading equivalent written articles today.

Since humans possess such small attention spans, we must maximize the information gained. Tweets are well known for their distinctive style of information exchange. As tweets are small chunks of text, they are relatively easy for humans to understand. Therefore, more is needed to solve the problem of only filtering the content relevant to the user. As discussed above, the statistics point out that visual media is more potent in painting a mental picture than simple texts. Thus, we can use these images to find out which tweets the users might be interested in.

Recommending tweets based on images is the prime focus of this project. However, we also add an extra element where users can annotate the images using tweet entities such as 'hashtags.' As many images on the internet do not have descriptive metadata associated with them, this step could be an essential medium to provide the required annotations to such images. This descriptive metadata can improve the accessibility

of the images for all people around the world.

This project aims to connect together different media forms, such as images and micro-blogs like tweets. The objective is to use the images a user likes to find relevant tweets from the pool of all Twitter content and use the entities of the recommended tweets to annotate the images. We propose a novel recommendation framework that will help us achieve the mentioned goals and open up new opportunities to enhance the framework and provide even more fascinating results.

In the next chapter, we will look at some of the existing recommendation systems and various elements essential for the proposed recommendation system. In Chapter II, we will see the literature survey, organized according to the structure described in Figure 1. The detailed methodology is described in Chapter III. Some experiments and observations are mentioned in Chapter IV. We will also see what enhancements could be made to this framework in the future in chapter V. Ultimately; we will conclude this report with chapter VI.

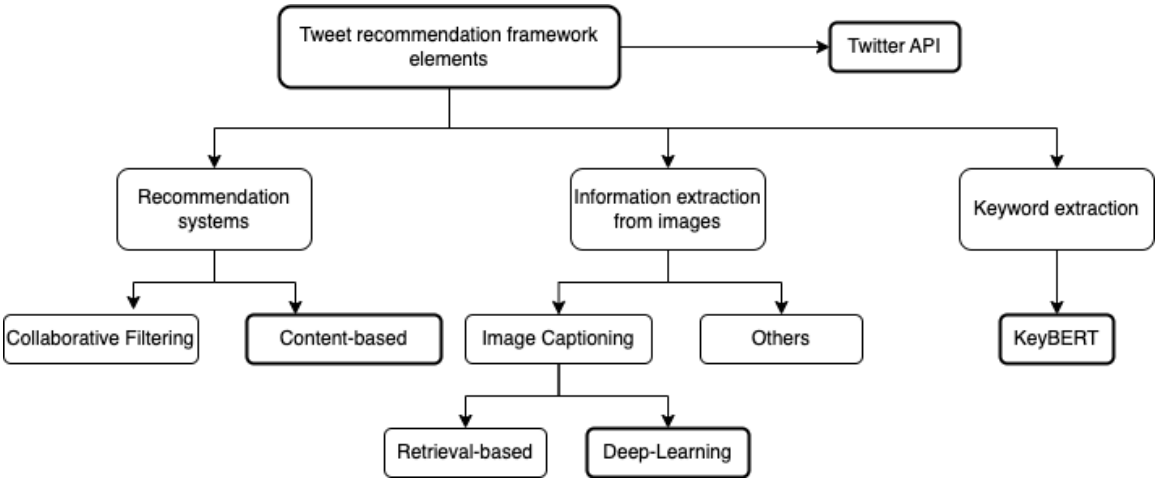


Figure 1: Organization of the literature survey

CHAPTER 2

Literature Survey

As discussed previously, the internet has burgeoned information growth. Many recommendation applications are also flourished since that point. These recommendation applications are prominent and critical in the fields such as news, social media, and politics. Filtering is an essential component in the area of news. Most predatory news outlets tend to produce both benign and malignant content on the internet, which can often promulgate fabricated information among users. Some outlets intentionally publish polarizing articles that can tilt the political preferences of particular users. Thus, filtering and building user-specific profiles are exciting research areas.

Choosing a recommendation scheme could be challenging, primarily if it deals with popular sites such as online social networks where internet traffic is always heavy. This literature survey attempts to answer the following questions:

- What type of recommendation system would suit the proposed framework?
- How would the individual components of this recommendation system look? and
- How can the components be designed to be as simple as possible and feasible?

2.1 Recommendation frameworks

After the recent breakthroughs in artificial intelligence, many studies have been published on general recommendations. Most of these studies have been adopted to build sophisticated techniques and led the groundwork for different recommendation frameworks. Adomavicius et al. [10] discretely categorize the recommendation techniques into three categories in their survey. These are collaborative filtering, content-based, and hybrid. Several factors contributed to forming these categories. These factors include recommendation algorithms, the data collection process, its background, and how users interact with the data.

2.1.1 Collaborative Filtering

Collaborative filtering techniques are famous for devising a recommendation system because they directly create user connections and match users with potentially trending items or interests [11]. E-commerce giants like Amazon.com use collaborative filtering to recommend buying items for users [12]. Sánchez-Moreno et al. [13] propose a collaborative filtering-based system for music recommendation. Apart from multimedia outlets like movies and music, Resnick et al. [14] describe a collaborative filtering framework for news recommendation.

Collaborative filtering gives promising results but has considerable demerits compared to other approaches. It suffers from a problem that Koren et al. [15] describe as a 'cold start.' In this scenario, when new users join the application, they have no interactions with other users. These interactions are built over time, thus delaying the availability of adequate data for collaborative filtering to work effectively. Additionally, collaborative filtering can suffer from the popularity bias problem, where popular items are recommended more frequently than less popular items [16].

As in this project, we directly deal with data. It makes sense to perform recommendations by leveraging the data instead of looking at the user's connections with others in a social setting. Thus, the collaborative filtering approach is not a prudent choice of framework for this project.

2.1.2 Content-based recommendation

Content-based recommendations rely on the content itself and not how the population interacts with the content. It significantly handles the disadvantages of collaborative filtering while also creating more personalized recommendations. As this approach uses a user's past interactions with the content, there are no privacy concerns for accessing other users' information. According to Lops et al. [17], content-based

recommendations would be efficient for information retrieval if we create a profile that accurately describes user preferences.

Studies in content-based recommendation have significantly enhanced the algorithms used in today's world of social media. This category of recommendation systems is effective in recommending not only the content itself but also other factors, such as user connections. Garcia and Amatriain [18] propose a technique that offers weighted content-based methods to recommend social links to users on social platforms like Twitter. Karidi et al. [19] create a content-based recommendation system for recommending tweets by building a 'concept graph' from a user's Twitter history. Content-based recommendation systems provide flexibility to include other intuitive ideas to fine-tune the recommendations.

As we have seen, multiple attempts have been made to create riveting recommendation systems. However, we have yet to see many systems connecting different content formats to generate recommendations. In other words, users might prefer the variety of content they consume to align with their interests in various outlets, such as music, videos, or photos. Thus, we devised a recommendation framework that could potentially open doors for recommendations correlating different media types.

2.2 Information extraction from images

Just as we can interpret images and videos in multiple ways, we can extract different kinds of information from visual media. For example, we can classify the objects in the images into different categories. Other ways of extracting information from images include computer vision tasks like object detection, semantic segmentation, instance segmentation, and image captioning.

As the name suggests, object detection detects the different objects in the scene and informs us about their class. In this method, we can also locate the location of the

objects in the image and draw bounding boxes around them. For example, suppose an image describes a typical vehicle traffic scenario. In that case, the detection task will produce a list of things present in the picture, and it would roughly look like {“car”, “person”, “bike”}. On the other hand, semantic segmentation classifies each pixel in an image with a category label to distinguish between different objects present within an image. Semantic segmentation takes object detection to the next level by creating an outline around the objects that accurately match the object's original boundaries. Thus, we get the object classifications and their respective ‘regions’ in the parent image.

2.2.1 Image Captioning

Object detection and semantic segmentation are intriguing computer vision problems. Exciting research is ongoing to create highly sophisticated machine-learning models to solve these problems. Both object detection and semantic segmentation provide a textual representation of the various objects with an image. However, they fail to accurately describe the relationships between different things in natural language. At this point, image captioning comes into the picture.

Image captioning is a technique of producing descriptive text by understanding the various objects within the image. This method not only mentions the classes of the objects in the image but also provides natural language adjectives and connecting words such as prepositions and conjunctions. Because of the higher descriptiveness of the captions, we use image captioning to extract textual information from the images in this project.

Image captioning is a hot research area, and a plethora of riveting breakthroughs have emerged. According to Hossain et al. [20], image captioning techniques can be categorized into retrieval-based and deep-learning-based methods.

2.2.1.1 Retrieval-based methods

In retrieval-based techniques, we have a fixed set of predefined image captions. We have to find the most similar instance to our test image, then assign the instance's caption to our test image. This technique is a lazy-learning technique analogous to nearest-neighbor classification. Some retrieval-based methods include 'k-nearest-neighbors' (kNN) [21], retrieval-augmented transformer based method [22], and ensemble-based techniques [23]. One major disadvantage of retrieval-based techniques is that they lack to provide different captions each time. Also, we must keep all the instances in memory even for generating a caption for a single image, which could be very memory inefficient.

2.2.1.2 Generative Adversarial Network-based methods

Generative Adversarial Networks (GANs) have been widely used in image captioning research due to their ability to generate images that are difficult to distinguish from natural images. GAN-based image captioning methods typically consist of two networks: a generator network that produces candidate captions and a discriminator network that evaluates the quality of the generated captions. GAN-based image captioning methods have been shown to generate more diverse and creative captions than other methods, such as encoder-decoder models. Additionally, GAN-based image captioning methods are robust to noise and other variations in the input image [24].

Recent research has explored using GANs in conjunction with reinforcement learning to improve the quality of generated captions further. Furthermore, GAN-based image captioning methods have shown potential for application in various domains, such as medicine and fashion, where accurate and detailed descriptions of images are crucial for diagnosis and recommendation systems.

GANs have shown promising results in terms of image captioning. However, they

have some associated caveats. GANs are more challenging to train and maintain than any other captioning models. For instance, compared to deep-learning-based methods, GANs are more complicated to train because they require training two neural networks separately, and the training process can be unstable. Because of this reason, we decided to avoid using GANs for image caption generation.

2.2.1.3 Deep-learning-based methods

Deep-learning-based methods aim to generate descriptive and coherent textual descriptions of images by leveraging the power of deep neural networks. Under the umbrella of the deep-learning-based image captioning methods, we have six major subcategories. These subcategories are decided based on [20]:

- the type of learning strategy used
- the architectures
- type of feature mapping
- type of captions to be generated, and
- miscellaneous methods which contain techniques such as an ensemble of deep-learning models

We will primarily focus on the learning strategy and the architecture of the deep-learning models.

Based on the learning strategy, we have image caption generators that use supervised learning. As we know, supervised learning techniques require the training dataset to contain both instances and their corresponding labels (in the case of image captioning, the ground-truth captions replace the labels). Various groundbreaking image caption generators have emerged that use the supervised learning approach. Chen and Zitnick [25] propose an image captioning technique where they achieved a human agreement score of 21% for the generated captions. Their proposed work

produced significantly superior results than the primitive techniques. A 'region-based' image caption generation architecture, proposed by Karpathy and Fei-Fei [26], creates an understanding of the scene presented in the input image describes different regions present in the scene independently. Looking at the success of supervised deep-learning techniques, we used the same learning approach to train our image captioning model in this project.

Deep-learning-based caption generators can also be categorized into different classes by considering their architecture. The most commonly used architectures by caption generators are 'encoder-decoder' architecture and compositional architecture. As the name suggests, the encoder-decoder architecture involves a component that encodes the input data to a set of features. The decoder component operates on the encoded features to 'decode' the output. On the other hand, in compositional architecture, we have a composition of multiple independently functioning blocks. Because of the multiple components, the compositional architectures tend to produce complex models, which could be troublesome for building, training, and debugging. Therefore, in this project, we tilted towards using the encoder-decoder architecture as they can be reasonably straightforward for implementations.

Vinyals et al. [27], proposed the ''Show and Tell'' method involving a supervised learning approach and encoder-decoder architecture. This technique used a sophisticated deep convolutional neural network (CNN) to encode the input images into their corresponding features. These features are passed to a recurrent neural network (RNN). They used a particular RNN consisting of multiple long short-term memory (LSTM) units as their decoder component to generate the captions. Training images are provided to the CNN, whereas their corresponding captions are provided to the RNN. They surpassed the scores of the existing methods to achieve state-of-the-art performance having Bilingual Language Understudy (BLEU) [28] scores of 30.3%.

Because of the architecture's simplicity, we implemented an image caption generator derived from the 'Show and Tell' approach for this project.

2.2.2 Keyword extraction

Once we train the image captioning model, we will have a set of weights and the model structure to hold the weights for prediction. After training the model, we pass the test images and generate their captions. We have now captured the essence of the images in their captions. However, for tweet recommendations, we have to process these captions further. Twitter has a rich API to fetch tweets based on topics, hashtags, and trends. In order to leverage the API's features, we have to find keywords from the generated captions. After getting the keywords, we can manipulate the Twitter search queries accordingly and fetch the tweets.

Identifying keywords from text is a challenging task in natural language processing. Keyword recognition depends on the size of the text and its complexity. If we process huge text, there is a possibility of getting different keywords that are not necessarily related. However, we can get semantically related keywords if we process a text caption typically 30-40 words long.

Numerous proposed techniques process text and give us the keywords with a particular confidence score. These techniques commonly use unsupervised learning as their supervised counterparts need vast data of manually chosen keywords. Unsupervised techniques for keyword extraction are not labor intensive, making them highly efficient to train and test.

Mihalcea and Tarau [29] propose a technique called 'TextRank' to extract keywords where they simulate a graph by treating the words as the vertices and the edges between the vertices are formed if they co-occur in a sliding window. After constructing the graph, they apply the 'PageRank algorithm' [30] until convergence to rank

the words and yield potential keywords. Although straightforward, this technique can produce undesirable results, such as inadvertently extracting a non-keyword. Several ranking-based methods such as Position Rank [31] and Word Attraction Rank [32] improve upon the TextRank algorithm, but they may still produce undesirable output.

Rose et al. [33], proposed a rapid automatic keyword extraction (RAKE) method for extracting keywords. This technique is language-independent and based on the fact that words with no lexical meaning (stopwords) rarely occur together with potential keywords. This technique also uses the sliding window strategy similar to TextRank. In a technique called 'YAKE!' proposed by Campos et al. [34], they went one step ahead of RAKE and used a language-independent list of stopwords and statistical features to rank the most important keywords. Both RAKE and YAKE! are popular methods but can be challenging to use.

KeyBERT is a modern, lightweight, easy-to-use keyword extraction tool developed by Grootendorst [35] that utilizes the BERT document embeddings [36] and key-phrases that are most similar to a document. This tool is a widespread keyword extraction as it is easy to install, its rich API is free, and all the pre-trained intermediate components the tool requires are readily available on the internet. Therefore, we considered using KeyBERT keyword extraction for the project over the other mentioned frameworks.

This literature review has given an assessment of the state of research in the areas of recommendation systems, image captioning methods, and keyword extraction methods. The advantages and constraints of various keyword extraction strategies as well as the strengths and shortcomings of various recommendation systems and image captioning methodologies have all been determined through a thorough comparison of numerous studies. Based on this literature survey, we determined the best possible

strategy to apply for this project. In the coming chapter, we will take a look at the methodology used for the proposed tweet recommendation framework.

CHAPTER 3

Methodology

This chapter will delve into the methodology employed in developing the proposed architecture. We will discuss the steps taken to arrive at the final design of the architecture and the tools and techniques used.

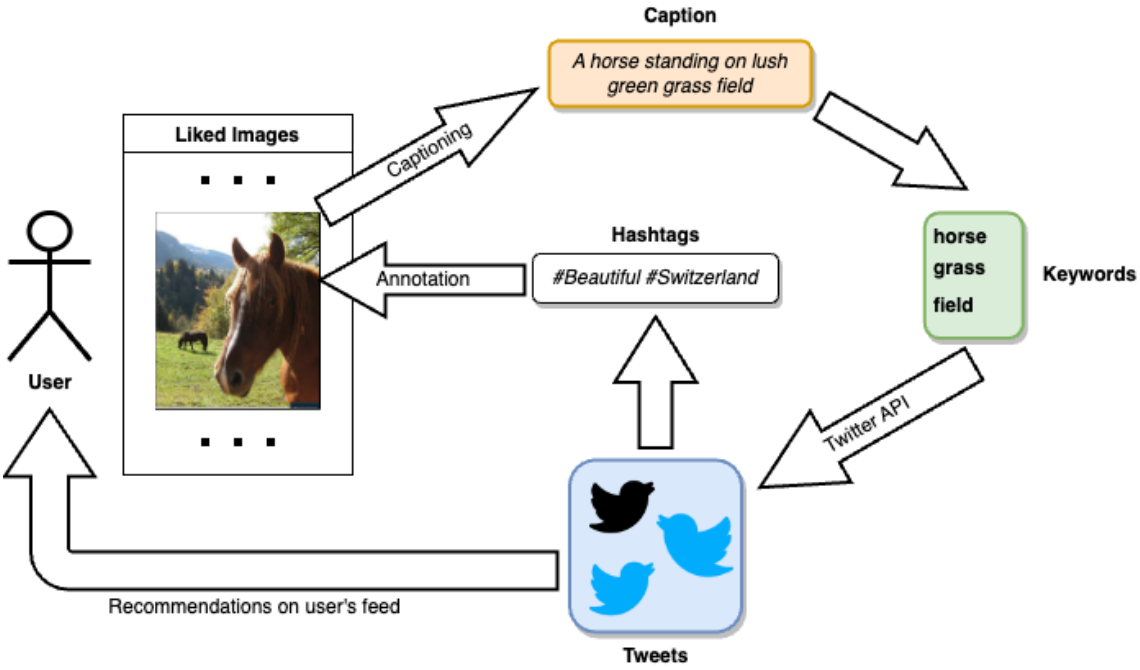


Figure 2: Tweet recommendation framework

As described in Figure 2, the recommendation system is broken into three major components. The components are an image caption generator, keyword extraction, and tweet fetcher. The figure describes the components with the colors orange, green, and blue, respectively. Now, let us delve deeper into the functioning of each component and how they are implemented for the project, in the upcoming sections.

3.1 Image Caption Generator

In this project, we implemented a deep-learning image caption generator. We specifically chose deep learning because of its feasibility. It is also reasonably straight-

forward compared to the Generative Adversarial Network (GAN) caption generators and is memory efficient compared to the instance-based caption generators.

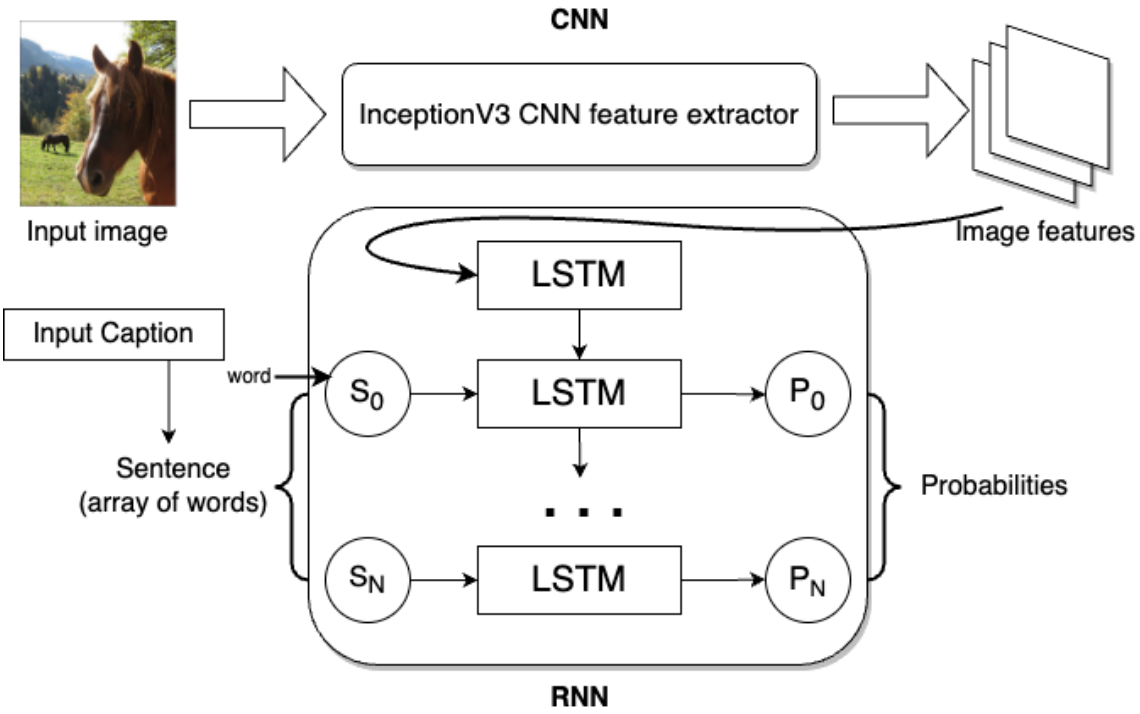


Figure 3: Image captioning architecture

Numerous deep-learning caption generation architectures are available, and we selected the ‘‘Show and Tell’’ architecture for this project [27]. This architecture has two main components, as described in Figure 3: a CNN for extracting the features from the images and reducing the dimensionality and an RNN for language modeling. The RNN component is responsible for generating the captions. The image captioning model based on this architecture is trained end-to-end on a large dataset of images and their corresponding captions. In this project, we chose the MS-COCO dataset [37] for training the caption generator. This dataset contains over 300 thousand images, each with five associated captions. However, for simplicity and computational feasibility, we trained the captioning model on a uniformly sampled subset of 25 thousand images

in this project.

The CNN component of the architecture extracts features from the fed images. Instead of training the CNN component from scratch, we used the pre-trained InceptionV3 model (trained on ImageNet dataset), freely available on the internet. The InceptionV3 model is a deep CNN architecture developed, particularly for image recognition tasks. It is characterized by its use of inception modules, composed of parallel convolutional layers with different kernel sizes and pooling operations. The architecture is designed to balance the trade-off between depth and computational efficiency, allowing for accurate image recognition with a relatively small number of parameters [38]. Because the feature extraction segment of any computer vision task is comparable, we can utilize the excellent image feature extraction of InceptionV3, which is trained by giving paramount importance to accuracy. As more accuracy of image classification implies more accuracy of feature extraction, we can efficiently extract the features and use them for any other possible image processing application, in our case: image captioning.

The RNN component of the architecture is responsible for generating the natural language description of the image. Specifically, the RNN takes the image features as input and generates a sequence of words that form the caption. The RNN is trained using 'teacher forcing,' where the ground-truth caption is fed as input at each time step during training [39]. For this project, we chose long short-term memory (LSTM) as our RNN. LSTMs retain dependencies in the data for a long term which is helpful in efficiently learning the meaningful dependencies between the words of the captions used for training.

In the following sections, we will take a closer look at the implementation of the image captioning program we implemented as a part of this project. We will first see the pre-processing step, followed by the training, and finally, caption generation.

3.1.1 Pre-processing

As we have seen earlier, the method for image captioning we are trying to adapt and implement is supervised-learning-based. Therefore, we need a data source for training and validation. For this purpose, we chose the MS-COCO dataset containing a large number of images and their predefined captions. However, as the dataset is large, we decided to sample it to make it feasible for implementation. We first sampled $25,000$ (without replacement) images and their corresponding captions for the training set. We again sampled $5,000$ instances from the raw data source for validation, independent of the training set.

Table 1: MS-COCO dataset outline

	File	**Contents**
annotations	**instances**	Instance IDs and category IDs
	captions	Instance IDs mapped to captions
images	train	Training images
	val	Validation images
	test	Test images

After sampling the dataset, we cleaned the data. The text captions contain irregularities such as disproportionate white spaces, improper punctuation, and random capitalization. Therefore, we cleaned the captions and made them uniform by trimming the white spaces and converting each word to lowercase. After cleaning the captions, we stored them in a new file.

For feature extraction, we are using the pre-trained InceptionV3 model. This model requires the images to be of a specific shape before input. For InceptionV3, the shape the images need to be in is $(299, 299, 3)$. Therefore, we created a generator object for training that automatically resizes the images in the target size with proper interpolation to preserve the image quality.

The RNN model generates the captions word by word. For the RNN model, we need unique starting and ending words for each caption. We use the words `startseq` and `endseq` and place them respectively at the start and end of each caption. These words signal the LSTM when to start and stop the caption generation. In other words, if we generate the word '`endseq`,' we will stop caption generation. We must do this pre-processing step. Otherwise, the LSTM may generate undesirably extended captions, which would make no sense.

3.1.2 Training and Validation

Once we complete all the pre-processing steps, we move toward the training part. As discussed before, 'Show and Tell' consists of two significant elements: CNN and RNN. Firstly, we downloaded the InceptionV3 CNN weights. Deep-learning libraries provide the skeleton of the model with their API. For this project, we used TensorFlow and Keras as our deep-learning libraries. After downloading the weights, we loaded them into the model imported using TensorFlow. Then, we ran all the training images and extracted their features. As described in the following Figure 4, the CNN's output is a linear vector of length 2048. Therefore, for each image, we get a one-dimensional vector of length 2048 as its corresponding feature vector. We saved the features for the training images on a binary pickle file to prevent re-computation every time the model is trained. Similarly, we performed the above steps for the validation image set.

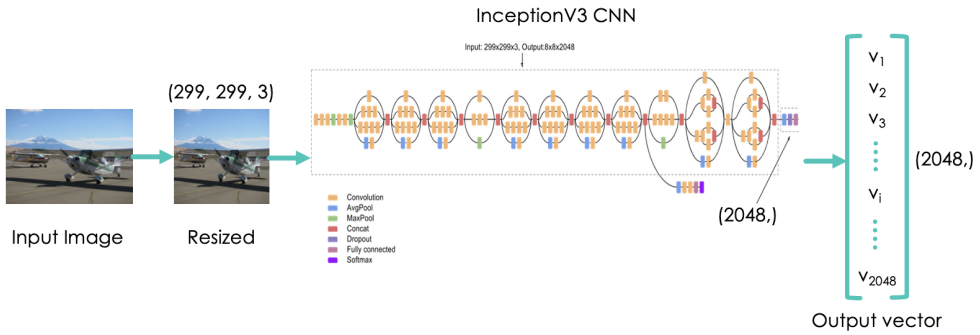


Figure 4: InceptionV3 input and output dimension details

The RNN model needs a fixed length of input caption sentence. Therefore, we ran through all the available captions and chose the caption length with the most number of words as the fixed input length. This maximum length will accommodate all the other captions too.

RNNs cannot directly process the words. We need to create their embedding vectors first. To make the embeddings, we need the size of the vocabulary and the size of the embeddings. TensorFlow-Keras provide this feature to create a layer in the RNN model for the embedding creation.

Then we add the LSTM layer to our model. After this step, we concatenate our CNN model with the RNN model, so the output of the CNN model goes into the RNN model. We also need a classifier layer at the end to get the word prediction. We use the 'softmax' activation function and categorical cross-entropy loss function for training. After compiling the model, we used the Adam optimizer to minimize the training loss.

We trained the image captioning model for eleven epochs and recorded the loss values after each epoch. We will gloss over the training scenario in Chapter IV.

3.1.3 Caption Generation

After training, we use the trained models in 'forward' mode. In other words, we only perform prediction and no backpropagation. First, we extract features of the test images and feed them to our caption generation module. The caption generation module uses the trained RNN to produce the captions word by word.

There are two methods of caption generation. The first method is straightforward and iterative. In this method, we iterate over a range of integers starting from 0 until the length of the wordiest caption. In each iteration, we use the RNN to predict the next word in the sequence. We accumulate all the words in a string. If the model

generates the word '`endseq`,' we break the loop. The accumulated words will be the generated caption.

The second method uses 'beam search.' This method has a beam parameter, 'k.' We consider the best 'k' sentences up to time 't' (the outer loop iterator). We will maintain a set of k best-generated sentences until the end of the outer loop (runs from 1 to the length of the wordiest caption). Beam search is better than the first method, giving us choices of different captions. We can choose the best from the 'k' chosen captions as our final output.

3.2 Keyword Extraction

This step was relatively simple to implement as we use a very straightforward and public tool for keyword extraction. We installed the 'KeyBERT' package using the `pip` functionality. The package includes all the necessary binaries and code for keyword extraction. All we need to do is pass the captions as input to the functions, and the library generates the desired results.

To enhance the tweet recommendation experience, we also implemented a 'word2vec' model [40]. A word2vec model creates 'word embeddings,' which are numerical representations of the words. These word embeddings are vectors in a multi-dimensional space (the vector space's dimensionality depends on the word vectors' intended length). Vectors can be very helpful in determining similar words in the vector space using different similarity measures like cosine similarity and Euclidean distance. The cosine similarity between a simple mean of the projection weight vector of the input word (our input keyword) and each word vector in the model can be calculated using word2vec. This similarity score will yield the top 'k' most similar words. Thus, using the word2vec model, we can augment the set of keywords by adding similar keywords output by word2vec. This augmentation can be beneficial to

refine the tweet fetching queries.

We trained the word2vec model on a massive corpus of Wikipedia data. The raw data contains over five million documents in English. We first parsed all the documents to collect all the unique words and only retained over 200,000 unique words. We used the Gensim library [41] to train the word2vec model. The model produces the most similar words if we pass a keyword.

3.3 Tweet Recommendation and Image Annotation

We collected all the necessary ingredients for the tweet recommendation system in the previous sections. In this section, we will discuss how actually to recommend tweets. This project component is the culmination of the pipeline of different components, hence the most important.

We first require Twitter API key access to fetch the tweets. Twitter has a collaborative developer community that provides general public and academic researchers access to developer tools to integrate Twitter into various applications. We first acquired academic access for the project, where we unlocked the full potential of the Twitter API (elevated access), including functions to read and write on Twitter in real time. We can then plug these keys into an open-source API handling Python library 'tweepy' available for free (For usage information, check Appendix). This library contains wrapper functions that hit the Twitter API endpoints and show results to the clients.

Twitter API needs specific input to produce a particular output. In our case, the input will be a specially formulated search query. As discussed earlier, we can formulate the search query based on different properties of keywords. For example, we can consider the 'importance' of the keywords as some keyword extraction tools return confidence scores associated with the keywords. In this project, we chose a

straightforward way of formulating the search query. We randomly selected a few keywords from the extracted keywords and joined them using the boolean operators 'AND' and 'OR' also selected randomly.

After formulating the query, we pass it into the '`search_tweets`' API along with other parameters like the number of results needed. We will receive a 'cursor' to iterate to access all the fetched tweets and their properties. Apart from the query formulation, we can also use the keywords as input hashtags to the query to get the tweets with the keywords as hashtags. The keywords can also be thought of as specific trends on Twitter. For example, if we have the keyword 'horse' and horses are trending on Twitter, we can show the trending tweets to the user on a separate interface.

When the recommended tweets include hashtags, they serve as valuable information to share with users. By presenting these hashtags to the users, they are provided with an opportunity to utilize them for annotating the images. These annotations, created by users based on the relevant hashtags found in the recommended tweets, can significantly enhance the overall accessibility of the images.

The annotations created by users play a crucial role in making the images more understandable and informative. By associating specific hashtags with the content depicted in the images, users contribute to a more comprehensive description and context. This benefits not only the individuals directly interacting with the images but also other people who may later encounter the annotated images. The annotations provide additional insights and allow for a broader range of interpretations, enabling a more inclusive experience for all users.

Furthermore, these annotations created based on the hashtags found in recommended tweets promote collaboration and crowd-sourcing of information. By sharing their knowledge and understanding through annotations, users collectively contribute

to a communal effort to improve the accessibility and interpretation of images.

In this way, we implemented the building blocks of the proposed recommendation framework. Now, let us take a look at the different experimental results and observation in the coming chapter.

CHAPTER 4

Experiments and Observations

This project report's experiments and observation section will detail the methodology used to evaluate the effectiveness of the novel tweet recommendation framework. This section provides an overview of the experiments conducted to validate the proposed framework's efficacy, including the evaluation metrics of the various framework components.

Overall, the experiments and observation section will provide valuable insights into the proposed tweet recommendation framework's performance and highlight improvement areas. While we do not have any quantitative metrics or results for some elements of this project to present at this stage, the user study and qualitative analysis will provide valuable feedback that can be used to refine the recommendation system and improve its performance in future iterations.

Let us look at the experiments and evaluation of different components of the framework in the following sections.

4.1 Image Caption Generator Results

The training process for the image caption generator of the project was conducted on a high-performance computer provided, whose access was provided by the SJSU CS department. The training though conducted on a relatively smaller subset of the whole dataset, was time intensive. We trained the model for 15 epochs with multiple callbacks, recording the checkpoints and logs and checking whether to stop early. The model was effectively trained for eleven epochs before stopping automatically, as no further improvement was seen. Figures 5 and 6 demonstrate the training and validation parameters curves over the number of epochs recorded using the open-source logging toolkit called TensorBoard provided by TensorFlow.

After training, the model was evaluated over a test set consisting of 1000 images.

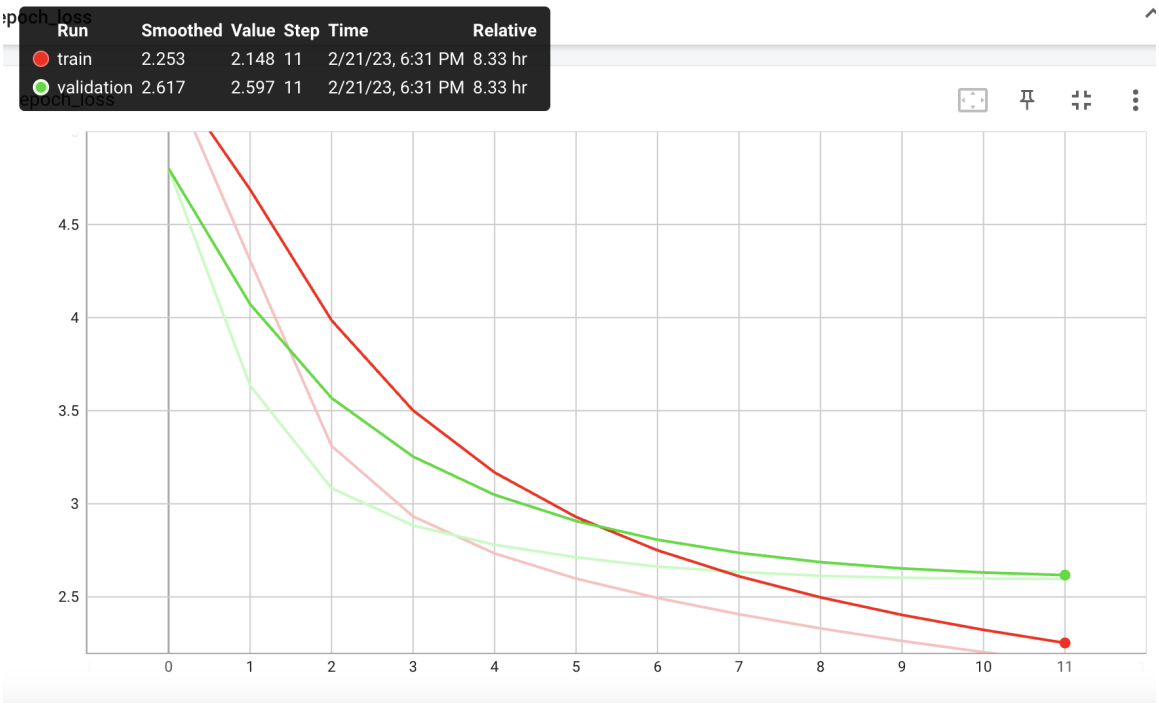


Figure 5: The loss values versus number of epochs

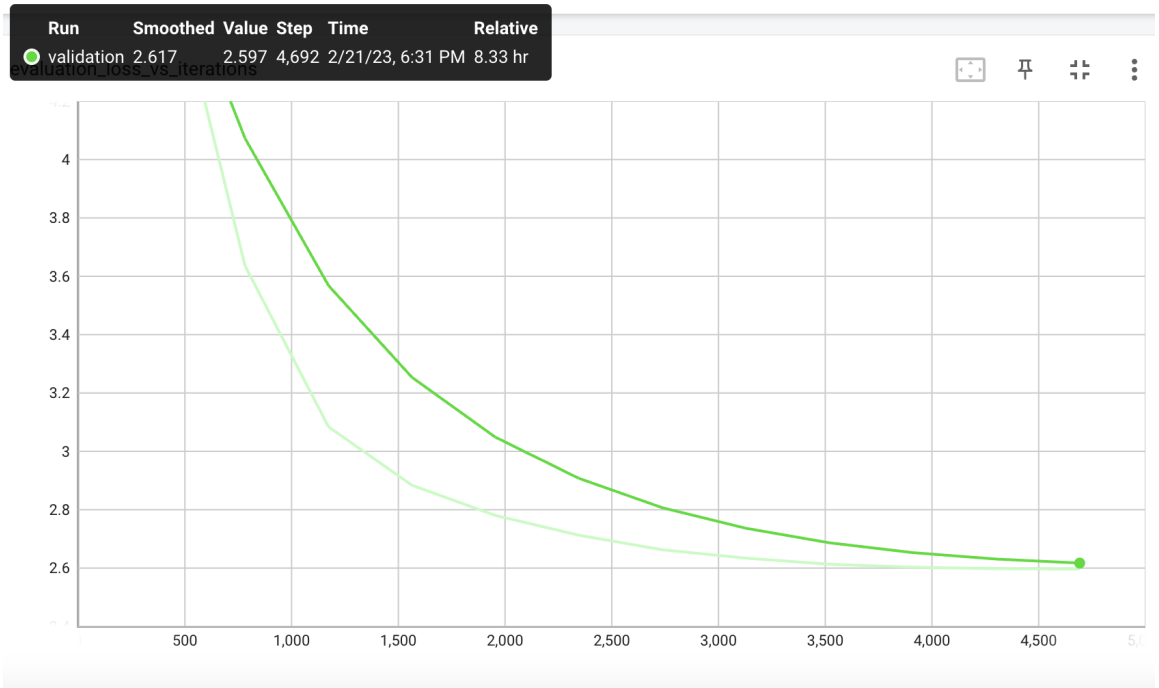


Figure 6: The curve of validation loss over the number of iterations

These images had strictly no influence on the training or validation process. After evaluation, we recorded the BLEU scores of the generated captions. The BLEU scores are computed based on the ground truth captions of those 1000 images. Table 2 summarizes the results obtained for the evaluation. In Table 3, we have the numbers showing how our current model performs compared to the various image caption generators. From Table 3, we can infer that the performance of our model is comparable to these different models.

Table 2: BLEU scores of the caption generator

BLEU-Score type	Weights	Score
BLEU-1-gram	$(1.0, 0, 0, 0)$	0.664
BLEU-2-gram	$(0.5, 0.5, 0, 0)$	0.434
BLEU-3-gram	$(0.3, 0.3, 0.3, 0)$	0.324
BLEU-4-gram	$(0.25, 0.25, 0.25, 0.25)$	0.192

Table 3: Comparison of image captioning performance on MS-COCO dataset of various models.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
M-RNN [42]	0.67	0.49	0.35	0.29
V-S.M [43]	0.625	0.45	0.321	0.23
Show and Tell (Current model)	0.664	0.434	0.324	0.192

The result of passing a test image to the trained image captioning model is shown in Figure 7. In the test image, we can see a horse standing on a green meadow, and the trained captioning model gave us a pretty accurate representation of the image.

4.2 Keyword Extraction Results

Figure 8 shows an example of the keyword extraction program on a sample caption. The Figure shows that five significant keywords and their confidence score

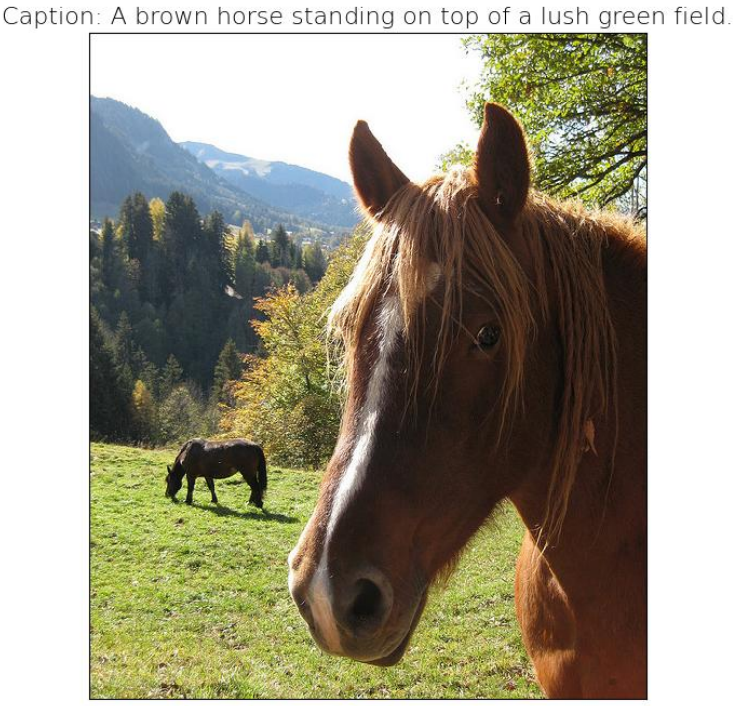


Figure 7: Image captioning result on a test image

values have been extracted. We can select any of these keywords or even limit our keyword extraction to produce only top k keywords.

```
Caption:
        A brown horse is standing on top of a green grass field.

[('horse', 0.5308), ('standing', 0.362), ('field', 0.3572), ('brown', 0.3168), ('grass', 0.293)]
(image_captioning) [skadam@spartan01 ImageCaptioning]$
```

Figure 8: The result of KeyBERT keyword extraction

The following Figure shows the output of the word2vec model that produces similar keywords for each input keyword.

```
In [1]:
1 import os
2 from rich import print

In [2]:
1 from gensim.models import Word2Vec

In [4]:
1 model = Word2Vec.load(os.path.join(os.getcwd(), "wiki.word2vec.model"))

In [5]:
1 keywords = ['horse', 'standing', 'field', 'brown', 'grass']

In [7]:
1 for keyword in keywords:
2     sims = model.wv.most_similar(keyword, topn=3)
3     similar_words = [t[0] for t in sims]
4     print(f"{keyword} -> {','.join(similar_words)}")

horse -> horses,riderless,standardbreds

standing -> sitting,seated,stands

field -> fields,preglabellar,nishigaoka

brown -> gray,smith,harris

grass -> grasses,mown,reedbeds
```

Figure 9: Sample output of word2vec model giving words similar to the input keywords

```
(sjsu_project) shashwatkadam@Shashwats-MacBook-Pro TweetFetch % /Users/shashwatkadam/miniforge3/envs/sjsu_project/bin/p
ython /Users/shashwatkadam/Documents/SJSU_MSCS/Spring2023/CS298/Project/TweetFetch/src/fetch_tweets.py
Query: horse OR grass

1) RT @jenny_sandbank: People protest and are up in arms about a horse race but not about the cruel and barbaric
practices of Halal?
Why?

2) Few hours in garden, got a bit more grass off and got some veg in the ground. Will see if anything grows 😂

3) RT @Gabriele_Corno: His majesty, the platinum white horse, takes a bath https://t.co/xYEZl10Vod

4) @owenclark3 @CatDisapproval .....can't have that!...(seriously, that grass has to be an enormous amount of work to
maintain)..

5) RT @Britain_People: MATT HANCOCK:  Horse Racing Connection

💥Matt Hancock GIVEN £350,000 from people linked to Cheltenham Festival

👉https…

6) @killrockhardt the grass guardian if you will

7) RT @domdyer70: The Grand National is not as the Daily Mail claims “The peoples race” to vast majority of caring
compassionate people in U.K…

8) RT @sarcitwat: Think this is the horse I put my money on in #GrandNational #Aintree https://t.co/R4cAcM4oLQ

9) RT @domdyer70: The Grand National is not as the Daily Mail claims “The peoples race” to vast majority of caring
compassionate people in U.K…

10) your horse is beautiful. https://t.co/itA8oL709C

(sjsu_project) shashwatkadam@Shashwats-MacBook-Pro TweetFetch %
```

Figure 10: Output of tweet recommendation

4.3 Tweet Recommendation Results

After running the tweet recommendation code on a sample query, we obtained the following results. In Figure 10, we can see the ten different tweets that either involve the keyword ‘horse’ or ‘grass.’ The tweet search query we use to fetch the

tweets is created by randomly picking the keywords from the extracted keywords. The operator (disjunction or conjunction) is also chosen randomly. However, as the query formulation is customizable, people can modify it themselves.

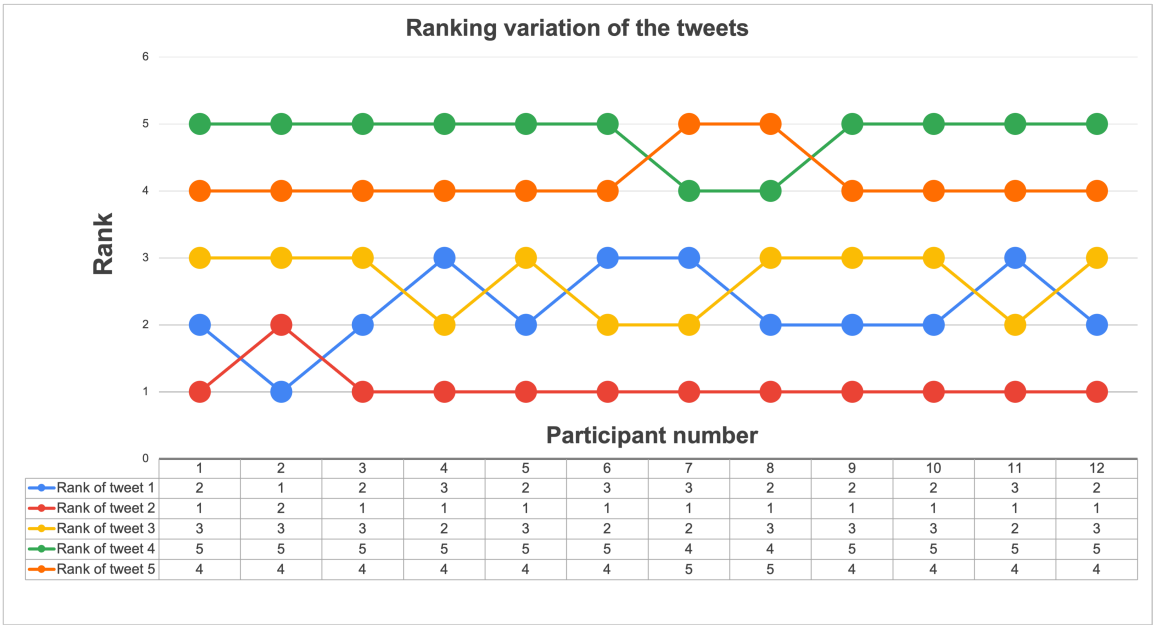
We also conducted a short real-user feedback experiment where we asked users to rank the tweets obtained from the system by processing an input image (Figure 11). The tweet recommendation system returned five tweets, as shown in Figure 12. To gather the user feedback and responses, we created a short Google form where we presented the users with the input image and the tweets retrieved by the system. The users ranked the tweets based on their opinion of the tweets and their relevance to the input image. The results are shown in Figure 13. According to Figure 13, most people found the second tweet most relevant to the image, while tweets 4 and 5 took the bottom two spots in the ranking.

Apart from the ranking, we also collected user feedback for each recommended tweet. Users rated the tweets on a scale of 0 to 10 (0 represents that the fetched tweet does not resemble the input image, and 10 represents that the fetched tweet is the most relevant tweet for the input image). The feedback is tabulated in Table 4, where users rated the tweets above 7, proving that they found the tweets relevant to the input image.

Figure 11: Input image for tweet recommendation user survey

```
(sjsu_project) shashwatkadam@Shashwats-MacBook-Pro ImageCaptioning % /Users/shashwatkadam/miniforge3/envs/sjsu_project/
bin/python /Users/shashwatkadam/Documents/SJSU_MSCS/Spring2023/CS298/Project/ImageCaptioningFromHPC/ImageCaptioning/src
/tweets.py
Query: broccoli AND vegetables
1) Nuts, seeds, legumes, algae and also many vegetables, such as broccoli or spinach, have a high calcium content! 😇

👉 Get Your Calcium Ethically: Pledge to Go Dairy-Free Today : https://drove.com/.2Cff

#ditchdairy #eatplants #dairyisscary #dairyfree #calcium #vegan
2) Get that potassium (K) from cruciferous vegetables (broccoli,cabbage and spinach) and also Avocado.

Bananas are SUGAR!
DROP THEM!

3) Did you know that a cancer-fighting enzyme is produced when you chew cruciferous vegetables like broccoli?

4) Zucchini soup.
Steak.
Rice.
Sausage with vegetables.
Kale and broccoli salad.
Tuna and chicken salads.
Done and ✔

#Weeklymealprep

5)
We need an equalizer, they're making all these fake meats out of vegetables, how about we make broccoli out of BACON
🥓.
```

Figure 12: Output of the implemented recommendation system for the input image for the user survey.



	1	2	3	4	5	6	7	8	9	10	11	12
Rank of tweet 1	2	1	2	3	2	3	3	2	2	2	3	2
Rank of tweet 2	1	2	1	1	1	1	1	1	1	1	1	1
Rank of tweet 3	3	3	3	2	3	2	2	3	3	3	2	3
Rank of tweet 4	5	5	5	5	5	5	4	4	5	5	5	5
Rank of tweet 5	4	4	4	4	4	4	5	5	4	4	4	4

Figure 13: Ranking of the tweets based on the user survey responses.

Table 4: Real user feedback on the fetched tweets

Participant #	tweet 1	tweet 2	tweet 3	tweet 4	tweet 5
1	9	10	9	7	8
2	10	9	9	7	8
3	9	10	8	7	8
4	8	10	9	7	8
5	10	10	9	8	8
6	7	10	9	7	8
7	8	10	9	7	7
8	9	10	8	7	7
9	9	10	8	7	7
10	9	10	8	7	7
11	8	10	8	7	8
12	9	10	8	7	7

CHAPTER 5

Future Works

The tweet recommendation framework we proposed in this project is novel. However, there are multiple possibilities in which this framework can be expanded and augmented. In this section, we will check out ways to enhance the framework in the future.

Currently, the framework utilizes only image data that we have with us. In future works, we can connect this system with famous social networks such as Instagram and Facebook. Through this connection, we can make a unified interface for the users to view all their content in one place. Also, by adding other social networks, we can leverage the social ties a user has with other users to improve the recommendations.

The current scenario needs to incorporate user feedback properly. For instance, if we can choose which recommended tweets are good or bad, we can use the feedback to tune the recommendation system to provide better results in the future. In this situation, we can apply reinforcement learning techniques leveraging user feedback to improve the recommendation system.

The proposed framework currently does not maintain any user profile. However, we can periodically build a user's profile in the future, storing the keywords extracted from their liked images. For example, if users like more images of nature, they should receive more tweets relevant to nature. Thus, we can have a weight associated with every keyword, and depending on that weight, we can tweak the recommendations.

As we only considered a content-based recommendation in this proposed framework, there is a possibility to create a 'hybrid' recommendation system that also incorporates elements of collaborative filtering. Using collaborative filtering, we can model user interrelations while preserving the original essence of the proposed framework, i.e., using images as the source to generate tweet recommendations.

CHAPTER 6

Conclusion

In conclusion, the novel tweet recommendation system we developed in this project has the potential to enhance the user experience on social media platforms significantly. By leveraging state-of-the-art machine learning techniques, we created a system recommending relevant, informative, and engaging tweets to the user. Our approach involves analyzing a user's image preferences on a platform, including the content with which they have engaged. This information is then used to generate personalized recommendations tailored to the user's interests and preferences.

Overall, this system has significant commercial potential for social media companies looking to improve engagement and retention rates. By providing users with personalized and relevant content, we can increase the amount of time they spend on the platform and foster a sense of community and connection among users. We believe that with further refinement and testing, this system could become an essential feature for actual social media platforms. Additionally, user involvement in the proposed recommendation system adds to the personalization factor of the system. The user feedback could be essential for further enhancements of the system.

LIST OF REFERENCES

[1] A. Petrosyan, ‘‘Number of internet users worldwide from 2005 to 2022,’’ Feb 2023, (Accessed Feb. 3, 2023). [Online]. Available: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/

[2] D. Sunil, ‘‘The information explosion: Trends in technology 2011 review,’’ *The Journal of Government Financial Management*, vol. 60, no. 4, p. 46, 2011.

[3] S. Kemp, ‘‘Digital 2021: Global overview report - datareportal – global digital insights,’’ Oct 2021, (Accessed Feb. 1, 2023). [Online]. Available: https://datareportal.com/reports/digital-2021-global-overview-report

[4] L. Andre, ‘‘53 important statistics about how much data is created every day,’’ Mar 2023, (Accessed Mar. 6, 2023). [Online]. Available: https://financesonline.com/how-much-data-is-created-every-day/

[5] A. Hutchinson, ‘‘Here’s why twitter is so important, to everyone,’’ Mar 2016, (Accessed Feb. 3, 2023). [Online]. Available: https://www.socialmediatoday.com/social-networks/heres-why-twitter-so-important-everyone

[6] D. Ruby. ‘‘58+ twitter statistics for marketers in 2023 (users amp; trends).’’ (Accessed Feb. 3, 2023). [Online]. Available: https://www.demandsage.com/twitter-statistics/

[7] A. Mitchell, E. Shearer, and G. Stocking, ‘‘News on twitter: Consumed by most users and trusted by many,’’ Apr 2022, (Accessed Feb. 3, 2023). [Online]. Available: https://www.pewresearch.org/journalism/2021/11/15/news-on-twitter-consumed-by-most-users-and-trusted-by-many/

[8] D. Jandhyala, ‘‘Visual learning: 6 reasons why visuals are the most powerful aspect of elearning,’’ May 2021, (Accessed Feb. 3, 2023). [Online]. Available: https://elearningindustry.com/visual-learning-6-reasons-visuals-powerful-aspect-elearning

[9] L. Stutsman, ‘‘You have eight seconds. differentiate your business through the art of storytelling.’’ Nov 2021, (Accessed Feb. 3, 2023). [Online]. Available: https://www.microsoft.com/en-us/us-partner-blog/2021/11/15/you-have-eight-seconds-differentiate-your-business-through-the-art-of-storytelling/

[10] G. Adomavicius and A. Tuzhilin, ‘‘Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,’’ *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734--749, 2005.

[11] Y. Shi, M. Larson, and A. Hanjalic, ‘‘Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,’’ *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1--45, 2014.

[12] G. Linden, B. Smith, and J. York, ‘‘Amazon.com recommendations: item-to-item collaborative filtering,’’ *IEEE Internet Computing*, vol. 7, no. 1, pp. 76--80, 2003.

[13] D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. L. Batista, and M. N. M. García, ‘‘A collaborative filtering method for music recommendation using playing coefficients for artists and users,’’ *Expert Systems with Applications*, vol. 66, pp. 234--244, 2016.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, ‘‘Grouplens: An open architecture for collaborative filtering of netnews,’’ in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work*, ser. CSCW ’94. New York, NY, USA: Association for Computing Machinery, 1994, p. 175–186. [Online]. Available: https://doi.org/10.1145/192844.192905

[15] Y. Koren, S. Rendle, and R. Bell, ‘‘Advances in collaborative filtering,’’ *Recommender Systems Handbook*, pp. 91--142, 2021.

[16] X. Zhao, Z. Niu, and W. Chen, ‘‘Opinion-based collaborative filtering to solve popularity bias in recommender systems,’’ in *Database and Expert Systems Applications: 24th Int. Conf., DEXA 2013, Prague, Czech Republic, August 26-29, 2013. Proc., Part II 24*. Springer, 2013, pp. 426--433.

[17] P. Lops, M. De Gemmis, and G. Semeraro, ‘‘Content-based recommender systems: State of the art and trends,’’ *Recommender Systems Handbook*, pp. 73--105, 2011.

[18] R. Garcia-Gavilanes and X. Amatriain, ‘‘Weighted content based methods for recommending connections in online social networks.’’ Association for Computing Machinery, 2010, pp. 68--71.

[19] D. P. Karidi, Y. Stavrakas, and Y. Vassiliou, ‘‘A personalized tweet recommendation approach based on concept graphs,’’ in *2016 Int. IEEE Conf. Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 2016, pp. 253--260.

[20] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, ‘‘A comprehensive survey of deep learning for image captioning,’’ *ACM Comput. Surv.*, vol. 51, no. 6, feb 2019. [Online]. Available: https://doi.org/10.1145/3295748

[21] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. Zitnick, ''Exploring nearest neighbor approaches for image captioning,'' *arXiv preprint arXiv:1505.04467*, May 2015.

[22] S. Sarto, M. Cornia, L. Baraldi, and R. Cucchiara, ''Retrieval-augmented transformer for image captioning,'' in *Proc. 19th Int. Conf. Content-based Multimedia Indexing*, 2022, pp. 1--7.

[23] M. Yang *et al.*, ''An ensemble of generation- and retrieval-based image captioning with dual generator generative adversarial network,'' *IEEE Transactions Image Processing*, vol. 29, pp. 9627--9640, Oct 2020.

[24] Y. Wei, L. Wang, H. Cao, M. Shao, and C. Wu, ''Multi-attention generative adversarial network for image captioning,'' *Neurocomputing*, vol. 387, pp. 91--99, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219317825

[25] X. Chen and C. Lawrence Zitnick, ''Mind's eye: A recurrent visual representation for image caption generation,'' in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[26] A. Karpathy and L. Fei-Fei, ''Deep visual-semantic alignments for generating image descriptions,'' in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, ''Show and tell: A neural image caption generator,'' in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156--3164.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, ''Bleu: a method for automatic evaluation of machine translation,'' in *Proc. 40th Annu. Meeting Association for Computational Linguistics*, 2002, pp. 311--318.

[29] R. Mihalcea and P. Tarau, ''Textrank: Bringing order into text,'' in *Proc. 2004 Conf. Empirical Methods in Natural Language Processing*, 2004, pp. 404--411.

[30] S. Brin and L. Page, ''The anatomy of a large-scale hypertextual web search engine,'' *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107--117, 1998, proc. 7th Int. World Wide Web Conf. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016975529800110X

[31] C. Florescu and C. Caragea, ''Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents,'' in *Proc. 55th Annu. Meeting Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1105--1115.

[32] R. Wang, W. Liu, and C. McDonald, ''Corpus-independent generic keyphrase extraction using word embedding vectors,'' in *Software Engineering Research Conf.*, vol. 39, 2014, pp. 1--8.

[33] S. Rose, D. Engel, N. Cramer, and W. Cowley, ''Automatic keyword extraction from individual documents,'' *Text Mining: Applications and Theory*, pp. 1--20, 2010.

[34] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, ''Yake! keyword extraction from single documents using multiple local features,'' *Information Sciences*, vol. 509, pp. 257--289, 01 2020.

[35] M. P. Grootendorst, ''Keybert,'' (Accessed Feb. 28, 2023). [Online]. Available: https://maartengr.github.io/KeyBERT/

[36] I. Tenney, D. Das, and E. Pavlick, ''Bert rediscovers the classical nlp pipeline,'' *arXiv preprint arXiv:1905.05950*, 2019.

[37] T.-Y. Lin *et al.*, ''Microsoft coco: Common objects in context,'' in *Computer Vision--ECCV 2014: 13th European Conf., Zurich, Switzerland, September 6-12, 2014, Proc., Part V 13*. Springer, 2014, pp. 740--755.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818--2826.

[39] S. Srihari, ''Rnns: Teacher forcing.'' [Online]. Available: https://cedar.buffalo.edu/~srihari/CSE676/10.2.1%20TeacherForcing.pdf

[40] Y. Goldberg and O. Levy, ''word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method,'' 2014.

[41] R. Řehřek and P. Sojka, ''Gensim—statistical semantics in python,'' *Retrieved from genism. org*, 2011.

[42] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, ''Deep captioning with multimodal recurrent neural networks (m-rnn),'' *arXiv preprint arXiv:1412.6632*, 2014.

[43] H. Katpally and A. Bansal, ''Ensemble learning on deep neural networks for image caption generation,'' in *2020 IEEE 14th Int. Conf. Semantic Computing (ICSC)*. IEEE, 2020, pp. 61--68.

APPENDIX

Twitter API Usage

Once you get the Twitter API access and all the required access tokens, keys, and secrets, we can fetch the tweets using the following code.

```
import tweepy

api_key = 'Obtained API Key'
api_key_secret = 'Obtained API Key Secret'
access_token = 'Obtained access token'
access_token_secret = 'Obtained access token secret'

auth = tweepy.OAuthHandler(api_key, api_key_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Search for tweets with specific keywords
query = 'horse OR grass' # We can modify this query as needed
# Reference: https://developer.twitter.com/en/docs/twitter-api/v1/
    tweets/search/guides/standard-operators
max_tweets = 10
searched_tweets = []
for tweet in tweepy.Cursor(api.search_tweets,
                           q=query,
                           lang="en").items(max_tweets):
    searched_tweets.append(tweet)
print(f"Query: {query}")
for i, tweet in enumerate(searched_tweets, 1):
    print(f"{i}) {tweet.text}")
```