DEPARTMENT OF
COMPUTER SCIENCE

RICARDO FREITAS BARQUEIRA
BSc in Computer Science and Engineering

MULTIMODAL ON-THE-FLY NEWS MEDIA EXPLORATION

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
March, 2023

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

MULTIMODAL ON-THE-FLY NEWS MEDIA EXPLORATION

RICARDO FREITAS BARQUEIRA

BSc in Computer Science and Engineering

Adviser: David Semedo
Assistant Professor, NOVA School of Science and Technology

Co-adviser: Rui Nóbrega
Assistant Professor, NOVA School of Science and Technology

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
March, 2023

Multimodal On-the-fly News Media Exploration

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.16) [28].

To my dear family, without your unwavering love, support, and encouragement, this project would not have been possible.

Acknowledgements

First of all, I am immensely grateful to my teacher David Semedo, whose guidance, encouragement and expertise have been instrumental in shaping my thesis journey. His unwavering dedication to education and his commitment to excellence have inspired me to push myself to new heights. I would also like to thank teacher Rui Nóbrega for his advice, support and helpful feedback that led me on the right path.

Second, I am thankful to NOVA School of Science and Technology for providing me with the opportunity to pursue my academic goals and for equipping me with the necessary tools and resources to complete this thesis.

Finally, I want to thank my friends and colleagues for their support which inspired me to always push further. I am grateful to my family for their unwavering belief in me and the endless motivation they have provided which allowed me to pursue my academic aspirations and achieve this significant milestone in my life.

“Believe you can and you’re halfway there.”

(Theodore Roosevelt)

Abstract

Information is presented to us in many ways and one of the most popular and trustworthy sources of information are the news media. Every day, news events from around the world are broadcasted through digital platforms and comprise a wide range of topics, divided into different categories and written by a diverse number of authors. These are presented to us online in the form of text but also in the form of images that help us to visually contextualize and "witness" the event with our own eyes. This way of presenting news, results in a multimodal news articles format.

Most news sites present us on their landing page with the latest and most popular news, allowing users to search for specific topics. However, given the large number of articles, especially on topics such as "COVID-19" or "War in Ukraine", enabling users to get a complete picture of the events and their origins in a dynamic and effective way becomes a particularly difficult task. Having a complete picture of the events also helps the users to be less susceptible to biased interpretations.

This thesis investigates zero-shot deep multimodal approaches for the news domain that is, given an image or a relevant text of a news article, we are able to analyze and aggregate related news pieces on-the-fly. Textual and visual processing with deep neural methods transform the text and images into the embeddings needed to reach the desired topic through context.

We collected the news' relevant information which resulted in approximately 4 million documents, processed the multimodal information to enable embedding-based searches and then provided aggregations of news according to topics and visualizations selected by the user using an interface that enabled the exploration of unfolding events. The outcome was a zero-shot news pipeline that made multimodal news pieces readily available for browsing in a semantic and efficient manner.

Keywords: News Media, Multimodal Data, Machine Learning, Zero-shot Learning, News Pipeline

Resumo

A informação é-nos apresentada de muitas maneiras e uma das fontes de informação mais populares e fiáveis são os meios noticiosos. Todos os dias, eventos noticiosos de todo o mundo são transmitidos através de plataformas digitais e compreendem uma vasta gama de tópicos, divididos em diferentes categorias e escritos por um número diversificado de autores. Estes são-nos apresentados online sob a forma de texto mas também sob a forma de imagens que nos ajudam a contextualizar visualmente e permitem aos leitores "testemunhar"o evento com os seus próprios olhos. Esta forma de apresentação de notícias resulta num formato de artigos de notícias multimodais.

A maioria dos sites de notícias apresenta-nos na sua página de destino as últimas e mais populares notícias e permite ao utilizador pesquisar tópicos específicos. Contudo, dado o grande número de artigos, especialmente sobre tópicos como "COVID-19"ou "Guerra na Ucrânia", permitir aos utilizadores obter uma imagem completa dos acontecimentos e das suas origens de uma forma dinâmica e eficaz torna-se uma tarefa particularmente difícil.

Esta tese investiga abordagens multimodais profundas de zero-shot para o domínio das notícias que, dada uma imagem e um texto relevante de um artigo noticioso, é capaz de analisar e agregar peças jornalísticas em tempo real. O processamento textual e visual transforma o texto e imagens nos "embeddings"necessários para chegar ao tópico desejado através do contexto.

Recolhemos a informação relevante das notícias que resultou em aproximadamente 4 milhões de documentos, processámos a informação multimodal para permitir pesquisas baseadas em "embeddings"e depois fornecemos agregações de notícias de acordo com os tópicos e visualizações que foram selecionadas pelo utilizador utilizando uma interface que permite a exploração de acontecimentos em desenvolvimento. O resultado foi um fluxo de notícias "zero-shot"que torna as notícias multimodais prontamente disponíveis para navegar de uma forma semântica e eficiente.

Palavras-chave: Noticias nos Media, Dados Multimodais, Aprendizagem Automática, Apredizagem Zero-shot, Fluxo de Noticias

Contents

List of Figures

List of Tables

1 Introduction

Acquiring information through news has never been easier than in this day and age with the internet proving to be the most dominant source of information. It was inevitable that news companies would also expand in this medium. This transition of newspapers to the digital world has allowed for more articles to be written, a wider range of topics to be discussed, and enables a much broader audience to visualize the news. News articles have many topics that address events from actuality such as technological and medical breakthroughs, elections, war, or even world-spread diseases. These are textual and visual evidences presented to us by journalists regarding the current state of the country or even the world and they need to choose the best textual and visual elements to contextualize the reader. News pieces can also be differentiated based on the development of the event they are describing, with some news pieces coming out while the event is still unfolding while others are only published when the event is concluded.

News pieces tend to come out at an incredible speed which allows the reader to stay "on top" of the events and there is also an inherent pressure for newspapers to be the first to share the information with the public, which results in acquiring a bigger share of the people that follow the news. Even though journalists are able to put out concise and informative news, it's possible that these news pieces could be missing additional information regarding the event. The follow-up news within the same topic can end up complementing each other with information as new articles come out, enabling the user to get a complete illustration of the event.

In some cases, such as the search engines from the New York Times or Google News, it is possible to obtain news from a specific topic by using specific keywords, but the way articles are presented disables a broader view of the topic and limits the search to merely textual similarities, thus restricting the context that is provided by the visual elements.

Therefore, news systems need to support the aggregation of topics as they unfold with semantic added context, provided by the visual elements of the articles. An efficient way to automate the organization of the news could be proven to facilitate the work of users that want to read all news from a specific event. To prevent biased interpretations, it's important that the users can have access to a broader set of on-topic news pieces, that more comprehensively cover the event and how it is unfolding.

1.1 Objective

The proposed system contrasts with traditional search engines because it counts on the additional context provided by the visual elements (which in our system are images that depict events) and accounts for new topics that may arise under the news domain. The context provided by images is an opportunity to retrieve more related news that the text couldn't get by itself. Processing the information from new topics enables associations with past events (using the context obtained in the embeddings) which can lead to better aggregations of topics.

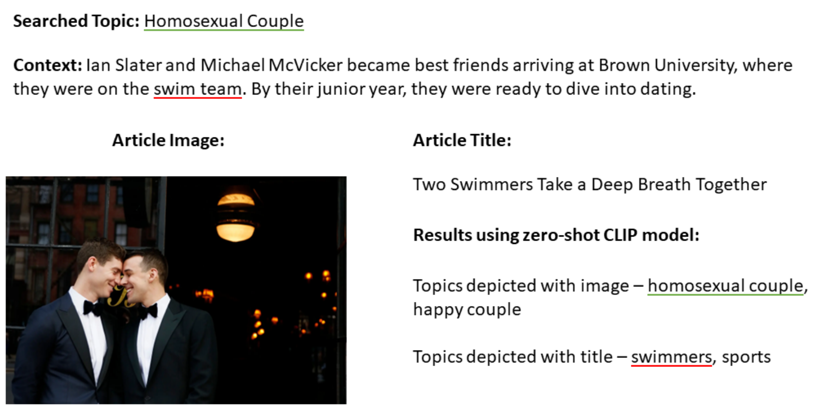
These new topics were extracted and processed on-the-fly disabling the possibility of training a model in real-time to process our samples and that's where zero-shot came in. The zero-shot technique enables the dynamic news pipeline because it looks at the similarities between multimodal elements for accurate predictions and doesn't require previous training to correctly match new content.

The goal of this thesis is to investigate a zero-shot deep multimodal model that analyzes and processes relevant textual and visual information in the context of the news, to support a system that aggregates events as they unfold to contextualize the user while preventing restricted views of events by staying accurate and flexible in regards to the topic.

Multimodal search is one of the means to achieve our objective. We assume the system receives the appropriate information through an interface that consists of the textual and visual elements of the article needed to perform the requested search. The search method needs input provided by the user such as a news piece and will have a result produced by analyzing the textual elements of the articles, analyzing the visual elements or by a combination of the two.

In the example shown in Figure 1.1, we can see how a news title can sometimes not be enough to reliably depict the topic the user searched for [11]. However, we can see that the image gives the context needed to validate the article as being part of the topic.

The result of said search is a list or a timeline of the articles that the system deems to be related to the topic. For the system to give the correct output, the system needs to have vast storage of articles that are selected based on the similarities of the textual and visual elements with the given input and also account for and prevent mistakes done in the embedding-based search.



Searched Topic: Homosexual Couple

Context: Ian Slater and Michael McVicker became best friends arriving at Brown University, where they were on the swim team. By their junior year, they were ready to dive into dating.

Article Image:

Article Title:

Two Swimmers Take a Deep Breath Together

Results using zero-shot CLIP model:

Topics depicted with image – homosexual couple, happy couple

Topics depicted with title – swimmers, sports

Figure 1.1: Example of recent news article where title doesn't give enough information to establish connection with topic but with image it does

1.2 Challenges and Research Hypothesis

The main problem with automatically organizing news pieces on-the-fly is that the news could be related to an event with no prior data which makes it a tough task for models trained with specific news to recognize these new events and categorize them properly.

Furthermore, we worked out an efficient way to extract the recent news streams and collect them in a format that facilitated their processing. The challenges were:

1. **U**nderstanding what information in the article should be extracted and then used for better predictions.
2. **U**nderstanding what strategy can be used to process news on-the-fly while still reaching significant representations of the data.
3. **F**inding out how to perform a similarity-based search using embeddings to reach relevant news.
4. **W**orking out a way to automatically extract the latest news and make them readily available to browse.

Pre-trained Visio-linguistic transformers models for zero-shot deep multimodal matching [11, 39, 37] were used to tackle some of the previously mentioned challenges since these pre-trained models can achieve very robust representations of the multimodal elements. Through a similarity-based approach over these representations, we were able to respond to complex queries using the similarities between texts or images and their respective dates. This led to better recognition of similar events throughout recent history and a much more proper aggregation of those events. Additional challenges arose with these models namely:

Relevance Criteria - The search gave us a list of results with relevant and irrelevant news for the specified query and we studied an objective method to find what makes a news article relevant.

Model issues - These models are trained with images and texts that are not necessarily related to news but instead rely on their vast number of diverse images and texts to correctly match content. Setting aside the information collected, mistakes were also done by the model possibly due to the lack of examples in the training regarding that topic.

Presentation Problem - After having the search component to perform queries on the collection of news and present the results, we designed an appropriate interface where users are able to search for news and also visualize its results.

Given our objective and the discussed challenges, we formulated the following hypothesis:

Is it possible to accurately aggregate pieces of news under the same topic with reduced prior data in the order in which the events unfolded using a zero-shot deep multimodal approach?

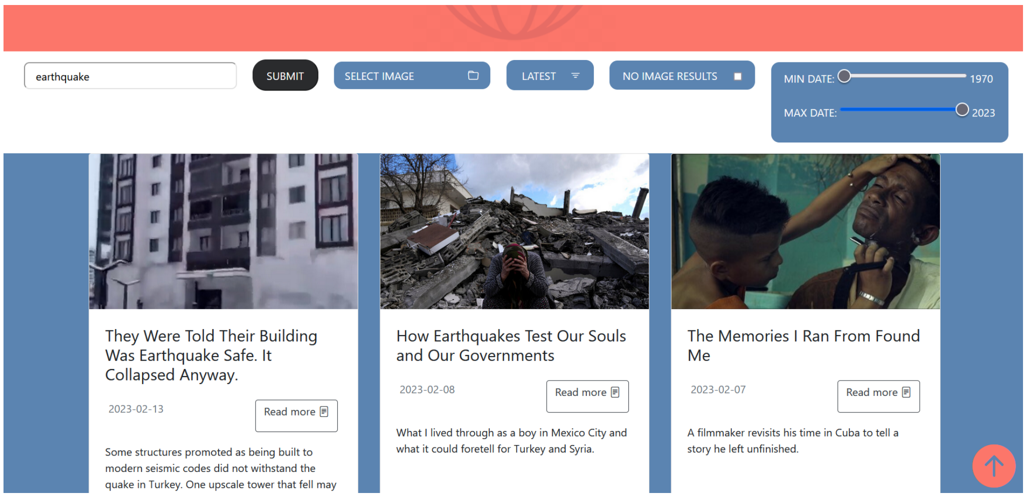
We addressed this hypothesis by evaluating how a state-of-the-art model performed for the news domain and under the zero-shot setting and also evaluated the impact of our dynamic news pipeline in the process of searching and aggregating news from the same topic.

1.3 Contributions and System Preview

This thesis contributed in the following aspects:

- Creation of a constantly updating news collection extracted using websites that allow users to access the relevant information from news articles and provision of processed information using a model capable of encoding information optimized for embedding-based searches. The news collection has approximately 4 million documents with all the relevant information including images from the extracted news and that same information in an embedded format. It continues to extract more information from articles and process it on-the-fly once it becomes available.

- Development of a multimodal search system that allows searches on the news collection using textual similarities but also similarity-based searches directly from the embeddings that can perform different matching combinations of both textual and visual modalities.



They Were Told Their Building Was Earthquake Safe. It Collapsed Anyway.

2023-02-13

Some structures promoted as being built to modern seismic codes did not withstand the quake in Turkey. One upscale tower that fell may

How Earthquakes Test Our Souls and Our Governments

2023-02-08

What I lived through as a boy in Mexico City and what it could foretell for Turkey and Syria.

The Memories I Ran From Found Me

2023-02-07

A filmmaker revisits his time in Cuba to tell a story he left unfinished.

Figure 1.2: System Preview when searching the topic "earthquake" in the form of text with the results organized from the latest news article to the oldest

- Implemented search filters on the multimodal search with the objective of filtering out incorrect results by taking advantage of useful information on the news articles such as the images and the dates.
- Designed an interface that allows the usage of the multimodal search system and provides visualizations with relevant news for the topic while also being capable of showing sequences of events such as timelines.
- Performed a study that evaluated the performance of the multimodal search in different scenarios, the impact of the filters used to acquire more precise results in the multimodal search and the fidelity of the different types of aggregations of news presented to the user.

The users will be able to interact with our system through an interface and we will take a look at what they can expect. An example of the system can be seen in Figure 1.2.

The user can utilize text, images or both to search for relevant news. The news can be organized in different views depending on what the user desires. Finally, the user can filter out some of the results to reach more specific news.

The server processes the information received from the user, chooses the appropriate search algorithm based on the information gathered and provides the results accordingly.

The results are news from the news collection that the system considers being relevant for the search. Each news has information about the news article and a button redirecting the user to the full article.

Application URL : `https://api.novasearch.org/visualseek/news-on-the-fly/`

1.4 Document Organization

The remainder of this document is organized into the following chapters:

Introduction - This chapter describes the context and motivation, as well as the objective and challenges of this thesis.

Background and Related Work - This chapter provides an overview and an analysis of the techniques used by existing literature in the context of our work.

Creation of dataset - This chapter looks into the pipeline for the creation of our dataset and analytics for the dataset.

Developing the system - This chapter explores the different design choices that our system went through until it reached the final product.

Evaluation - This chapter makes for a sequence of experiments that evaluated the system in different scenarios considered to be relevant in order to further improve the system.

Conclusions and future work - This chapter entails our final thoughts on whether we were able to overcome the challenges and reached the objective proposed and also what can be added in future works.

2 Related Work

This chapter discusses previous works and more specifically their methodologies and results while considering the context of our work. This chapter will follow a similar order to the line of our work by starting with the extraction of news where we analyze works about news articles; the connection of images and text in the context of news and different types of news events. We will cover works on the study of zero-shot deep multimodal models in the context of news such as textual, and visual representations and the combination of both; contrastive learning; zero-shot classifier; zero-shot deep multimodal model, and event detection. Finally, we will show works in the process of browsing news where we analyze the ability to structure news events on the fly; file storage/databases where the processed information will be stored, and news sources to serve as our dataset.

2.1 News Extraction

In the next sections, we will talk about the first step in the line of our work which is the process of extracting news. In the process of extracting news, there are many factors and methods that should be taken into account and will be further explored in this chapter. We will be taking into account other works that explored those methods and get the information that could be proven to be relevant in the context of our work.

To begin the process of extracting news we need to first figure out what is the composition of a news article and what information could be relevant to us.

2.2 News Articles

News articles discuss current, recent, or older news of general interest or specific topics, and they are usually accompanied by eyewitnesses of the event such as photographs, accounts, statistics, graphs, recollections, interviews, polls, debates on the topic, etc. The images present in the articles have captions that describe those images.

The news composition follows this order:

1. **Headline** - The headline of a story is typically an appealing and complete sentence that sometimes omits the subject or verb.
2. **Abstract or billboard** - An article billboard is a summary of the article depicted in one sentence to encourage the reader to stop and read the article.
3. **Lead paragraph** - The lead usually doesn't pass two sentences and has a length of 20-25 words. A lead must balance the ideal of maximum information conveyed with the constraint of not prolonging the sentence past its limit.
4. **Paragraphs** - Paragraphs that form the majority of the article.

This composition follows an inverted pyramid style with all the most important information in the first paragraph or two and the less vital details pushed towards the end of the story [16]. This is important knowledge in regards to the information we want to extract from the news articles. We know that the headline will be the most relevant information we can extract, followed by the abstract and the lead paragraph, and finally the rest of the paragraphs.

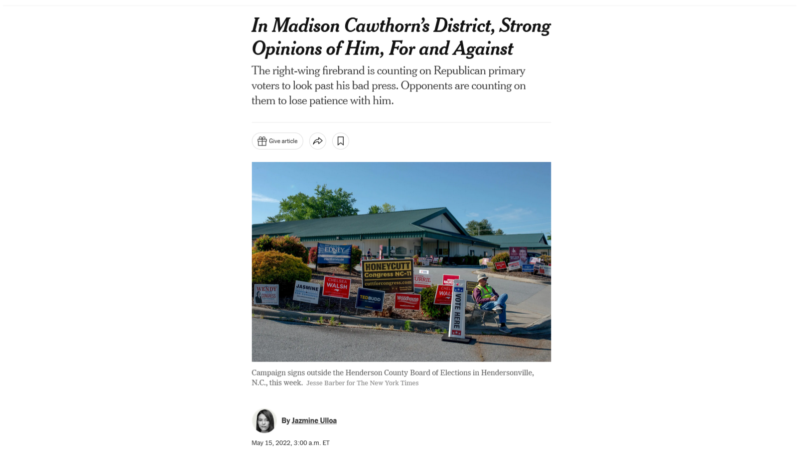
However, we also saw in the example of Figure 1.1 that the headline could also fail to depict the topic and so, further information is needed to successfully depict the topic. Of course, we shouldn't discard the rest of the information just yet because it could be proven to be useful in some cases.

For now, the headline, abstract, lead paragraph and paragraphs, images, captions, and date of the article seem to be the relevant information we need to progress into the next stages of news extractions. To evaluate their relevance with precision in the context of our work we need more experimentation. In Section 5.2.1 there are initial evaluations and results that demonstrated what type of information was more relevant.

We can expect that training with different combinations of textual elements could lead to significant improvements in the performance on the task of using an image as a query and retrieving the corresponding news piece and the task of using a news piece to retrieve the correct image [11]. Next, we should analyze the importance of the connection between images and text in the context of news articles.

2.3 Connecting Images and Text in the Context of News Articles

It's important for the authors of the articles to pick the best images to contextualize the user with the situation described in the text but it's also a complex endeavor to learn how to match those. For users, establishing the relations between text and image might be more straightforward but computationally mimicking this process becomes challenging. This analysis is important for our line of work to understand the limitations of the connection between text and images in a broader sense but also computationally speaking.



In Madison Cawthorn's District, Strong Opinions of Him, For and Against

The right-wing firebrand is counting on Republican primary voters to look past his bad press. Opponents are counting on them to lose patience with him.

Campaign signs outside the Henderson County Board of Elections in Hendersonville, N.C., this week. Jesse Barber for The New York Times

By Jazmine Ulloa

May 15, 2022, 3:00 a.m. ET

Figure 2.1: Example of recent news article from the New York Times website [1]

The connection between image and text can often be explained through many types of journalistic criteria such as how transparent the journalist was when writing the text and choosing the images for the article, how significant it is to provide the correct wording for a specific topic, how important is image aesthetic or simply image availability [30, 31, 33]. These are factors that must be considered when trying to associate visual and textual elements where many connections, between an image and a text piece, can be made possible. There are two other factors to take into account when talking about the connection between images and text in the context of news: temporal distances and spatial distances.

In the case of temporal distances, we can see the example of the War in Ukraine where the text mentions the events that happened in the location that was affected but the image can only go so far as to show the aftermath of what happened in the location. In this example, Figure 2.2, we can see how the image can contextualize the reader with the destruction caused by the event but not the event itself. This is something characteristic of news regarding catastrophes or events with similar magnitudes of danger.

Spatial distances are another factor when considering the connection between images and text because the text can be talking about an event but the image can show a similar event occurring in another location. The spatial distance is a harder factor to spot due to the lack of geographic knowledge when trying to locate the image but also relevant in the context of connecting text and images [33].

There was an investigation for the connection in the context of floods and in the context of the temporal distance [33] where they observed a good amount of images that depicted the occurring flood mentioned in the text but also many from the aftermath and proposed that if the image and text are not temporally aligned the merging of the two modalities will not be straightforward. They concluded, "it's important to abandon the assumption that images depict events described in the text". We can speculate this is due to the fact that it's a catastrophe-related article so the topic could have played a factor in the poor results obtained in their experiments(the factors we presented before could have also played a role) and therefore cant be generalized to all situations but important to keep in mind nevertheless.

In the computational setting, there are other factors related to connecting images and text such as the difference between the connections characterized for directly connecting visual elements to textual elements in contrast to connections where a direct connection can't be made and so a deduction is needed to properly connect both modalities but also figuring out relations between the named entities could lead to an added context which leads directly to better deductions. Finally, some text-image connections could be weaker because it's not easy to capture all the entities/information and the context if we get them only based on one image which leads us to believe better connections will be achieved with multiple images [11].

In sum, it's important to understand that the connection between text and images isn't always going to be the highest fidelity depending on many factors, and taking into account those factors and other challenges that arise in a computational environment is something we intend to analyze going forward. We will need to keep in mind these factors, properly evaluate them and if possible, act on them to lessen the impact of their limitations on our results. It's also important to keep in mind because of the nature of our work based on a dynamic news pipeline there won't be time to capture some connections acquired during training and so we need to take advantage of certain properties for example the added context associated with multiple images and the other factors explored in this chapter.

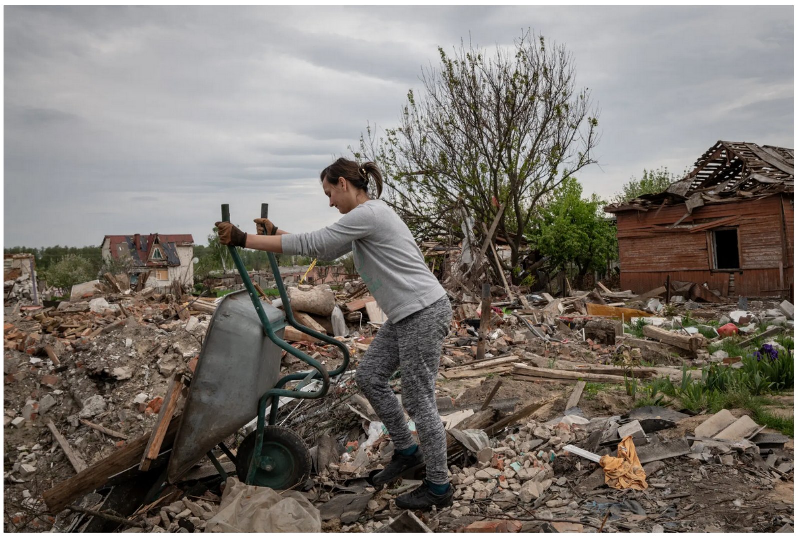


Figure 2.2: Photography in a recent news article that shows the aftermath of war in Ukraine [1]

2.4 News Dynamics

In this section, we will talk about the different types of news that we can identify depending on their progression and the importance of distinguishing the different types in the context of our work. There are three main types of news events we identified.

As we mentioned before, we want our system to account for new topics that may arise under the news domain while doing it on-the-fly and in some cases, it will be possible to predict what news topic will arise. These can be seen as recurrent types of news events. In Figure 2.3 there is an example of a recurrent type of news event which is the case of elections that we know it's an event that almost certainly will be reoccurring after every certain number of years, the same goes for the football world cup and many others.

There are also spike-based events where a certain event will lead to a lot of follow-up events on that same topic. Namely the COVID-19 disease news, the invasion of Ukraine, or even the Johnny Depp trial. In other cases, we have events that are not as easy to predict but it's possible that these events occur as a consequence of other ongoing events even though these could be events that explore different topics. As an example, the current economic situation in Europe due to the COVID-19 disease and the conflict between Russia and Ukraine makes it likely that news events of an economic crisis in Europe will arise in the future.

Lastly, we have isolated events that, unlike spike-based events, don't really have many follow-up events due to their shorter time window and their popularity. Events like the release of movies or other media, speeches by politicians. There is also a thin line that separates spike-based and isolated events. How much follow-up news must be made for the event to be considered spike-based? That line should be properly evaluated and described going forward.

Having this knowledge of what type of news events appear in news articles can play a role in identifying changes in established correlations and anticipating less predictable new events through deduction or even unpredictable new concepts. However, there's no telling how influential or performant it will really be when considering the works related to this topic are focused mainly on the textual modality and our context of extracting and processing news on-the-fly. This concludes the step of news extraction with the processing of these news being the next relevant topic to discuss.

2.5 Textual and Visual Representations

In the processing of visual and textual information, there are many techniques and methods that were developed across the years and will be further discussed in the following sections. We will be taking into account relevant works that explored those techniques of processing information and analyzing the models that show the most potential in the context of news.

Pre-training models [15, 23, 36, 17, 38] have immense relevance due to the great amount of diverse information they were trained in. These models use generic datasets to train the

Figure 2.3: Joe Biden is elected the 46th president of the United States [1]

data and initialized weights (weights saved from previous iterations of the neural network) which results in great efficiency when learning representations. The resulting textual representations that come from these models can depict a lot more different features than a model trained for a specific task, however, the different features have less impact if we consider a task-specific scenario which results in worse predictions [35]. In the context of our work, we need the ability to represent multiple features so we can solve the problem of novel concepts without belittling the accuracy of the results.

It's important to understand that the study for this work is only possible because of major breakthroughs of pre-training methods that have rapidly gained popularity due to their promising results when competing with task-specific trained models. Models like GPT-3 [12] that work in Flagship systems can now keep up across many tasks with bespoke models without needing much dataset-specific training.

This section will discuss different textual, visual, and multimodal representations and their relevance in the context of our work. We will be taking a look at different state-of-the-art pre-training models for text, image, and multimodal processing. These representations are known as embeddings and represent the mapping of a discrete variable to a low-dimensional numeric vector. With embeddings, we can correlate multiple inputs by computing distances between them in a given vector space.

2.5.1 Text Representations

We will need to represent the textual information on the news as embeddings in order to aggregate news on topics chosen by the user. This is possible because the text embeddings take the context of the sentence used, capturing the use of words in different situations. There are many techniques that can be used to turn the text into text-embeddings and many use scenarios such as translation, retrieval, summarizing, and question answering and we should inspect what characteristics of these techniques will benefit us.

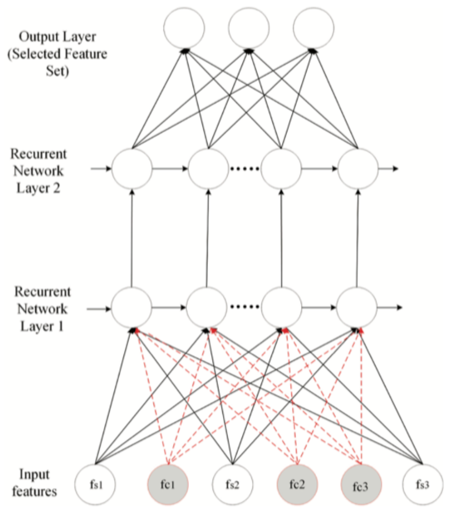

Output Layer (Selected Feature Set)
Recurrent Network Layer 2
Recurrent Network Layer 1
Input features
fs1
fc1
fs2
fc2
fc3
fs3

Figure 2.4: Recurrent neural newtork mechanism [14]

2.5.1.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) is an interesting model to look at for the fact that it was specifically designed to deal with sequences of data such as text sequences and shows good performance when capturing nonlinear relationships. These are a class of deep neural networks since they use as a basis a definite number of layers composed of neurons that propagate the information from layer to layer. However, RNN neurons are separate from typical feed-forward neural networks neurons because they receive not only an input vector but also the output vector of the previous layer as shown in Figure 2.4 Two common architectures Long Short-Term Memory Units (LSTM) and Gated Recurrent Units (GRU) that are based on RNN but are able to solve problems such as the vanishing gradient problem that occurs when the sequence of data is too long which results in some information being lost in the process. These architectures are able to learn what information is important which attempts to prevent the loss of relevant information [44, 19, 20].

Even though this technique shows the potential to process sequences of textual information and additional context into embeddings that can capture the relevant features in vector space, it still proves to be a challenge when dealing with longer sequences as they cannot retain information from the earliest elements. The technique known as the attention mechanism could be a solution to tackle this issue which is directly related to the transformer architecture.

2.5.1.2 Attention Mechanisms

Before looking into the transformer architecture and its benefits in achieving good textual representations, we need to look into the most important mechanism in the transformer architecture when it comes to persisting relevant information in longer sequences of textual information. The attention function gets the information from the entire sequence by calculating attention weights for every token simultaneously. The attention weight for each token is calculated by using the dot product between the query and key vectors which are obtained through the multiplication of the input word embedding with the query weights matrix and the key weights matrix respectfully [43].

The previous function produces its result by computing the set of matrices which is known as the attention head. Each attention head attends to the tokens that are relevant to each token but multi-head attention can do that with different definitions for relevance with the field representing relevance projected in successive layers. Multi-head attention enables the joint of attention from different representations at different positions which is not possible to achieve by averaging the values acquired by single attention heads.

The multi-head attention mechanism has been the standard method to attend over encoder outputs in transformers. This mechanism will be further seen in Section 2.5.3 where we will attend multiple domains with models that attempt to learn rich multimodal representations.

2.5.1.3 Transformer Architecture

The transformer model [43] follows an encoder-decoder with attention architecture that can be seen in Figure 2.5. The encoder consists of encoding layers that process the input and generates encodings that tell which part of the input is relevant passing this information as input for the next layer of encoders. The decoder also consists of decoding layers that take all the encodings and their contextual information to generate an output sequence. Each encoder and decoder will make use of the attention mechanism to weigh the relevance of every input and extract from them to produce the output. The number of encoders and decoder layers can lead to effectiveness improvement enabling the learning of different and more high-level representations by stacking these layers.

This architecture contains no recurrence which means it looks at tokens separately and not as a sequence and for that reason, it's important to keep the position of each token in the sequence with information such as the relative position or the absolute position. We can do that by adding positional encodings to the input embeddings at the bottom of the encoder and decoder stacks. For that reason, the transformer model learns positional embeddings to keep the order of the sequence.

BERT [17], Bidirectional Encoder Representations from Transformers, which is based on the transformer architecture [43], attempts to alleviate the unidirectionality limitation where the authors use a left-to-right architecture (tokens can attend to other tokens if they are positioned before them in the sequence and not after) by using a "masked language model" task that randomly masks certain tokens from the original sequence while trying to predict those tokens. This enables the combination of the context of the left side of the sequence with the context of the right side of the sequence.

This technique demonstrates that the length of the text sequence won't be an issue to process sequences of textual information and additional context into embeddings that can capture the relevant features in vector space because it provides us with a more structured memory to handle long-term dependencies which result in robust transfer performance across diverse tasks. Results obtained when comparing both LSTM (mentioned in Section 2.5.1.1) and Transformer architecture in the studies done with different models further support this idea [43, 17, 42]. This technique shows promise when we talk about textual representations but we also need to find an image encoder that can draw the important features in a vector space.

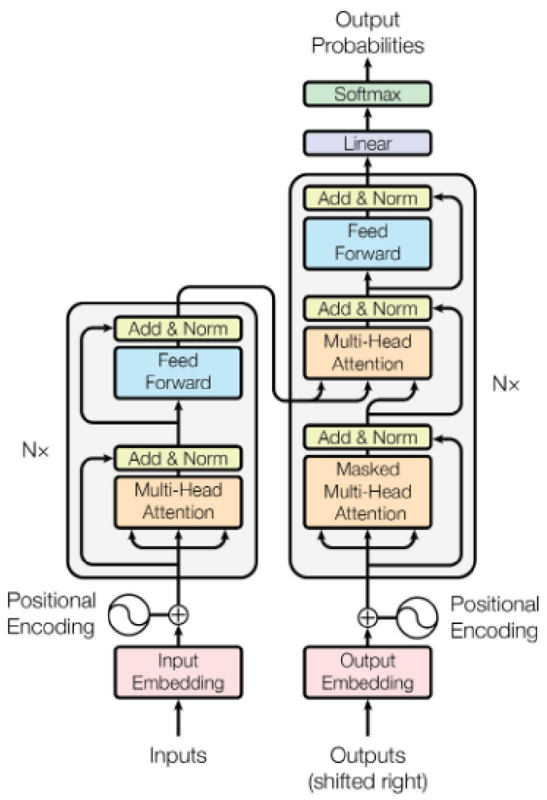


Figure 2.5: Transformer model mechanism [43]

2.5.2 Image Representations

We will need to represent the visual information on the news as embeddings in order to aggregate news on topics chosen by the user. This is possible because the image embeddings take the visual data, capturing many concepts in different scenarios. There are many techniques that can be used to turn the images into image-embeddings and many use scenarios such as object detection, face detection, classification, and segmentation and we should inspect what characteristics of these techniques will benefit us.

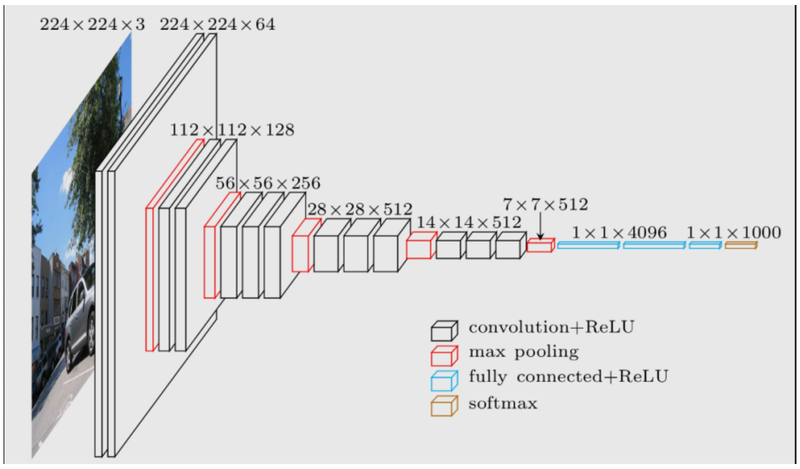


Figure 2.6: Example of a Convolutional Neural Network structure [34]

2.5.2.1 Convolutional Neural Network Technique

Deep convolutional neural networks (CNN) have led to a series of breakthroughs in image classification. These networks learned from low to high-level features and classifiers in an end-to-end multi-layer fashion with the level of features enhanced by stacked convolutional and pooling layers. Its ability to detect faces, objects, actions and location patterns could be worthy of notice in the context of news [34].

To recognize patterns in the data, CNNs are composed of different combinations of filters and layers which can be seen in Figure 2.6. The convolution layers will make the image become reduced to a feature map where every pixel turns into an input feature. This feature map originated from the dot product of the convolution kernel with the layer's input matrix and is usually subjected to an activation function known as ReLu that in its simplest form tells if the neuron is accounted for or not. It's followed by a pooling layer that reduces the dimensions of feature maps by combining features into a single representative feature in the following layer and can be max pooling or average pooling where max uses the maximum value of the local cluster of neurons in the feature map and average uses the average value as the representative feature. Multiple convolution layers and pooling layers are applied before reaching the final matrix that is then flattened to have a single column of inputs and sent to a fully connected layer that connects every neuron in one layer to every neuron in the next layer and is followed by the classification of the images which is usually associated with a loss function like the Softmax [34].

Various modifications have been made to this technique to account for problems like the vanishing gradient which prevented certain neurons from changing their weight values and ultimately preventing training, such as the Residual Network architecture (ResNet) [22] that allows building much more complex networks with thousands of convolutional layers which results in better and more detailed pattern recognition.

Despite the major breakthroughs accomplished by CNNs, it comes with its issues such as the pooling layer that because it reduces the dimensions of data it could lead to potentially valuable information being lost in the process and certain relationships between parts of the image ignored which leads to visual context gone missing. A transformer model in the context of image processing could solve this issue.

2.5.2.2 Vision Transformer Technique

As mentioned before, transformer techniques possess characteristics that allow maintaining all the information and all the relationships with the different parts of the image maintained. Inspired by the transformer success in NLP, Dosovitskiy et al. [18] proposed the ViT model that uses transformer techniques such as self-attention layers instead of the previously studied CNN.

The vision transformer receives a sequence of patches from the original image as input and similarly to BERT we append a learnable embedding to the sequence of patches which serves as the image representation used to perform classification and add a positional embedding to keep track of the positional information. To achieve the output consisting of patches embeddings which are then fed to the transformer encoder, the patches from the original image are flattened and transformed through a trained linear projection reducing their dimensions [18].

The main issue with using the transformer technique in the visual processing context is scalability. If the image size increases it becomes exponentially more difficult to calculate the combinations of each pixel. This technique however overcomes this issue by splitting the image into different patches that are handled by the model as if they were tokens in an NLP application. These patches are a great workaround mostly because the pixels that are most relevant to keeping the relationships are the ones closer to themselves. In Figure 2.7 we can see an example of how the different patches are processed into embeddings, added an additional positional embedding, and the resulting sequence is fed to a transformer encoder which finally results in the final image embedding with all the important features and relationships between them retained.

The results of ViT also showed great promise with great integration of information throughout the image, greater efficiency with smaller patches with impressive results in image classification datasets matching other state-of-the-art models with the added benefit of being relatively inexpensive to pre-train. This suggests that the Vision Transformer is an adequate model to consider for the system being developed in this thesis.

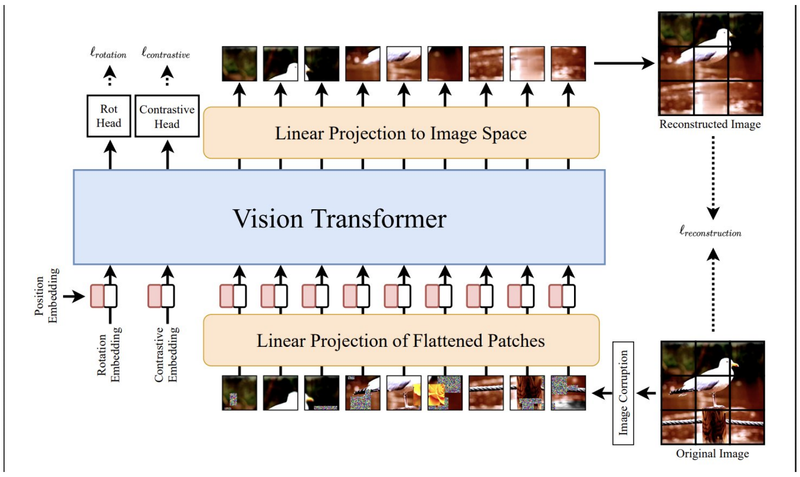


Figure 2.7: Simple vision transformer mechanism [10]

2.5.3 Multimodal Representations

While the text representations and image representations can be acquired using the techniques we have shown before, our proposal is to connect both of these representations into one final representation known as a multimodal representation. Multimodal representations should enable the model to attend to both text and vision modalities which is something crucial in the context of our work.

Multi-head attention briefly discussed in Section 2.5.1.2 is prominent in models that attempt to create multimodal representations. In the Transform and Tell model used to create new enriched captions for images in the news setting, there is the need to attend over four context domains such as image, text, faces, and objects. Each domain has many attention heads and the attention from each head is calculated and concatenated into overall domain attention which by finally concatenating the various domains and feeding into a feedforward layer will result in the final output that will be used as input in the next transformer block [42].

There are some possibilities on how we can connect both images and text in the same vector space combining both their relevant features: there are models that focus on richly connecting vision and language with a joint attention model meaning the language and vision networks at some point connect this is the case of LXMERT [39] and VinVL [45] or there are models with a cross-modal architecture that does not densely connect the two domains of text and vision with a joint attention model meaning the language and vision networks act disconnected this is the case of CLIP [37] and VisualBERT [29].

LXMERT model is constituted by a language encoder based on the transformer model from Section 2.5.1.3, an object-relationship encoder based on the vision transformer technique from Section 2.5.2.2 and a cross-modality encoder that helps the model trade information between both modalities so that it learns cross-modality representations. It was pre-trained with large amounts of text-image pairs via five diverse pre-training tasks: one language task, two visual tasks, and two cross-modality tasks.

For the language task, they masked random words and the model is asked to predict the masked words. The two visual tasks used were masking random objects and masking their region of interest (RoI) feature and the model was asked to get the value of the RoI feature and to identify the labels of the masked objects. Lastly, for the cross-modality task, they used cross-modality matching and image question answering tasks and asked the model if the image and sentence match each other and to predict the answer to image-related questions. These tasks allow the model to learn intra-modality and cross-modality relationships.

Visual News Captioner [26] also pointed out how it's a limitation to encode the image and article separately, ignoring the connection between them during encoding. They propose a Visual Selective Layer that updates textual embedding with a visual information gate in order to generate representations that can capture contextual information from both images and articles. However, this multimodal representation was designed for the task of generating image captions which are significantly different than our news tasks of news retrieval and aggregation based on topic.

CLIP model [37], seen in Figure 2.8, is constituted by a language encoder based on the transformer technique from Section 2.5.1.3, two different architectures for the image encoder for testing purposes which are a modified ResNet-50 based on the residual network architecture from Section 2.5.1.1 and the ViT based on the vision transformer from Section 2.5.2.2 (the results reported were obtained with the vision transformer because it was found to perform best) and the only interaction between these modalities is a linear projection to map from each encoder's representation to the multi-modal embedding space. It was pre-trained with large amounts of text-image pairs and the pre-training consists in jointly training an image encoder and text encoder (image and text tasks are similar to the ones seen before) to maximize the cosine similarity of the image and text embeddings while minimizing the cosine similarity of the embeddings of the incorrect pairings.

We will return to these models on Section 2.6.3 when we compare their performance on a zero-shot setting.

2.6 Zero-shot Models for Images and Texts

In this section, we will look into specific techniques required for our line of work such as contrastive learning, and zero-shot transfer, and evaluate models when they apply these techniques. These techniques are incredibly relevant for only they enable the on-the-fly processing of textual and visual information due to their similarity-based approach properties which makes it possible to compare embeddings acquired through pre-trained models without any further training.

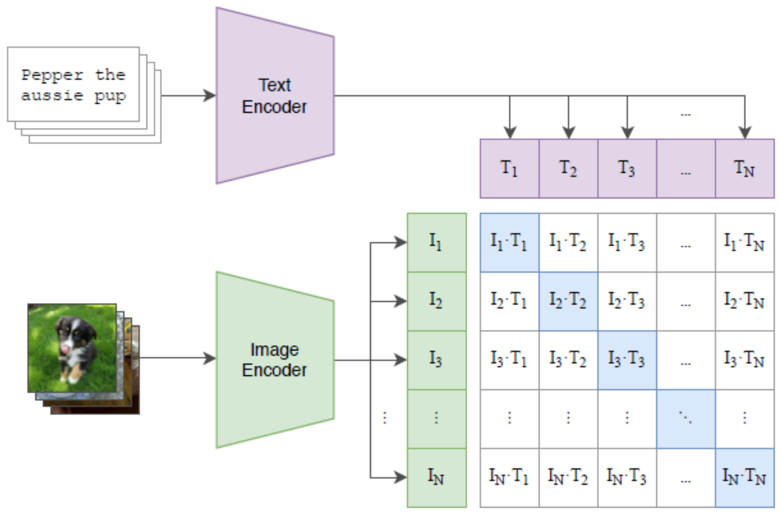

Pepper the aussie pup
Text Encoder
Image Encoder
T1 T2 T3 ... TN
I1 I2 I3 ⋮ IN
I1·T1 I1·T2 I1·T3 ... I1·TN
I2·T1 I2·T2 I2·T3 ... I2·TN
I3·T1 I3·T2 I3·T3 ... I3·TN
IN·T1 IN·T2 IN·T3 ... IN·TN

Figure 2.8: Clip pre-training architecture (adapted from [37])

2.6.1 Contrastive Learning

Contrastive learning [13] is a technique used in the task of predicting which text is paired with which image that is performant in the self-supervised setting where no labels or annotations are used which may lead to significant improvements in regards to efficiency and in learning better representations for our use scenario, making it a technique worth analyzing.

Firstly, contrastive learning is a learning technique that, in essence, consists of looking at pairs of representations with different views of the same scene and learning high-level features about the data by maximizing the mutual information between them whereas in predictive learning we learn high-level features in one representation and use that information to predict the resulting view. In Figure 2.9 we can see the process of both predictive learning and contrastive learning where the main differences consist in the fact that in predictive learning the loss is measured in the output space with the learned latent representation predicting one view from the other and in contrastive learning the loss is measured in the representation space with representations learned by contrasting congruent and incongruent views [40].

When comparing both learning techniques, the results show that contrastive learning consistently outperforms predictive learning in the scenario where both the task and the dataset are unknown when evaluating the quality of representations. The two reasons that were conjectured for this result were that pixel remake loss associated with predictive learning usually forces conditional independence over the model while in contrastive learning this is not the case and also the random jittering and cropping between views allows the contrastive learning approach to benefit from spatial co-occurrence (contrast in the same view) and contrast across views.

This technique was used with model NewsEmbed [27] in the news domain specifically in document representation which requires document triplets. Triplets are denoted as anchor document, positive document, and negative document where it's expected that a well-trained encoder captures meaningful similarity in the embedding space such that the pair anchor, positive is more semantically similar than the pair anchor, negative. The model demonstrated strong performance regardless of context length through a series of unsupervised evaluations which is motivating to further explore this technique.

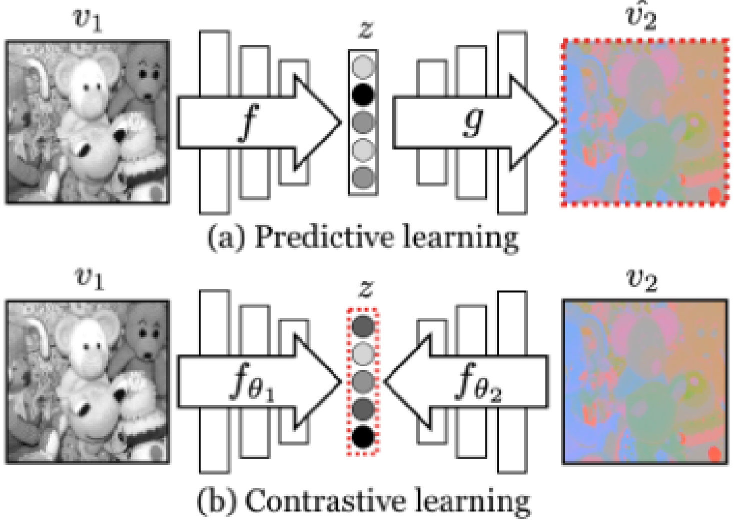


Figure 2.9: Predictive vs Contrastive learning [2]

2.6.2 Zero-shot Transfer

Using zero-shot transfer is a crucial part of our work that is usually associated with weaker performances where common benchmarks don't reach the desired accuracy parameters but with the necessary technical enhancements such as rich multimodal representations and models trained at a large scale, it opens up opportunities for more flexibility when it comes to providing dynamic and accurate outputs which could lead to impressive breakthroughs of efficiency in the area of deep learning. Zero-shot is specifically interesting in the multimodal and on-the-fly setting because it enables us to evaluate the similarity between two different representations without having to incur additional training necessary to classify data [37].

Unlike standard generalization in machine learning, where classifiers are expected to correctly classify new samples to classes they have already observed during training, with zero-shot learning no samples from the classes have been given during the training of the classifier. Usually, zero-shot learning refers to the study of generalizing to unseen categories in image classification but can also study generalization to unseen datasets. Our system will support a dataset that contains unseen samples of news that will be extracted on-the-fly and for that reason, they will have undetermined classes during classification which is why it's important to utilize the task-learning capabilities of zero-shot transfer to find similar representations. In Figure 2.10 we can see an example of how zero-shot prediction works in figuring out what text is associated with the image by looking at the similarities in their representations. The computation of this prediction is usually associated with cosine similarity which measures the similarity between two vectors of an inner product space [24].

Visual N-Grams [25] was the first known study in zero-shot transfer to existing image classification datasets that focused on evaluating the task-learning capabilities by using a generically pre-trained model. Their approach learns the parameters of thousands of visual n-grams (continuous sequence of words, symbols, or tokens) and uses the Jelinek-Mercer smoothing to maximize the probability of all text n-grams for a given image. They convert the dataset's class name into its n-gram representation and compute the probability according to their model, predicting the one with the highest score.

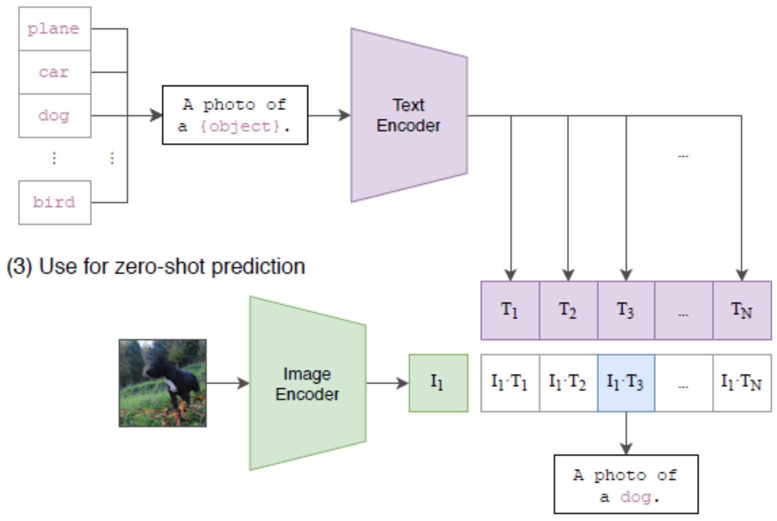

plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
T_1 T_2 T_3 ... T_N
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

Figure 2.10: Zero-shot prediction (adapted from [37])

2.6.3 Zero-shot Deep Multimodal Models for News

In this section, there will be a general discussion of all the techniques and models shown thus far and how they compare to each other in the specific settings we've set in our use scenario. As a reminder, so far we've seen many models that were able to attend both the text and vision modalities due to the multimodal representations achieved by tasks performed during pre-training such as the LXMERT and CLIP. We've also seen a model that used a generically pre-trained model but with the added benefit that it used zero-shot transfer to perform standard image classification in Visual N-Grams. The task done on these models to evaluate how they perform is primarily image classification which predicts if the image is related to the specified text with an average accuracy score (sum of the accuracy for every image-text pair and divided by the total number of image-text pairs).

First, we can see that Visual N-Grams [25] is a model that adapts well to our line of work with the specific characteristics we've seen before. The performance, however, shows that this model's task-learning abilities are not up to the tasks we've set such as learning novel concepts or even capturing the best representations for the text and vision modalities. In three different scenarios with three different datasets, we can see scores that range from 11.5 in ImageNet, 23 in SUN to 72.4 in aYahoo with zero-shot transfer image classification. The biggest differences between Visual N-Grams compared to the other models are related to its approach during pre-training and the four-year difference of technological advancements in the area: 10x smaller dataset than the other pre-trained models; 100x less computation required for prediction using vision models; 1000x less training compute because of hardware limitations; they don't use techniques that show benefits in performance such as the transformer-based models because they didn't exist at the time this model was created; the baseline wasn't trained from scratch instead they initialized from pre-trained ImageNet weights.

Secondly, LXMERT [39] is a model that showed great promise from its abilities to obtain rich multimodal representations that were acquired through the combination of the text representation and image representation into a cross-modality encoder and is pre-trained in a variety of tasks that promote the learning of both modality-specific and cross-modality relationships. This multimodal Transformer-based architecture has been demonstrated to be highly effective at modeling image and text semantics, by achieving state-of-the-art performance in Visio-linguistic benchmarks. The results are proof that, when in their studies with three different datasets, we can see scores that range from 60.3 in GQA, 72.5 in VQA to 76.2 in NLVR with linear probe (fully supervised) image classification. These are impressive results but we haven't seen this model's zero-shot transfer capabilities which is a required setting in the line of our work.

In the zero-shot transfer setting and trained with the NYTimes800k dataset [11] (dataset adopted for our work) and using as metric Recall@100 we can observe how LXMERT is not indicated to leverage zero-shot transfer image classification with the LXMERT model achieving a score of 0.13 in the task of news piece-image matching and a score of 0.24 in the task of news image-caption matching. In the linear probe setting (linear classification where all samples are associated with at least one label during training) this model can achieve in the same tasks scores of 0.61 and 0.93 accuracy. These results could be a reflection of the lack of generalization achieved during pre-training, as the model was trained with around 9 million image-sentence pairings from many datasets and that wasnt enough to establish an extensive amount of concepts necessary for task-agnostic scenarios such as the ones we're trying to pursue in our use scenario and much less task-specific scenario of news since the many image-sentence pairings used in training were gathered from generic datasets not related to the news domain.

Other works that were evaluated using news datasets such as the GoodNews dataset and the NYTimes800k dataset were the Transform and Tell [42] and the Visual News Captioner [26] however, these models were designed for the task of generating image captions using the textual and image domains to create enriched multimodal representations (with transformer technique) and using those representations to generate captions integrated with out-of-vocabulary named entities. Even though the tasks for our respective works are significantly different, it's worth noticing how the multimodal representations were able to capture enough multiple representative features so that these models were able to create novel out-of-vocabulary and accurate captions.

Finally, there is CLIP [37], a model that possesses multimodal representations just like LXMERT but not the representations achieved through a cross-modality encoder which provides more connections between the text and image reflected in the multimodal representation, instead it simply connects the image representation and text representation in the same vectorial space. However, there were techniques used in the creation of the model that suggests it might show good performance in the zero-shot transfer setting.

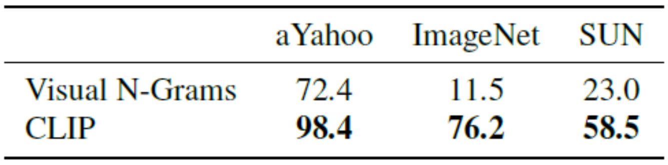
Compared to the two previous models, CLIP has by far the largest number of image-sentence pairings used during training with around 400 million this dataset will be able to cover a wider variety of visual concepts that the other models couldn't. The contrastive learning technique was used during CLIP's pre-training to predict if an image and a text snippet are paired together in its dataset and to perform zero-shot classification that capability is reused. All steps that occur during CLIP's pre-training intend to optimize the performance of a randomly generated representation to a computer vision dataset which contains 1 example per class and has 32,768 total classes defined via natural language description. During zero-shot evaluation the zero-shot classifier is saved and used for future predictions which reduces the cost of generating it across predictions. This model also improved its zero-shot performance through prompt engineering and ensembling where single words in the dataset were added prompts to help describe the image and for specific datasets customized prompts allowed for significant improvements of almost 5 points on average across datasets.

Looking at CLIP's zero-shot performance, we can see in Figure 2.11 that there is a major difference in the scores from CLIP and Visual N-Grams in the task of zero-shot transfer image classification with the improvement being a reflection of the many differences in the 4 years since the development of Visual N-Grams. It's still mostly sub-optimal when compared to the fully supervised CLIP version with around 10 to 25 points lower across datasets but when compared to a fully supervised baseline (ResNet50), it's able to outperform in 16 out of the 27 datasets used and it's able to stay competitive with the best 16-shot classifier results publicly known (BiT-M) in the task of image classification. Another great characteristic that zero-shot CLIP has is its high robustness to distribution shifts which means it doesn't exploit spurious correlations or patterns that hold in a specific distribution since it wasn't trained in that distribution, making bias, not an issue.

All of this data leads us to believe CLIP will be able to adapt and perform in the news task setting. Initial evaluations and results are in Chapter 5.2.1 where CLIP zero-shot performance is going to be tested with NYTimes articles.

Some final notes regarding the studies done with CLIP pointed out how CLIP learned to perform a wide set of tasks during pre-training including recognizing text within a digital image (OCR), geo-localization, action recognition, and many others which led them to analyze the model performance in relation to a downstream task and that is surveillance. This task also had the aim of analyzing potential future impacts of increasingly general-purpose computer vision models and came to the conclusion that, although it's not exceptional at surveillance-related tasks such as models supervised with that task in mind, it was still performing at a non-trivial level which could give us some hints on what we can expect in our evaluation in the news task setting. Finally, in CLIP's study, it was mentioned how the rich multimodal representations accomplished by models like the LXMERT could be a step to capture better connections between the image and the text and could lead to improvements in this line of work, this is something to keep in mind moving forward.

This concludes the processing of multimodal information topic which leaves the browsing news topic as the remaining one to analyze.



	aYahoo	ImageNet	SUN
Visual N-Grams	72.4	11.5	23.0
CLIP	**98.4**	**76.2**	**58.5**

Figure 2.11: Evaluation of Zero-shot prediction with Visual N-Grams and CLIP on 3 datasets [37]

2.7 Multimedia Event Detection

Event detection is an important task associated with the news extraction step and is particularly influential for our work for its ability to aggregate related news. The task of detecting events has been improving through the work of many researchers that found ways to diminish the shortcomings of this task such as being able to detect events in a clear and detailed manner, being able to detect events in a generalized setting where the task is unknown and extract event descriptors that are made available for the public to develop their own proposals with event descriptors being patterns in the text that allow identifying the event [21].

GiveMe5W1H [21] is a universal system for extracting events from news articles that utilizes six questions of "who did what, when, where, why, and how" to extract the necessary information from the news text to answer these questions and finally combining all the answers to describe the main event in the article. Three methods are used in this system in order to achieve the goal of event detection: in the preprocessing stage, they utilize techniques in the article text to achieve better extraction accuracy and transform the text into a brief format which is preferred for this model, followed by the phrase extraction phase that utilizes the processed text and picks sentence candidates that could answer the six questions and finally, the candidate scoring stage that scores the candidates independently given the question it's trying to answer and also does a combined score of all the candidates for one question. Despite their good results, this system is limited by not accounting for the visual modality when determining the event which is an indispensable part of our work.

In the work of Tong et al. [41] they propose a model where the visual modality is taken into account and takes center stage for its ability to not only represent the core event through an article image but also solve ambiguities that are introduced by event descriptors in the text. They propose a new dataset ACE2005 and a new model DRMM that aggregates image and text representations and achieves a new state of the art when compared to models that only use text as their resource which suggests the superiority of multimodal resources in the event detection task.

These works have allowed us to identify ways to detect events through event descriptors available in the news text and how news images can also enable the detection of events that sometimes the text can't. In our approach, we intend to take full advantage of both modalities' ability to detect events which will potentially result in more accurate news aggregations.

2.8 Embedding-based News Browsing

In the next few sections, we will talk about the last step in the line of our work which is the process of browsing news. In the process of browsing news, there are many factors and methods that should be taken into account and will be further explored in this chapter. We will be taking into account other works that explored those methods and get the information that could be proven to be relevant in the context of our work.

To begin the process of browsing news we need to first figure out what storage should we use to store the information we extracted as well as the processed information.

Databases are able to store information in an organized and structured manner while at a great scale and able to retrieve that same information with queries. We can also store different types of information such as text, images, and even files depending on what models the database supports. Most databases can successfully store the information extracted from news articles in a way that facilitates their search however if we consider the processed information such as the text and image represented in the form of embeddings it's not as straightforward, more specifically the ability to search for specific embeddings based on their similarity is a crucial functionality for our line of work and should be considered when exploring different solutions for our database.

2.9 Embeddings Indexing and Retrieval

K-NN, short for k-nearest neighbors, search is a form of proximity search whose purpose is to optimize the problem of finding a point in a given set that is closest to a given point where closeness is expressed as a dissimilarity function: the less similar the objects, the larger the function value.

OpenSearch [3] is a distributed search and analytics engine based on Apache Lucene that allows users to perform full-text searches on it with all of the features we can expect: search by field, search multiple indices, boost fields, rank results by score, sort results by field, and aggregate results but most importantly it supports the embeddings indexing and retrieval functionality by using the k-NN search as an optional plugin.

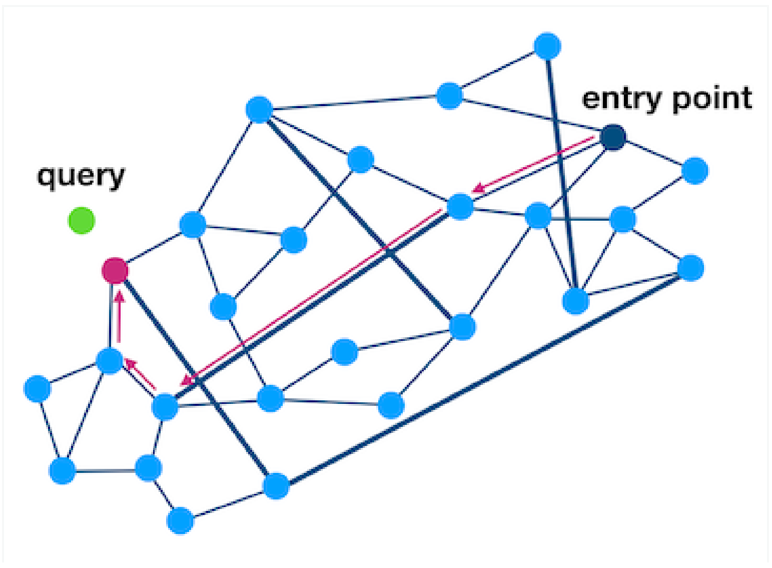


Figure 2.12: A depiction of an NSW graph built on blue data points [32]

The Hierarchical Navigable Small World Algorithm (HNSW) graph algorithm [32] is a fast and accurate solution to the approximate k-nearest neighbors (k-NN) search problem and it's the approach used when building the search engine in OpenSearch. In Figure 2.12 it shows the NSW graph and how it operates, starting at the entry point, with each iteration the greedy algorithm changes the point until it reaches the closest neighbor to the query. The ideal path from the entry point to the query's nearest neighbor is highlighted in magenta and is likely to be the shortest path from the entry point to the query's nearest neighbor with the dark blue edges representing long-range connections that help ensure the small-world property(a short average shortest path length and high clustering). HNSW extends the NSW algorithm by building multiple layers of interconnected NSW-like graphs and to find the approximate nearest neighbors to a query, it starts by searching the neighbors at the top layer and uses these points as entry points in the following layers.

NewsEmbed model [27] for the task of obtaining document triplets uses nearest neighbors search with the auxiliary embeddings obtained from each document (entity, image, and text embedding) to obtain all the top-K nearest neighbors for a document and decide if a given pair of documents should be considered positive (more semantically similar) or negative. This technique is used to generate triplets for training and not in the process of news retrieval like we intend to but is promising in regard to identifying similar embeddings.

The k-NN solution used in OpenSearch enables the development of scalable, distributed, and reliable frameworks for similarity searches with added support for calculations using cosine similarity which is used in zero-shot prediction making it a potential candidate for our similarity-based search.

2.10 Structuring News Events On-the-fly

In this section, we will see different ways to structure news events on-the-fly in ways that will facilitate users to obtain the latest news and also get the complete picture of events. Users will be able to provide us with a specific keyword that depicts a topic and the system will give the user the list of news related to the topic. This list can be structured by event classification or event timeline.

Event Classification structures events according to the relevance to the topic. The news articles that showed more similarity scores with the topic are shown to the user which is important to give the user the most trusted information about the topic. Event classification can be relevant for spike-based and isolated events. However, this could leave out some information that is not the most representative but can capture information from similar events in the past.

Event Timelines structure events according to the chronological order in which the events happened. The news articles related to the topic are picked from different dates in chronological order to show the user similar events from the past all the way up to the present. An example of an event timeline can be seen in Figure 2.13 where the topic is United States presidents and the news shown will follow the chronological order presented in the image. Event timelines can be relevant for recurrent and spike-based events. However, this could leave out the additional context obtained from the most representative news.

These structures have their advantages and disadvantages and it's up to the user to decide what structure can be more relevant to obtain the information he desires. It's also important to know what is the type of news event from Section 2.4 to facilitate the process of structuring news events.

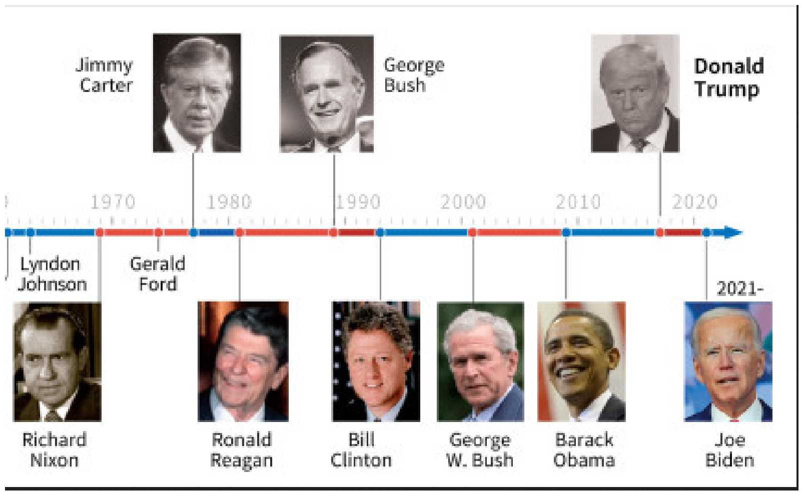


Figure 2.13: United States presidents event timeline [4]

2.11 News Sources

In this section, we will take a look at different news sources and observe what characteristics from the dataset we can take advantage of in the context of our work.

The characteristics we intend to have on our source of news are as follows:

1. – The composition of the news articles should be followed, with all the necessary text elements to give more context available to use.
2. – Most news articles should have one or more images that are authentic, provide complementary context over the aesthetic and provide a good connection or connections to the text.
3. – Easy for developers to access the information with well-structured application programming interfaces (API).
4. – Vast amount of news articles from sources that cover news from the beginning of online news until the most recent news which will allow us to determine different types of events (in section 2.5 we will be analyzing these types of events)

2.11.1 News API Dataset

The premise of News API [5] is that it can search for articles from over 80,000 news sources and blogs with its JSON API. The NewsAPI simple and easy-to-use REST API can search for all articles published on a specific date and from over 50 countries which sounds promising considering the vast amount of news we need in the context of our work.

The fact that we have 80,000 different news sources can be seen as a disadvantage because we can't prove if all these sources possess the characteristics that we are looking for in their news articles and it's also possible that a lot of these articles will have similar or even duplicated news.

With all of that said, there are a lot of unexplored territories when it comes to how different news sources can influence the results, and even though there is a sizeable amount of news, for our work, we should look to use a dataset that takes news articles from one trusted source but is also able to gather a vast amount of news with the required characteristics.

2.11.2 GoodNews Dataset

The GoodNews dataset was previously the largest dataset for news image captioning. Each example in the dataset contains an article, an image, and a caption. Since only the article text, captions, and image URLs are publicly released, the images need to be downloaded from the original source.

Several issues were observed when using the GoodNews dataset that may limit the system in our line of work such as text missing out often included in the first paragraphs which frequently contains important information and contains some non-English articles. The NYTimes800k dataset mentioned next is 70% larger, solves all of the previously mentioned issues, and is the more complete dataset [42].

2.11.3 NYTimes800k Dataset

The New York times dataset comprises online articles from the New York Times which is a well-known source for the authenticity of their texts and for being a reliable source of image captioning. The dataset contains a total of 445,870 articles, each with one or images that are used throughout the article to better contextualize the events described in the text, making it a total of 762,353 images [11].

From all the characteristics that the NYTimes800k dataset possesses, it's safe to assume this is not the dataset we need to start the process of news extraction. We intend to dynamically extract and process the news as they come out but this dataset disables the extraction of the most recent news.

2.11.4 General Discussion

In the previous sections, we came to the conclusion that none of the datasets used in previous works for developers to extract news allow us to retrieve the latest online news from a trusted source and process them in a dynamic way at the rate at which news are published every day and the information is available for the user to utilize.

This is why we need to go further than the options available and create our own dynamic dataset that contains all the news from the NYTimes800k Dataset in addition to the most recent news published under the NYTimes website. The most recent news can be acquired by the New York Times API which makes it easier to obtain information from the present but also from the past with the API allowing us to obtain news articles since 1852. In Figure 1.1 we can see an example of what information is available in a New York Times news article when we make a request to the API.

Finally, to evaluate our future solution, we intend to use many texts and images that depict a wide variety of topics and present an aggregation of news related to those topics based on the text and image representations from the extracted news. The success of this solution will be dictated by the performance of the similarity-based search using the multimodal representations (we need to reach significant scores using metrics such as accuracy, recall, and precision for topics comprehended in our news database), the efficiency in which past and recent news are aggregated and shown to the user and most importantly we need to make sure the list of news presented to the user is composed of mostly relevant news (we need to reach significant scores using metrics such as Recall at K with $K = 5, 10, 50, 100$ and Mean Average Precision for news in the resulting list). We follow image-text matching works [11, 39, 29] to evaluate our models.

3 Creating An Ever-Evolving News Collection

This chapter will focus on explaining the different tools we used to create our news collection. We will explore their functionalities and possible drawbacks while figuring how to apply them in our system. We will start right where we left off in the previous chapter with the making of our new dataset by inspecting the data provided by our news source; extracting the relevant information from news; preprocessing and storing the required data; indexing the extracted documents; automating the processes of extraction, preprocessing and indexation and the overview of the dataset up to date including statistics, examples and analytics.

3.1 Updating the New York Times Dataset

We mentioned how the different datasets used for previous works didn't suffice our needs for a dataset that could provide the latest news for our users and thus, we need to explore a different solution. In the next sections, we will be showing the proposed solution. This process started by determining our new data source.

In Section 2.11 we analyzed by looking at the characteristics of the datasets which dataset would be ideal for our use scenario and came to the conclusion that the New York Times Dataset. In addition, the New York Times API can satisfy most of our requisites.

The first step towards evaluating this dataset is to assess its limitations empirically. We managed to access this dataset and most of its information through the New York Times API [6]. The API provides us with different options for filtering the news we want such as the Archive API, Article Search API, etc. In our case, we want to extract the information from articles from as early in time as possible up until the present. We can find this information in the Archive API which possesses news from 1852 to the present day.

The Archive API allows the user to choose a year and a month and provides all the New York Times news articles for that time interval. The data fields available in JSON format for each news article are the following: abstract, URL, snippet, lead paragraph, image, caption, headline, keywords and the published date. The relevant data we need is the headline where we can find the most important terms, the lead paragraph where we can find more context on the news, the image to enable the multimodality in our system, the published date to use as a filter and possibly the caption and URL to redirect users who want to know more.

Even though the Archive API allows us to obtain news articles since 1852, we observed that the earlier articles in fact only give us a URL that redirects us to a page with the article as an image. This happens until 1970 when the articles start getting recovered and presented in digital format which allows retrieving the information but only the text. For us to be able to extract information from articles with both modalities (image and text) we need to fast forward to 2006 when online articles started to massively appear with images and text.

There were some other limitations we found with the API, more specifically the fact that the full article text is not available. Instead, they give us the lead paragraph and for articles with more than one image, the additional images and their captions are not available. The full article text can be something useful to give to our users because it has all the information regarding the news article. The additional images are important in case the main image is not representative enough of the news article and so we have more optional images tied to the same news article.

After we identified these drawbacks, we looked for ways to mitigate the losses on the information. When it came to the dates before 1970, we couldn't acquire the relevant information, since these were only available in the format of an image from an old New York Times newspaper. The images also did not exist before 2006 which made it impossible to acquire them. For the full article text and the additional images, it was manageable because that information was available on the webpage from the URL. We managed to use a crawler to extract this additional information so we can have it available in our dataset and add further context to the search. The crawler will be looked upon in the next Section 3.2.

These were the limitations we found out empirically with the remaining information being accessible. In the next section, we will analyze our empirical studies regarding news extraction.

3.2 Extracting Relevant Data

In this section, we look at executing our news extraction pipeline and determine what news will be extracted. To execute this step, we developed a pipeline from scratch with the New York Times API and BeautifulSoup web crawler library to acquire the required data.

The news extraction step implies that we extract all the relevant information from news available to us from 1970 until the most recent interval in the API. We are looking at a very large number of requests made to this API and for that reason, we need to limit the number of requests done daily and also limit the number of requests done in a minute. We can achieve this by reducing the number of months processed daily and using sleeps (stops execution of the program for a determined time) before each request.

We extract the information using the Archive NYT API. We specify which month and year we want our news articles and extract all the information from those articles: title, paragraph, URL, and date. After that, we use the URL of each article to obtain the rest of the information we need: the article text, all the images and their captions related to the article. This is accomplished by using the BeautifulSoup web crawler.

The BeautifulSoup web crawler [7] can extract information directly from HTML and XML files. We only need to provide the news article URL to the crawler and we gain access to all the information on the webpage through the HTML. From there, we identify the HTML elements we need and acquire their values.

At this stage, we needed to start figuring out what news articles would stay and which would be filtered out. We noticed some news didn't have a title, lead paragraph or article paragraphs and so we ignored these news articles altogether because we considered this information essential for the user. Some other news didn't have an image or caption but since this information was not essential for the user, we decided to extract those news articles anyways.

It's important to keep in mind that some news will not have an image associated with them either because the New York Times website didn't associate images with news articles before 2006 or simply because the news writer did not provide an image.

3.3 Pre-processing and Storing the Data

Following the step of news extraction, we need to store this information in structured files so it can be accessed later with the objective of efficiently searching said information. Before proceeding to the step of storing information we should look into one of the most important steps for creating our dataset: pre-processing the data.

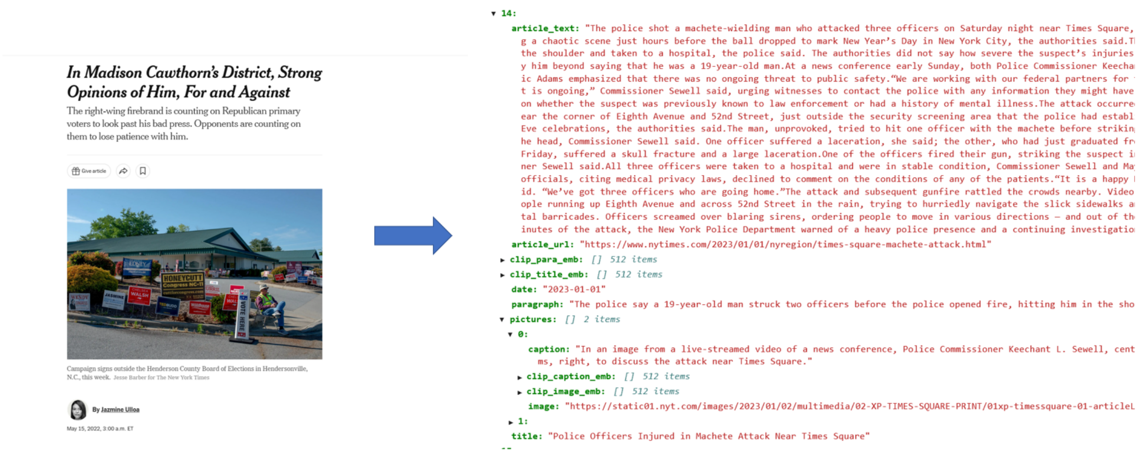
Pre-processing information can only be performed once the data is obtained from our data source and this is the phase where we prepare the data for operations we will perform later in our system. This preparation, in our scenario, consists in encoding the specified data into relevant embeddings by using our model of choice which is CLIP [37].

CLIP, which was one of the models reviewed in Section 2.6.3 and chosen to transform our data into relevant representations, was pre-trained with a massive number of text-image pairs and optimized for zero-shot matching scenarios, more specifically to encode data in relevant representations that can be compared with each other, resulting in a similarity score. This pre-processing stage will allow us to store the embeddings in our database and perform the operation of directly searching through embedding similarity.

The process of encoding requires us to change the data into a good and compatible format. The data compatible with this model are text and image files (JPG, PNG, etc...). The data that we will be transforming are the titles, the lead paragraphs, the images and the captions. All of the previous data will be transformed into a 512-dimensional vectorial space.

We realized a lot of the news lead paragraphs, titles and captions are longer than 200 characters which hinders the ability to turn this information into an embedding using our model. Therefore, we limit the number of tokens just enough to capture most of the context in those texts. We also noticed there are some texts with mostly noise in which case the text is disregarded. Some of the images are not supported by the model and are also disregarded.

Finally, the extracted and processed information is stored in JSON files. At the end of the process of extracting the news, we had a JSON file with all the news articles' information for each month between the years 1970 and 2022.


In Madison Cawthorn's District, Strong Opinions of Him, For and Against

The right-wing firebrand is counting on Republican primary voters to look past his bad press. Opponents are counting on them to lose patience with him.

Campaign signs outside the Henderson County Board of Elections in Hendersonville, N.C., this week. Jesse Barber for The New York Times

By Jazmine Ulloa

May 15, 2022, 3:00 a.m. ET

```
▾ 14:
    article_text: "The police shot a machete-wielding man who attacked three officers on Saturday night near Times Square,
      g a chaotic scene just hours before the ball dropped to mark New Year’s Day in New York City, the authorities said.Tl
      the shoulder and taken to a hospital, the police said. The authorities did not say how severe the suspect’s injuries
      y him beyond saying that he was a 19-year-old man.At a news conference early Sunday, both Police Commissioner Keecha
      ic Adams emphasized that there was no ongoing threat to public safety.“We are working with our federal partners for
      t is ongoing,” Commissioner Sewell said, urging witnesses to contact the police with any information they might have
      on whether the suspect was previously known to law enforcement or had a history of mental illness.The attack occurre
      ear the corner of Eighth Avenue and 52nd Street, just outside the security screening area that the police had establ
      Eve celebrations, the authorities said.The man, unprovoked, tried to hit one officer with the machete before strikin
      he head, Commissioner Sewell said. One officer suffered a laceration, she said; the other, who had just graduated fr
      Friday, suffered a skull fracture and a large laceration.One of the officers fired their gun, striking the suspect i
      ner Sewell said.All three officers were taken to a hospital and were in stable condition, Commissioner Sewell and Ma
      officials, citing medical privacy laws, declined to comment on the conditions of any of the patients.“It is a happy
      id. “We’ve got three officers who are going home.”The attack and subsequent gunfire rattled the crowds nearby. Video
      ople running up Eighth Avenue and across 52nd Street in the rain, trying to hurriedly navigate the slick sidewalks a
      tal barricades. Officers screamed over blaring sirens, ordering people to move in various directions — and out of th
      inutes of the attack, the New York Police Department warned of a heavy police presence and a continuing investigatio
    article_url: "https://www.nytimes.com/2023/01/01/nyregion/times-square-machete-attack.html"
  ▸ clip_para_emb: [] 512 items
  ▸ clip_title_emb: [] 512 items
    date: "2023-01-01"
    paragraph: "The police say a 19-year-old man struck two officers before the police opened fire, hitting him in the sho
  ▾ pictures: [] 2 items
    ▾ 0:
        caption: "In an image from a live-streamed video of a news conference, Police Commissioner Keechant L. Sewell, cent
          ms, right, to discuss the attack near Times Square."
      ▸ clip_caption_emb: [] 512 items
      ▸ clip_image_emb: [] 512 items
        image: "https://static01.nyt.com/images/2023/01/02/multimedia/02-XP-TIMES-SQUARE-PRINT/01xp-timessquare-01-articleL
    ▸ 1:
    title: "Police Officers Injured in Machete Attack Near Times Square"
--
```

Figure 3.1: News article relevant information from New York Times extracted and stored into a JSON file

3.4 Indexing Documents for K-NN Search

At this stage, we have acquired all the JSON files needed to perform different types of queries in order to get relevant news articles. The step that follows requires storing the relevant information and the embeddings into structures that allow the use of regular searches and k-NN searches. The k-NN searches are especially important because we require an efficient and dynamic way to perform embedding-based searches where the embedding chosen as query will be compared to the stored embeddings and give us the most similar ones.

We've mentioned previously in Section 2.9 that our solution would require a database that could support embedding-based search and that's where OpenSearch [3] comes in. OpenSearch is an open-source software that allows storing data in the form of JSON objects and performing search and analytics on this data. Each JSON Object will be identified by an identifier and will have multiple fields that can be used as matching elements in our search. We use the opensearchpy library that allows us to use Python code to perform the necessary operations of creating indexes, indexing and searching our data.

Creating indexes is an important step that allows OpenSearch to differentiate between different types of searches such as simple text and number matching or, in the case we want to search directly for embeddings using the OpenSearch search plugin k-NN, we must indicate in the index settings that the search must account for embedded fields. Once we identify what kind of search will be handled by said index, we must also designate what mappings will be used and what properties should be expected for the fields we specify.

3.5 K-NN indexing

We understood that 2 indexes would need to be created for our system but we created an additional one for testing purposes. These indexes were the index for the text embeddings, the index for the image embeddings and the index for regular text.

Text embedding index: The text embeddings index was flagged with possible embedded fields and created with 2 matchings which were the embedded title and paragraph with type k-NN vector, dimension 512 and we used cosine-similarity as our means to perform zero-shot classification.

Image embedding index: The image embeddings index was flagged with possible embedded fields and created with 2 matchings which were the embedded image and caption with type k-NN vector, dimension 512 and also used cosine-similarity.

Regular text index: The regular text index is not required for our system but can be used later on to perform some regular searches and compare its results to the searches using embeddings. This index was created with 2 simple matchings which were the title and paragraph with type String.

Our data, stored in JSON files, is organized in a JSON Array where each JSON object corresponds to a news article's relevant information. For us to access all the information of all the news articles, we began to iterate through the JSON Arrays in all the JSON files, gave each JSON object an identifier and finally associated the regular text index and the text embedding index to all of the JSON objects. However, the images and captions were in a JSON Array inside each JSON object which made them unable to be used as a field for matching. For that reason, we iterated through the image array in each JSON object, gave each image an identifier that would add to the JSON object identifier and finally associated the image embedding index to all the images.

These steps ensured that we had access to each news article's relevant information by specifying which field we wanted to match and what index had that field as a possible match. Later on, we added a mapping on the field date for each of the created indexes which allowed us to filter our search within specific time intervals.

3.6 On-the-Fly Extraction and Indexing of News Media

We've mentioned how it's important for the users to be informed on the latest events that are reported in the news and for that we managed to find a solution that would automate the various steps needed to make the latest news article's relevant information available and accessible to the users. This implies running the pipeline dynamically and on-the-fly.

At that point in time, we had managed to extract, preprocess, store and index news from January 1970 until November 2022. The proposed solution in order to keep our news updated would be to start the process as soon as news were available to be extracted and that would require us to check the New York Times API regularly throughout the day. Instead of making many requests to the API throughout the day, we decided it would be more efficient to only do it once a day and retrieve the news from the day before.

Compared to the manual extraction and indexing of news, there were a couple of changes we needed to make in the extraction step. During extraction, we would manually tell what year and month we want the articles to be from but if we need to extract the articles from the day before automatically that solution won't work. Instead, we used a Python library, *datetime*, to tell us the date for the day before and use it to extract the required articles.

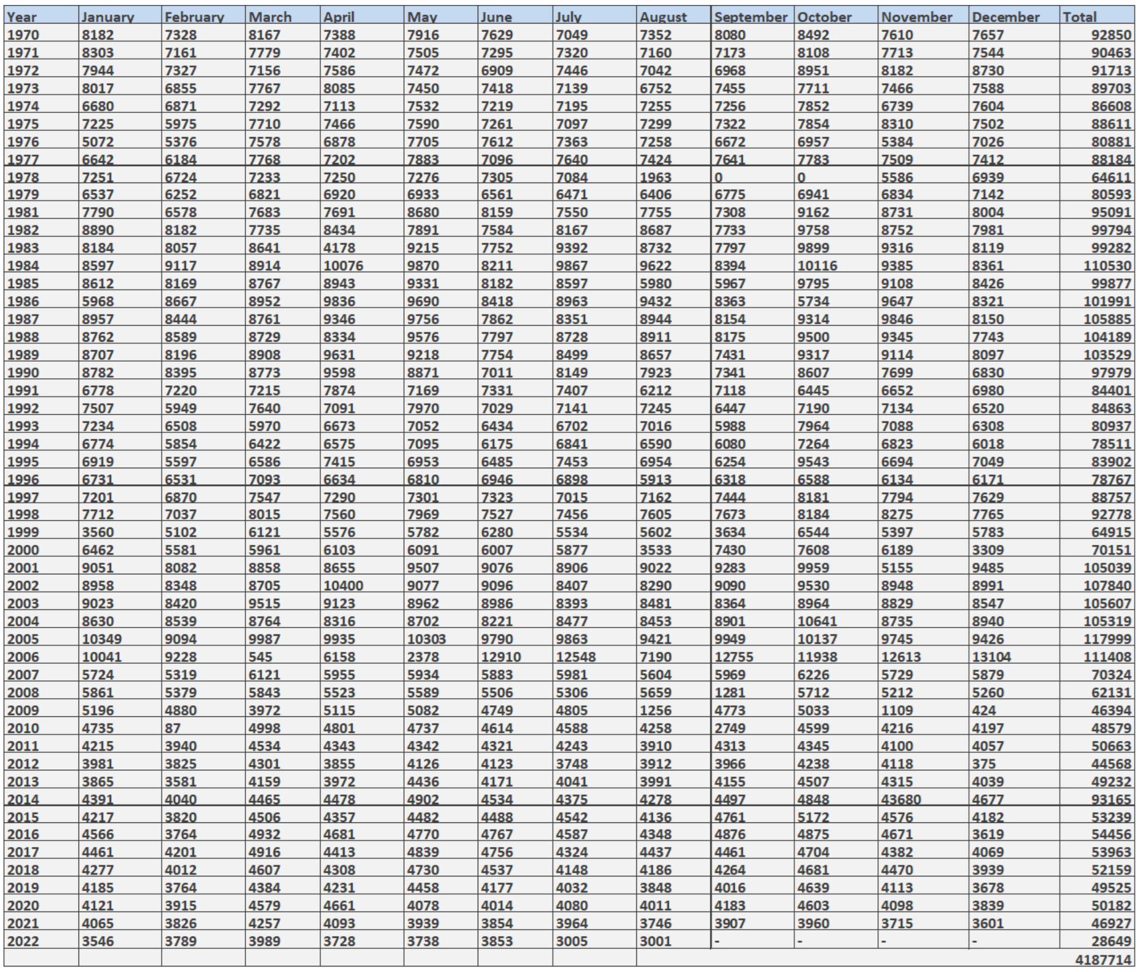
There was also a need to keep track of what identifier would be assigned during the automatic indexing step because we couldn't allow for repetitions of identifiers and so we made the identifiers based on the date of the article alongside the number of the article in the list of articles to make sure they were unique identifiers.

The last step in order to make the process fully automatic would be to find a way to run this script every day at the same time. This was accomplished with the cron job scheduler present in Unix-like systems.

That would conclude all of the steps we needed to create our dataset, make its contents fully accessible and compatible with multiple searching features that will be further explored in later sections while making sure it's updated with the latest events.

3.7 Dataset Statistics and Analysis

In this section, we will make an overview of our dataset, present statistics on the data available and analyze its intrinsic characteristics that makes it for a very diversified and complete article collection.



Year	January	February	March	April	May	June	July	August	September	October	November	December	Total
1970	8182	7328	8167	7388	7916	7629	7049	7352	8080	8492	7610	7657	92850
1971	8303	7161	7779	7402	7505	7295	7320	7160	7173	8108	7713	7544	90463
1972	7944	7327	7156	7586	7472	6909	7446	7042	6968	8951	8182	8730	91713
1973	8017	6855	7767	8085	7450	7418	7139	6752	7455	7711	7466	7588	89703
1974	6680	6871	7292	7113	7532	7219	7195	7255	7256	7852	6739	7604	86608
1975	7225	5975	7710	7466	7590	7261	7097	7299	7322	7854	8310	7502	88611
1976	5072	5376	7578	6878	7705	7612	7363	7258	6672	6957	5384	7026	80881
1977	6642	6184	7768	7202	7883	7096	7640	7424	7641	7783	7509	7412	88184
1978	7251	6724	7233	7250	7276	7305	7084	1963	0	0	5586	6939	64611
1979	6537	6252	6821	6920	6933	6561	6471	6406	6775	6941	6834	7142	80593
1981	7790	6578	7683	7691	8680	8159	7550	7755	7308	9162	8731	8004	95091
1982	8890	8182	7735	8434	7891	7584	8167	8687	7733	9758	8752	7981	99794
1983	8184	8057	8641	4178	9215	7752	9392	8732	7797	9899	9316	8119	99282
1984	8597	9117	8914	10076	9870	8211	9867	9622	8394	10116	9385	8361	110530
1985	8612	8169	8767	8943	9331	8182	8597	5980	5967	9795	9108	8426	99877
1986	5968	8667	8952	9836	9690	8418	8963	9432	8363	5734	9647	8321	101991
1987	8957	8444	8761	9346	9756	7862	8351	8944	8154	9314	9846	8150	105885
1988	8762	8589	8729	8334	9576	7797	8728	8911	8175	9500	9345	7743	104189
1989	8707	8196	8908	9631	9218	7754	8499	8657	7431	9317	9114	8097	103529
1990	8782	8395	8773	9598	8871	7011	8149	7923	7341	8607	7699	6830	97979
1991	6778	7220	7215	7874	7169	7331	7407	6212	7118	6445	6652	6980	84401
1992	7507	5949	7640	7091	7970	7029	7141	7245	6447	7190	7134	6520	84863
1993	7234	6508	5970	6673	7052	6434	6702	7016	5988	7964	7088	6308	80937
1994	6774	5854	6422	6575	7095	6175	6841	6590	6080	7264	6823	6018	78511
1995	6919	5597	6586	7415	6953	6485	7453	6954	6254	9543	6694	7049	83902
1996	6731	6531	7093	6634	6810	6946	6898	5913	6318	6588	6134	6171	78767
1997	7201	6870	7547	7290	7301	7323	7015	7162	7444	8181	7794	7629	88757
1998	7712	7037	8015	7560	7969	7527	7456	7605	7673	8184	8275	7765	92778
1999	3560	5102	6121	5576	5782	6280	5534	5602	3634	6544	5397	5783	64915
2000	6462	5581	5961	6103	6091	6007	5877	3533	7430	7608	6189	3309	70151
2001	9051	8082	8858	8655	9507	9076	8906	9022	9283	9959	5155	9485	105039
2002	8958	8348	8705	10400	9077	9096	8407	8290	9090	9530	8948	8991	107840
2003	9023	8420	9515	9123	8962	8986	8393	8481	8364	8964	8829	8547	105607
2004	8630	8539	8764	8316	8702	8221	8477	8453	8901	10641	8735	8940	105319
2005	10349	9094	9987	9935	10303	9790	9863	9421	9949	10137	9745	9426	117999
2006	10041	9228	545	6158	2378	12910	12548	7190	12755	11938	12613	13104	111408
2007	5724	5319	6121	5955	5934	5883	5981	5604	5969	6226	5729	5879	70324
2008	5861	5379	5843	5523	5589	5506	5306	5659	1281	5712	5212	5260	62131
2009	5196	4880	3972	5115	5082	4749	4805	1256	4773	5033	1109	424	46394
2010	4735	87	4998	4801	4737	4614	4588	4258	2749	4599	4216	4197	48579
2011	4215	3940	4534	4343	4342	4321	4243	3910	4313	4345	4100	4057	50663
2012	3981	3825	4301	3855	4126	4123	3748	3912	3966	4238	4118	375	44568
2013	3865	3581	4159	3972	4436	4171	4041	3991	4155	4507	4315	4039	49232
2014	4391	4040	4465	4478	4902	4534	4375	4278	4497	4848	43680	4677	93165
2015	4217	3820	4506	4357	4482	4488	4542	4136	4761	5172	4576	4182	53239
2016	4566	3764	4932	4681	4770	4767	4587	4348	4876	4875	4671	3619	54456
2017	4461	4201	4916	4413	4839	4756	4324	4437	4461	4704	4382	4069	53963
2018	4277	4012	4607	4308	4730	4537	4148	4186	4264	4681	4470	3939	52159
2019	4185	3764	4384	4231	4458	4177	4032	3848	4016	4639	4113	3678	49525
2020	4121	3915	4579	4661	4078	4014	4080	4011	4183	4603	4098	3839	50182
2021	4065	3826	4257	4093	3939	3854	3964	3746	3907	3960	3715	3601	46927
2022	3546	3789	3989	3728	3738	3853	3005	3001	-	-	-	-	28649
													4187714

Figure 3.2: Number of news articles from our dataset from January 1970 until August 2022

We will start by revising some of the dataset characteristics. The dataset comprises approximately 4.2 million documents as of November 2022 and they have all the relevant information in a news article such as the title, the lead paragraph and the rest of the paragraphs, the image and its caption, date and the news article URL. Besides that information, we also have the embedded information for the title, the lead paragraph and the image to enable search by embeddings. The articles were and are still being collected using the proposed pipeline that runs daily using the crontab tool to automate the process.

In Figure 3.2, we can observe how news were distributed over the years on our dataset. Over the years, we see that the number of news published has some slight increases and decreases but after 2009, there is an overall decrease in the number of news where in recent years the number of articles barely reaches 50000 per year.

The eight plots in Figure 3.3 indicate the distribution of news from eight different topics over the years: Racial Justice, Ukraine, Covid-19, Trump, Earthquake, Snowden, Climate Change and Gun Control.

Racial Justice Figure 3.3a shows the distribution of the topic racial justice throughout the years and we can consider it to be a recurrent type of news event when the topic is reported regularly every year. We can also observe that in the years 2020 and 2021, there is a spike in news reporting on racial justice topics.

Ukraine Figure 3.3b shows the distribution of the topic Ukraine throughout the years and even though it's slightly reported throughout the years, it's not until 2022 that this topic spikes with the event known as the Russian Invasion of Ukraine.

Covid-19 Figure 3.3c shows the distribution of the topic Covid-19 throughout the years and it's evident we can call it a spike-based event due to the popularity of reports in the years 2020 and 2021.

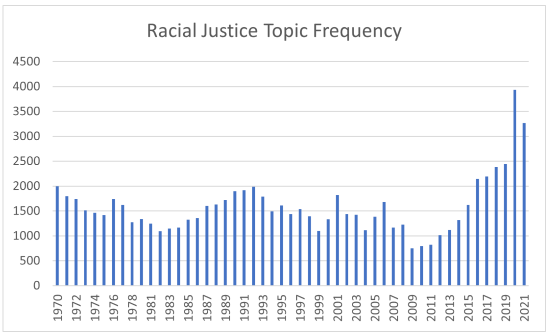
Trump Figure 3.3d shows the distribution of the topic Trump throughout the years and how this topic is spike-based because of the time interval of 2017 to 2020 when this topic gains massive popularity.

Earthquake Figure 3.3e shows the distribution of the topic Earthquake throughout the years and it's a recurrent type of news event because every year there are occurrences of that event.

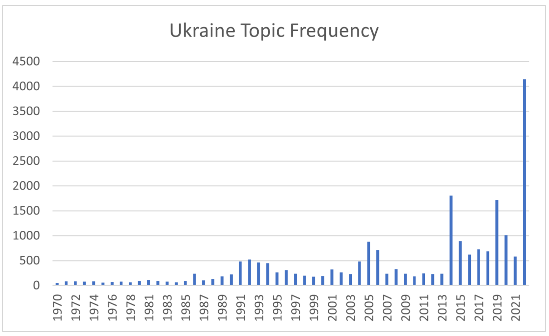
Snowden Figure 3.3f shows the distribution of the topic Snowden throughout the years and how this topic is spike-based because only in 2013 we saw massive popularity for the topic.

Climate Change Figure 3.3g shows the distribution of the topic Climate Change throughout the years and how this topic is a recurrent type of news because every year starting in the 2000s showed regular reports of this event.

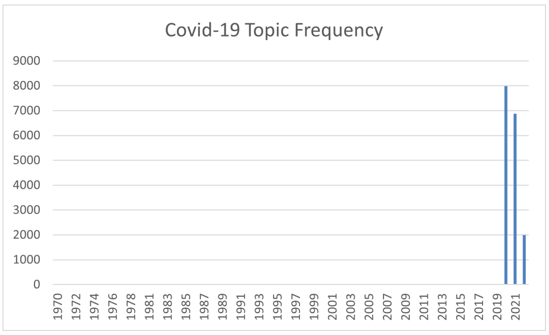
Gun Control Figure 3.3h shows the distribution of the topic Gun Control throughout the years and how this topic is a recurrent type of news where almost every year there are discussions on the topic.



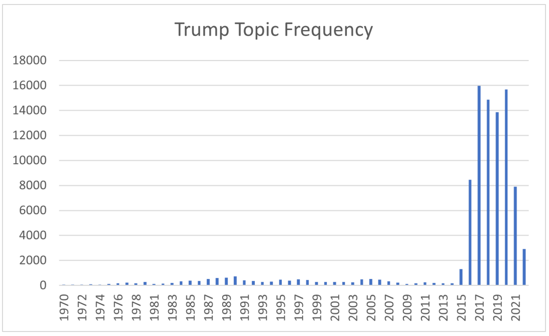
(a)



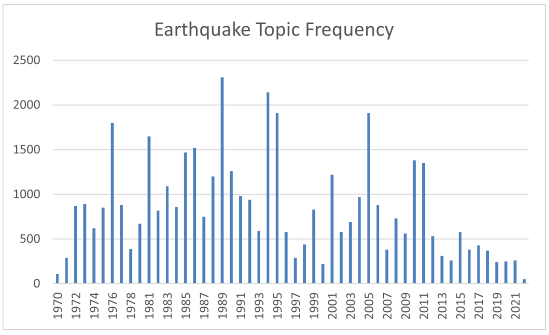
(b)



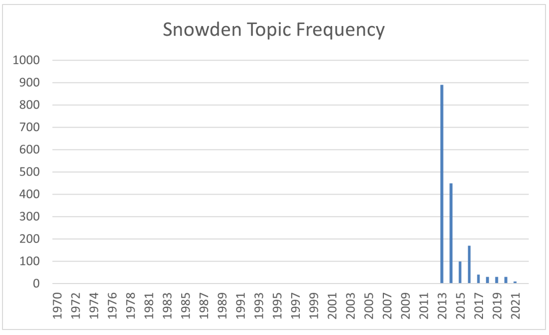
(c)



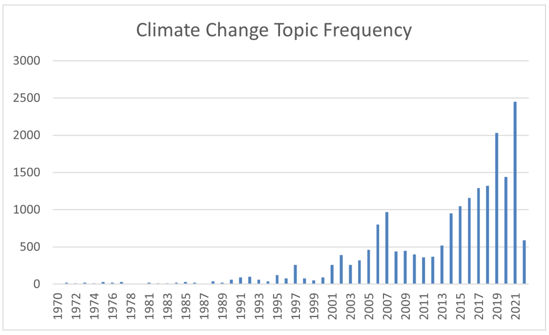
(d)



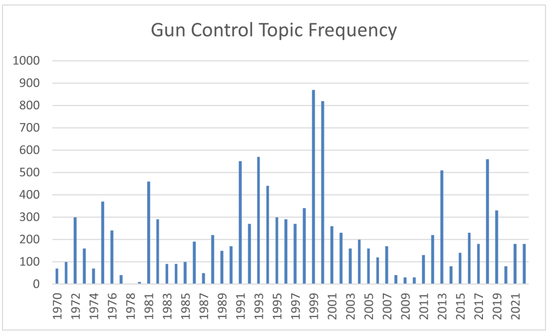
(e)



(f)



(g)



(h)

Figure 3.3: Distribution of eight different topics between the years of 1970 and 2022

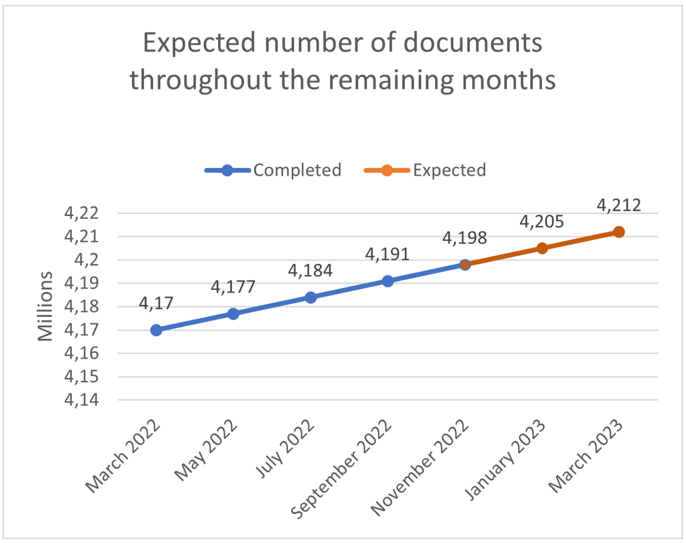


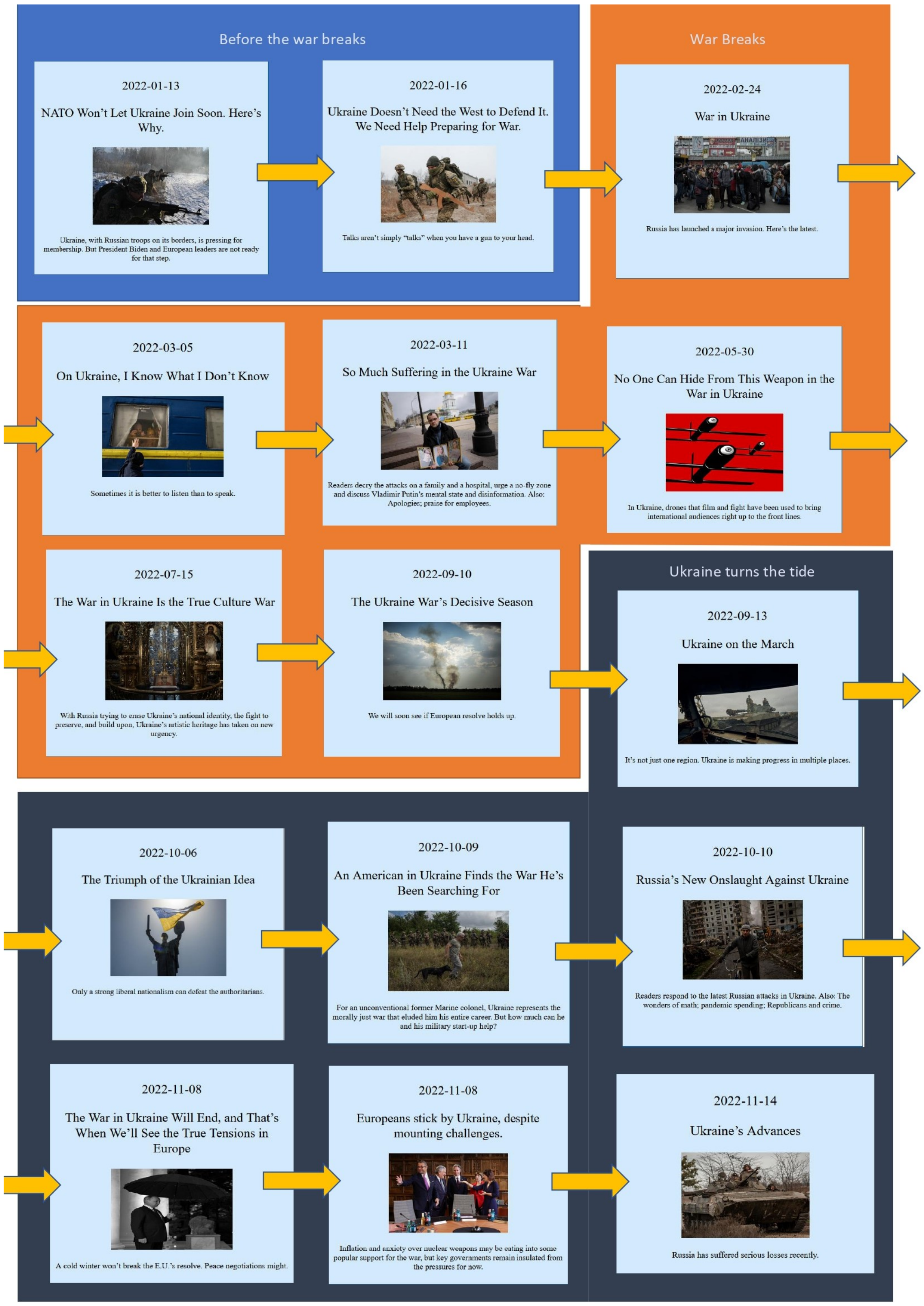
Figure 3.4: Expected evolution of the dataset until the end of the thesis

The previous plots showed that we have a great number of news samples for different topics and it's mostly up to the model's ability to make accurate representations of the data and the embedding-based search to reach relevant aggregations of news.

We saw different topics but we also know that the topics can be very diverse because we can pick specific events related to the topics. Figure 3.5 shows an edited timeline of the specific event "War Ukraine" from the topic "Ukraine" which is an occurrence from 2022. In that sequence of events, we can have an idea of the type of news we can find in our dataset and also how the sequence can capture a good scope to contextualize the user on the topic.

By the end of the thesis, we expect to have a total of 4.21 million news articles information extracted to JSON documents, indexed to OpenSearch and ready to be used in our News Search program. We also expect to find a great number of the latest news events that our model has not acquired information through training and check its capabilities of contextualizing the event and achieving accurate results. Figure 3.4 shows the expected evolution of the number of news in our dataset.

It's important to keep in mind from this point forward in this thesis the dataset will be continuously growing and certain results we obtain today might not be the same tomorrow and for that reason, it will be important to identify on which dates did we perform our evaluations.



2022-01-13

NATO Won't Let Ukraine Join Soon. Here's Why.

Ukraine, with Russian troops on its borders, is pressing for membership. But President Biden and European leaders are not ready for that step.

2022-01-16

Ukraine Doesn't Need the West to Defend It. We Need Help Preparing for War.

Talks aren't simply "talks" when you have a gun to your head.

2022-02-24

War in Ukraine

Russia has launched a major invasion. Here's the latest.

2022-03-05

On Ukraine, I Know What I Don't Know

Sometimes it is better to listen than to speak.

2022-03-11

So Much Suffering in the Ukraine War

Readers decry the attacks on a family and a hospital, urge a no-fly zone and discuss Vladimir Putin's mental state and disinformation. Also: Apologies; praise for employees.

2022-05-30

No One Can Hide From This Weapon in the War in Ukraine

In Ukraine, drones that film and fight have been used to bring international audiences right up to the front lines.

2022-07-15

The War in Ukraine Is the True Culture War

With Russia trying to erase Ukraine's national identity, the fight to preserve, and build upon, Ukraine's artistic heritage has taken on new urgency.

2022-09-10

The Ukraine War's Decisive Season

We will soon see if European resolve holds up.

2022-09-13

Ukraine on the March

It's not just one region. Ukraine is making progress in multiple places.

2022-10-06

The Triumph of the Ukrainian Idea

Only a strong liberal nationalism can defeat the authoritarians.

2022-10-09

An American in Ukraine Finds the War He's Been Searching For

For an unconventional former Marine colonel, Ukraine represents the morally just war that eluded him his entire career. But how much can he and his military start-up help?

2022-10-10

Russia's New Onslaught Against Ukraine

Readers respond to the latest Russian attacks in Ukraine. Also: The wonders of math; pandemic spending; Republicans and crime.

2022-11-08

The War in Ukraine Will End, and That's When We'll See the True Tensions in Europe

A cold winter won't break the E.U.'s resolve. Peace negotiations might.

2022-11-08

Europeans stick by Ukraine, despite mounting challenges.

Inflation and anxiety over nuclear weapons may be eating into some popular support for the war, but key governments remain insulated from the pressures for now.

2022-11-14

Ukraine's Advances

Russia has suffered serious losses recently.

Figure 3.5: Edited results for the topic "War Ukraine" that shows a possible sequence of events with news available in our dataset

4 Multimodal News On-the-fly Exploration System

This chapter will focus on explaining the different tools we created for our multimodal news on-the-fly exploration system. We will explain what features were developed for our system and how we will implement those features. We will start by analyzing the system requirements; searching different types of data; results presentation; system protocol; the different types of matching features; different types of sorting features; different types of filtering features and an automatic news suggestion feature.

4.1 Requirements

We managed to make our article collection accessible through our database and now we will explain the different functionalities we intend to develop to make the best use of the extracted information for users of the system.

The system will mainly be a way for users to search for news-related topics from the earliest to the latest events reported using two different types of data as queries: text and images. We planned to design a system that would allow users to get the news they desired but also give additional context on how and why they happened through a diversified array of news that could be witnessed through a simple interface. We will be performing an embedding-based search to achieve a diversified and accurate array of news using the approximate k-NN search algorithm [8] enabled by the k-NN indexing detailed in Section 3.4 followed by a relevant aggregation of the results. The aggregation of results could result in different structures depending on the scenario the user desires or even the topic chosen by the user.

The system will need to provide a way for users to register a topic through text or images and possibly both at the same time, resulting in a multimodal search; a section of the system should be reserved to visualize the resulting news; the system needs to account for different possible matchings such as the input text or image matching with the title, paragraph or image fields in our news article collection; the system needs to account for different ways of sorting the results such as the regular list of news, a list structured in a timeline format or even a list sorted by the latest news; the system needs to account for different filters applied to our results such as a filter that utilizes a different modality to provide more context to the search or a filter that provides news from different time intervals and finally, the system should provide a way for users to search with the best combination from all the features without having to worry about changing the settings individually.

4.2 Interface

The first step towards designing our system is the development of a simple interface that allows us to provide data in the format of text or images and that can receive the results from the search algorithm and show them using different formats.

During the dataset creation steps, we used Python to perform all the necessary operations from extraction to indexing the news in OpenSearch and the flexibility offered with libraries such as the *opensearchpy* library that allowed us to perform all the required functionalities. We continued to use Python as our main programming language to develop our system for consistency reasons and for library compatibility reasons.

We developed our system as a web application using the web framework Flask [9] compatible with Python which provides a built-in development server and a fast debugger. This framework provides us with all the tools we need to search with queries and collect the results. For the users to provide queries and view the results, we used HTML forms and lists which are compatible with the Flask web framework.

4.2.1 Data Search

In this section, we will talk about how users will provide us with different types of data for search operations. The users will need to be able to provide the data to be used as a query using the text format, the image format or even both.

We found that the best way for users to provide us with the text and image data would be with an HTML form along with the Flask-WTF library which integrates the Flask framework and the WTForms library used for forms validation and rendering. We indicate that we will be using an HTML form but instead of using the built-in HTML input types, we will use the forms provided by the *wtforms* library which are the StringField form for the text and the FileField form for the image files.

The forms available in the *wtforms* library provide simple visual indicators for users to know where they should input the data with the text form being shown as a search bar and the image form shown as a file upload button. We will also use a SelectField form to allow the user to select what sorting option he desires, one BooleanField form to choose between having results with or without an image and two IntegerRangeField forms to select the minimum and maximum dates for the time interval. Finally, we will have a SubmitField form to allow the user to confirm the search with the settings based on what forms he filled out.

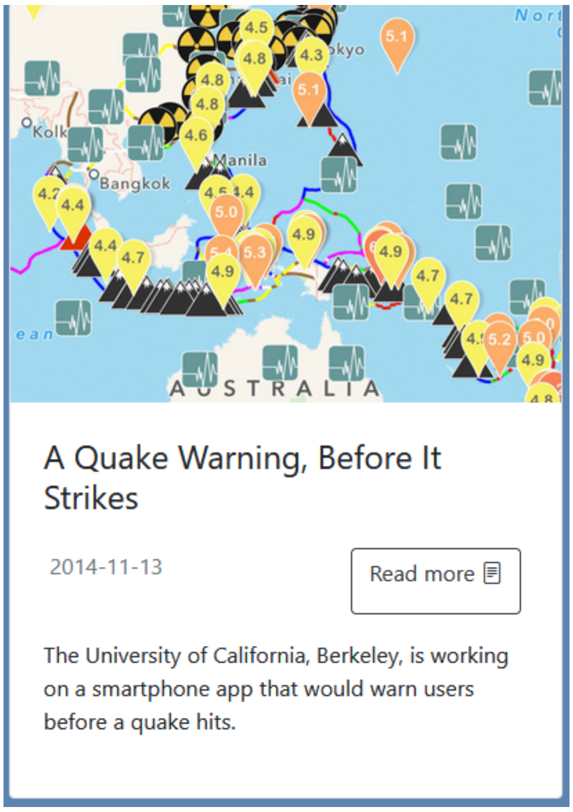
As a last step, we need to make sure that when the data is submitted everything is valid to be used as a possible query in our search algorithm. For the text, we validate the query when the number of characters is less than 200 because the model doesn't support encoding a text with more than 200 characters and for the image, we validate the query when the file provided is an image file.

4.2.2 Results Visualization

For this segment, we will explain our design choice for presenting the news in a simple list structure that would show all the relevant details for each news article.

The best format we found to present a single news article without skipping on important information was by using HTML containers where inside each container we would show the title, the publishing date of the news article, an image, the lead paragraph and a button that will redirect the user to the website of the news article. Figure 4.1 shows how the news article information is shown to the user.

After having the relevant information inside a container, we need a way to structure those containers in an efficient way to show a good amount of news but also dynamic in the sense that it will add more containers depending on the number of news articles in the results list. The results list will have a minimum size of news and a maximum when presented to the user and this size will depend on the relevant results gathered from the embedding-based search. We found out that an HTML grid would be a good way to add containers dynamically while preserving the structure of grouped-up containers.



A Quake Warning, Before It Strikes

2014-11-13

Read more

The University of California, Berkeley, is working on a smartphone app that would warn users before a quake hits.

Figure 4.1: News article presentation in our application

4.3 Matching Features

In this portion, we will start by looking into the different possible queries we can make and what types of matching can be made with the relevant information that was associated with possible matching fields during the indexing of news from Section 3.4.

For queries in our search algorithm, we can use regular text and vectors, and to perform the embedding-based search, we need to encode the validated data provided by the user using the CLIP model. The search algorithm can only perform queries using multiple forms of data by running multiple queries on the approximate k-NN search [8] because the usage of multiple forms of data on a single query is currently not supported by approximate k-NN search.

One of the indexes we created during the creation of the dataset was the regular text index which allowed us to perform searches on fields such as the title and paragraph. There is a specific feature that is only available for searches using this index which is the multi-match query setting. This setting allows using regular text as the query and uses multiple fields as possible matches of our search. In our scenario, we made use of this setting by using both the title and paragraph as our search-matching fields and that would mean that we have double the chances of finding the news article relevant because even if the title was not a good match for our search, the paragraph might be or vice-versa.

When dealing with vectors as queries we don't have the multi-match search setting which means we can only use a single field as a possible match of our search. Without being able to use multiple forms of data in our query or multiple fields as possible matches in our search, it limits the options for multimodality because we can't use both the text and the image simultaneously for our searches. We will explore multimodality as a possible filtering feature in Section 4.6.

The main matching features we will be using in our system will be title embedding matching, paragraph embedding matching and image embedding matching. These fields are associated with the text embedding index and the image embedding index created during the indexing step in the creation of the dataset.

The title embedding matching will allow a user to provide a text or an image and after validating that data and encoding it using the CLIP model, we will be able to perform a similarity-based search with the encoded data and the embedded titles that were processed and indexed during the creation of the dataset. This search will result in a list of news articles where the titles of the news share similarities with the data provided by the user. The paragraph embedding matching will be processed similarly to the title embedding matching except for the fact that we will be performing a similarity-based search with the encoded data and the embedded paragraphs instead of the embedded titles. This search will result in a list of news articles where the lead paragraphs of the news share similarities with the data provided by the user.

The image embedding matching will be processed differently from the other two. We still perform an embedding-based search with the encoded data and the embedded images but the image embedding index didn't have the news article to give as a result instead it provides a list of images. Each image in that list had a unique identifier which was the identifier from the news article with the additional number corresponding to the number of the image. We need to reach the relevant information from the news article and not just the image and for that reason, we need to perform an additional query where we provide the identifiers of the news articles by removing the additional number of the image and use them as a query to retrieve the list of news articles. This search will result in a list of news articles where the images of the news share similarities with the data provided by the user.

4.4 Similarity-based Ranking and Sorting

We've already acquired lists of news using the similarity-based search so the next feature we need to explore is in what ways should we sort those news articles to achieve relevant views for the users.

$$d(A,B) = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{4.1}$$

During the indexing step from our dataset creation in Chapter 3.4, we mentioned that we created the text and image embedding indexes with the cosine-similarity as our metric, and this metric will calculate the similarity between two vectors using the formula in Equation 4.1. This metric is used as the similarity score for our similarity-based search.

The list of news articles that resulted from the similarity-based search with text or image embeddings will be sorted by default based on the similarity score between the encoded data provided by the user and the desired matching field which means we will be getting the most relevant news based on the similarity of our vectors in the top of the list while the least relevant news will be in the bottom of the ranking. This will be one of the possible sortings in our system and will essentially order the news based on the ranking provided by the search algorithm and we will mention it as relevance sorting.

The users may want to check the latest news for a given topic. For that reason, we have a sorting feature that can sort the news from the latest to the oldest. For this feature, we mainly needed to filter out a list of results with a relevant similarity score and order them based on their date field. The scores of our embedding-based search can go from 0 to 1 and, based on experiments, we decided on a similarity threshold of 0.9 because the news that got a score equal or higher were considered relevant. If for some reason we don't have enough news to show the user because the similarity threshold is too high, we lower the threshold until we have the desired amount of news. We will mention this feature as latest sorting.

4.5 Timeline Exploration

We provided another format that makes sense to contextualize users on how the news event came to be and how it's still relevant in the present and that is the timeline format.

We mentioned how it's important to show the user the latest news for a given topic but having the news sorted from oldest to newest might not be as relevant for the user. Thus, we decided to show older news in a more relevant format that allows users to get the gist of the history regarding that topic. We will introduce to the user a timeline starting from the latest news article and finishing by showing the oldest news article that is also relevant to the topic.

After having the oldest and newest news articles we need to fill the timeline with other relevant news while keeping the order from latest to oldest. Those news articles were chosen based on the news article's date frequency while ensuring we have a good range of time intervals, meaning that time intervals where there are more news related to the topic are more likely to be shown in the timeline than interval times where there are less news.

We accomplished this by first using the same techniques from the latest sorting feature and then keeping track of a selection score that dictates if the news article is selected for the timeline or not. The selection score accumulates every time a news article is not selected and resets after a news article is selected. The amount accumulated depends on the frequency of the date from the news article in the list.

For the previous sorting features, we used the HTML grid to show the resulting news articles to the users but for a timeline structure, the grid format is not very appropriate. Consequently, we chose to design our own timeline structure using different HTML elements that would make it clear that we were presenting a timeline rather than just sorting the news by date. Part of the timeline using the timeline structure we created and using the topic of gun control as an example can be seen in Figure 4.2. We will mention this feature as timeline sorting.

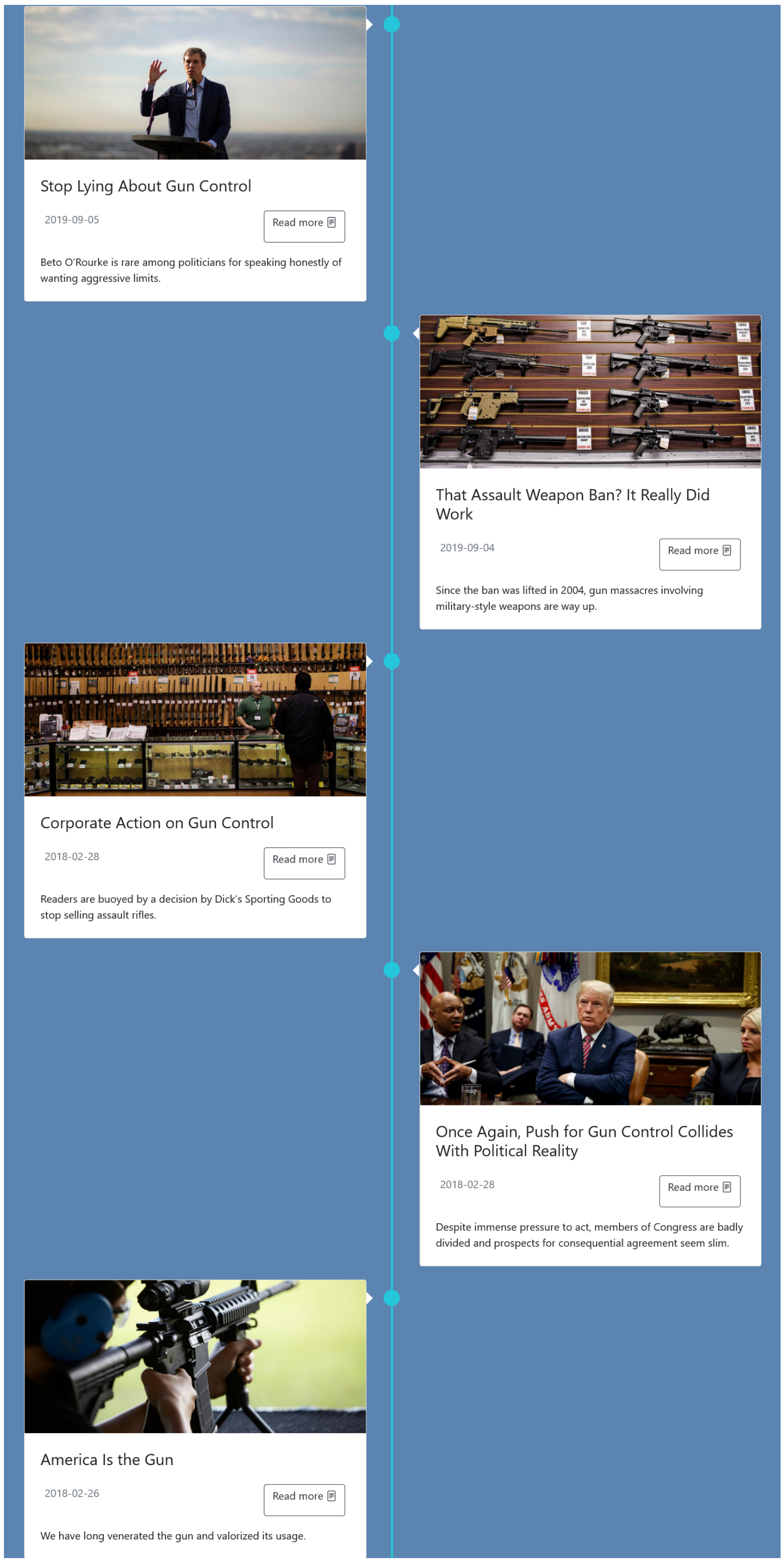
4.6 Filtering Features

We've managed to make similarity-based searches with our matching features and organize the results of those searches with our sorting features but we still haven't covered how we can improve the results from our searches by using filters with additional modalities in our data or even filters that can utilize dates to limit our embedding-based search.

These filtering features will allow for relevant news articles to stay on our final rank while removing the articles that are not relevant to the topic at hand. We will explore two different approaches for filtering: a multimodal filter and a time interval filter.

4.6.1 Multimodal Filter

The multimodality filter will be using the two different possible modalities provided by the user which are the text and the image meaning that if either the text or the image is not provided this feature cannot be implemented.



Stop Lying About Gun Control

2019-09-05

Read more

Beto O'Rourke is rare among politicians for speaking honestly of wanting aggressive limits.

That Assault Weapon Ban? It Really Did Work

2019-09-04

Read more

Since the ban was lifted in 2004, gun massacres involving military-style weapons are way up.

Corporate Action on Gun Control

2018-02-28

Read more

Readers are buoyed by a decision by Dick's Sporting Goods to stop selling assault rifles.

Once Again, Push for Gun Control Collides With Political Reality

2018-02-28

Read more

Despite immense pressure to act, members of Congress are badly divided and prospects for consequential agreement seem slim.

America Is the Gun

2018-02-26

Read more

We have long venerated the gun and valorized its usage.

Figure 4.2: Timeline Structure

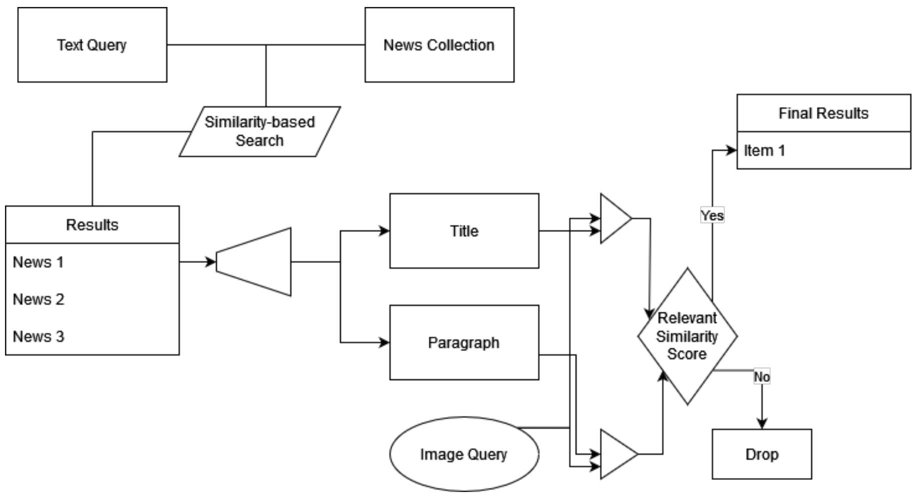


Figure 4.3: Diagram for the Multimodal Filter

This filter will be using one of the modalities to perform a query and obtain an initial rank of results that will be filtered using the other modality. The results will be acquired by using any of the matching features we explored in Section 4.3. To perform the filtering, we will be iterating through the list of results and collect the embedded title and paragraph to compare both fields to the additional encoded data using the cosine-similarity metric. After comparing both fields, at least one of those fields will need to achieve a relevant similarity score in order for that news article to remain part of the list of results. If for some reason we don't have enough news to show the user (because the similarity score is too high) we lower the score until we have the desired amount of news. Figure 4.3 shows the diagram for the process of filtering using multiple modalities.

We decided that we would use the text provided by the user to perform the initial query and the image would then serve as a filter to compare to the relevant information in each of the results to confirm if the news article would remain on the list or not. The basis for this decision lies in the evaluations done in Section 5.2.1 when using the text and image separately as a query and reaching the conclusion that the text is generally better to use as the base of our search because it was more accurate at identifying relevant news for the topic.

4.6.2 Time Interval Filter

The time interval filter will be using a time interval provided by the user which is a time interval at year granularity meaning that if the user doesn't provide a time interval using the range of years available this feature is not used.

This filter is a tool that can be used when performing embedding-based searches which is the range tool. When we implement the query using one of the different possible matching features, we will provide a setting that will ensure that the range of dates from the results is inside the time interval provided by the user. The resulting list will still be getting the results based on the similarity score but it will be limited by the specified interval.

The time interval provided by the user using a range for the minimum year and a range for the maximum needs to be somewhat relevant to that topic, otherwise, the results produced might not be as good as expected or even produce no results at all. This made us realize how we're putting too much responsibility in the hands of the user to achieve satisfactory results when there is a chance the user doesn't have enough knowledge about the topic. Therefore, we decided to make one final feature that would require minimum knowledge from our users regarding the topic they wanted to explore while also requiring minimum knowledge from the user regarding our features.

4.7 Suggestion Feature - Improve User Contextualization on Reported Events

In order to provide the best possible contextualization for a user that is not familiarized with the features we've previously mentioned or even certain topics, we developed one last feature which essentially chooses the best combination of features available in our system for the user. This involves the matching features, the sorting features and the filtering features. We called this the suggestion feature.

Instead of choosing what features they want, we will pick the features that will produce the best possible result and give that result to the user while the users only need to provide a text, an image or both and the size of the results. We start by picking the best matching feature to give us some base results and after analyzing those results, we can choose the best filtering feature to produce improved results, and finally, choose the best sorting feature based on the final results to reach a good structure of news.

4.8 Suggestion Algorithm

We've seen the general results we want to reach after having performed all the relevant features, so now we need an algorithm that reaches those results without manual interference.

For the matching feature, we can choose between the title-matching feature, the paragraph-matching feature and the image-matching feature and according to the evaluations done in Section 5.2.1, we arrived at the conclusion that the title-matching would generally be the best base for our suggestion search due to its accuracy at identifying relevant news articles for the topic presented.

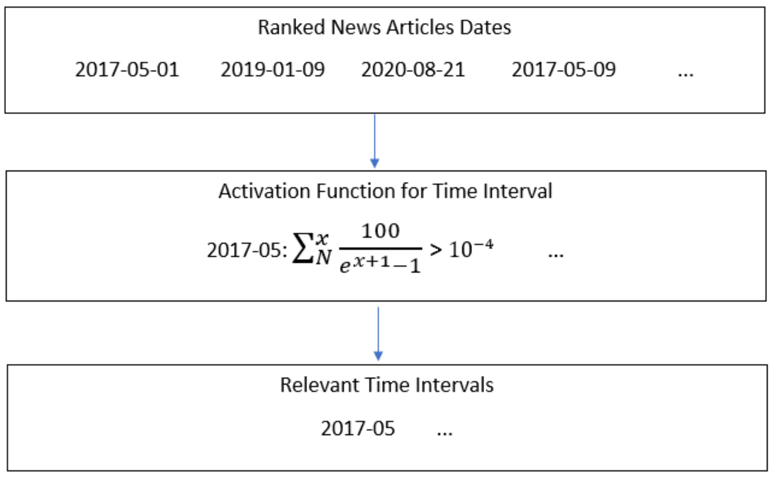

Ranked News Articles Dates
2017-05-01
2019-01-09
2020-08-21
2017-05-09
...
Activation Function for Time Interval
2017-05: $\sum_N^x \frac{100}{e^{x+1}-1} > 10^{-4}$
...
Relevant Time Intervals
2017-05
...

Figure 4.4: Determine relevant time intervals with x as the position in the ranking and N as the number of articles in that time interval

After having the base news for our search, we need to know if we will be using the multimodal filter feature or the time interval filter feature to improve the results. This was essentially decided based on whether the user introduced an image, in which case we would use the time interval filter followed by a multimodal filter or not and we would have to use only the time interval filter. The main issue with the time interval filter is that we can't know which is the most relevant time interval while using an automatic process because our list of results from the base search doesn't guarantee that the first news article has the most relevant time interval for that topic and neither does the frequency of time intervals in those results.

As a result, we decided to create an activation function that would decide if the time interval was relevant or not based on not only the frequency of time intervals but also on their position in the ranking of the base search. This algorithm was made mainly to avoid the issue of considering a time interval as relevant by having many irrelevant news articles with the same time interval in the lower rankings. Figure 4.4 shows the process of having all the time intervals going through the activation function and the resulting relevant time intervals.

We ended up with a multiple time interval filter where instead of using only one range of dates on our query as a filter, we had multiple dates that had relevant news based on the topic and that was also a filter compatible with our search algorithm.

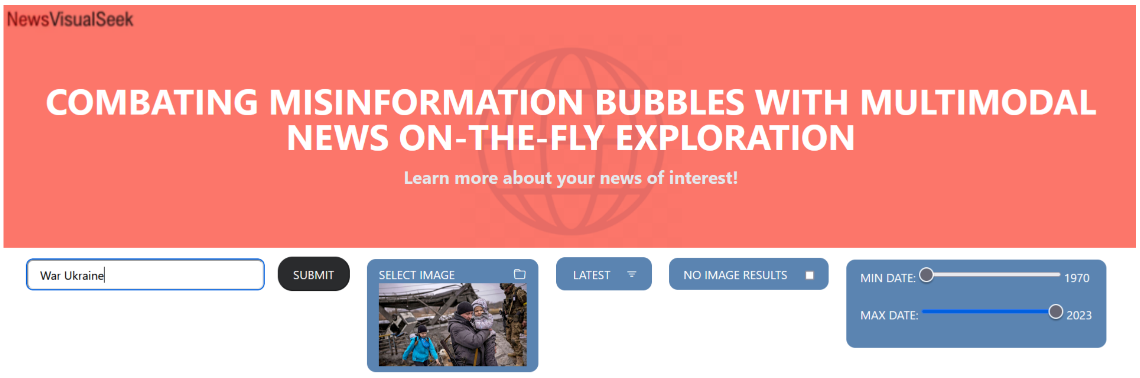

NewsVisualSeek
COMBATING MISINFORMATION BUBBLES WITH MULTIMODAL NEWS ON-THE-FLY EXPLORATION
Learn more about your news of interest!
War Ukraine
SUBMIT
SELECT IMAGE
LATEST
NO IMAGE RESULTS
MIN DATE: 1970
MAX DATE: 2023

Figure 4.5: Example of using endpoint "/" in the interface of our application

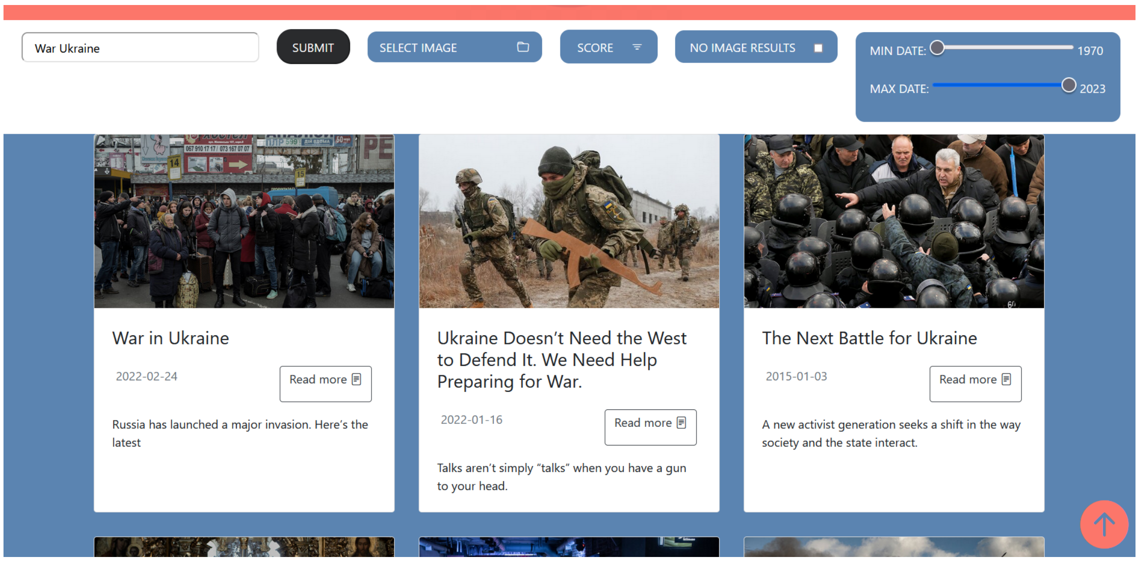
Finally, we had an improved list of results and we only needed to decide between showing the news using the relevance sorting feature, the latest sorting feature or the timeline sorting feature. The regular view would be the relevance sorted list of news and we decided that we would change the view to a timeline format if the number and range of the relevant time intervals were to be significant to the point where the topic had a relevant sequence of events that would favor the use of the timeline structure over the relevance sorting. The latest sorting feature would then stay as an optional visualization of the results where the user wants to view the results organized from newest to oldest.

4.9 System Protocol

We considered it important for the user to get the best possible search experience when it comes to learning about a certain topic. Thus, we can make an overview of all the information needed in order to use the system and all its features in a knowledgeable manner.

The system works relatively similarly to a web application with different endpoints, different forms to enter data and a view for the list of results. The different endpoints are associated with different features and there are 4 endpoints in total. The forms and view of results are essential parameters that will be available in all the endpoints.

The endpoints are "/title" to use the title matching feature, "/paragraph" to use the paragraph matching feature, "/image" to use the image matching feature and "/" to use the suggestion feature. Figure 4.5 shows the interface from the endpoint "/", which visually is equivalent to the other endpoints, and Figure 4.6 shows the results from a query.



War in Ukraine

2022-02-24

Russia has launched a major invasion. Here's the latest

Ukraine Doesn't Need the West to Defend It. We Need Help Preparing for War.

2022-01-16

Talks aren't simply "talks" when you have a gun to your head.

The Next Battle for Ukraine

2015-01-03

A new activist generation seeks a shift in the way society and the state interact.

Figure 4.6: Example of results using a query in the interface of our application

The system interface consists of 2 different types of possible forms whose data will be fed directly to our queries: a search bar used for text queries and a file form used for image queries which only allows the use of images of file type JPG. The system will be feeding encoded text to the query if the search bar is filled with text and the file form does not possess a file, feed an encoded image to the query if the search bar is empty and the file form possesses a JPG image and use the multimodal filter feature if both the search bar is filled with text and the file form possesses a JPG image. The number of results shown is dynamic and based on the number of relevant articles. We can select between the three sorting features which are the relevance sorting feature, latest sorting feature (default) and the timeline sorting feature and toggle between results with image (default) and without image. Lastly, we can select a minimum year and a maximum year using the two range filters to use the time interval filter. The resulting list of results shows the publishing date of the article, the title, an image, the lead paragraph of the articles and a button that redirects the user to the full article. If the resulting list is empty it means the search did not have relevant results given the query and also some results will not show an image if the article didn't have any images and the user used the "No images result" toggle.

Thus, we have concluded the guide for users to explore the system and try out the features that they desire. The system was changed and reinvented a different number of times based on evaluations we did of the results put together from different types of queries and that's exactly what we are going to analyze and evaluate in the next chapter.

5 Evaluation

This chapter will focus on evaluating the similarity-based search and the different features we created for our news search system. We will evaluate the searches in different scenarios while using different features and comment on whether the results reached significant accuracy scores or if the results performed below expectations. We will start by explaining the settings of our evaluation; splitting the dataset; evaluation scenarios; assessing relevance; initial results using manual assessment; initial results using automatic assessment; exploring an in-depth use scenario; recommendation feature results and a final comparison of the different relevant features.

5.1 Methodology

After developing all the features available in our system, we need to test how accurate they are at giving relevant news to the user based on the topic. We need to specify some settings regarding our evaluation such as what part of the dataset will be considered for evaluation, in what scenarios should we evaluate the different structures and how we identify a news article as being relevant.

5.1.1 Splitting Dataset

To evaluate our system we will first divide our dataset into a training split and a test split. The training split will have documents with concepts and context acquired through the training of the CLIP model which ranges from the beginning of NYTimes news until a specific date and the test split will have documents with concepts and context that weren't acquired through the training of CLIP model which ranges from that specific date until the current date.

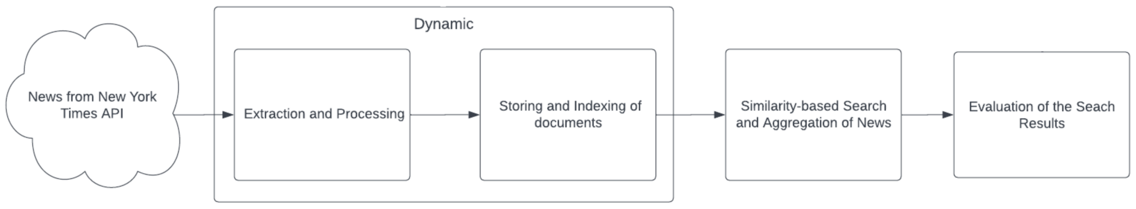

Dynamic
News from New York Times API
Extraction and Processing
Storing and Indexing of documents
Similarity-based Search and Aggregation of News
Evaluation of the Seach Results

Figure 5.1: System Pipeline

The specified date will be obtained by taking a look at the sources used for the creation of the CLIP dataset to establish what concepts were acquired. Unfortunately, because the sources were not disclosed we will estimate the specified date using the date of CLIP's publication which is 2021.

We can now specify the number of documents for each split: The training split will have documents from 1970 to 2020 resulting in a total of 4.1M documents. The test split will have documents from 2021 to the present resulting in a total of 150K documents (calculated in December 2022).

5.1.2 Scenarios

For this section, we will analyze three different relevant scenarios with the different features we implemented:

a) Topic-based queries ordered by relevance - In this scenario we use a topic as a query and order the resulting news using the relevance sorting feature (sorted by similarity score). The topic can be depicted through text, image or both. For this scenario, we will use as a metric the average precision on the ranking because we want the rank to influence the accuracy of the results.

b) Topic-based queries ordered in timeline format - In this scenario we use a topic as a query and order the resulting news using the timeline sorting feature. The topic can be depicted through text, image or both. For this scenario, we will use as a metric the precision on the results since it's sorted in a timeline fashion, the ranking of predictions won't matter in this setting.

c) News-based queries ordered by relevance - In this scenario we use a news article as our query and order the resulting news using the relevance sorting feature (sorted by similarity score). The article can be depicted through its title or paragraph and depending on what field we choose, we will use the other as a matching for search which means if we choose the title as the query we will use the paragraph matching feature and vice-versa. For this scenario, we will use as a metric the Recall@5,10,50,100 because we want to see if the article can be found in the top-k ranked documents.

5.1.3 Assessing Relevance

Our system will have 100 candidate documents for each query with the exception of the multimodality filter feature where we need 200 candidate documents in case some news articles are dropped because the image didn't deem them relevant. We will assess relevance in two different ways: manually and automatically.

Manual Assessment - We manually assess 18 documents from the 100 candidate documents because it wouldn't be practical to assess the relevancy of 100 news manually. After obtaining the results in our interface we can start checking if each news article is relevant based on the topic. This is done by checking keywords on the title and paragraphs and also the context which requires some general knowledge of the selected topics. This assessment will be performed on the topic-based queries ordered by relevance scenario and topic-based queries ordered in timeline format (using the AP and the P as metrics, respectively) because the news-based queries ordered by relevance scenario can check if a specific news piece is among the candidate documents automatically.

Automatic Assessment - We will maintain the 100 candidate documents. The topic, in this case, will be a keyword associated with a group of news and we will match it to the title or the paragraph. After obtaining the results we will check for each article's keywords and check if the keyword used for the search is there or not. This assessment will be performed only on the topic-based queries ordered by relevance scenario and news-based queries ordered by relevance scenario (using the AP and the R as metrics, respectively) because to evaluate the topic-based queries ordered in timeline format we need to take into consideration the structure of the results. For images, we can only assess their relevancy automatically by associating specific keywords from news articles to them and checking for matching keywords in the resulting news and that evaluation is made in Section 5.3.

5.2 Initial Comparison of the System's Features

Following the scenarios and settings we defined, we will start to make evaluations on the features we created and the features we chose for our initial evaluations were the following:

1. **N**on-embedded text matching which matches the topic as a text query with the title and paragraph fields simultaneously.
2. **E**mbedded title matching which uses the topic as a text query and the title matching feature from Section 4.3.
3. **E**mbedded paragraph matching which uses the topic as a text query and the paragraph matching feature from Section 4.3.

4. **E**mbedded image matching which uses the topic as an image query and the image matching feature from Section 4.3.

5. **E**mbedded title and image matching which uses the topic as a text and image query and the multimodal filter feature from Section 4.6 with the title matching feature.

6. **E**mbedded paragraph and image matching which uses the topic as a text and image query and the multimodal filter feature from Section 4.6 with the paragraph matching feature.

In these evaluations, we will be taking a look at five different topics that have been looked upon in Section 3.7 which we consider to be relevant topics to study for either their continued broadcast by the media outlets throughout the years or news from that topic had a sudden spike of interest in recent years. These topics are Racial Justice, War Ukraine, Elections, Covid-19 and Trump.

5.2.1 Results using Manual Assessment

Table 5.1: Results using the entire dataset in Scenario (a) using average precision@18 as metric with 5 different topics on 9th November

	Racial Justice	**War Ukraine**	**Elections**	**Covid**	**Trump**
Non-embedded text	1.00	0.95	1.00	1.00	1.00
Embedded Title	1.00	0.93	1.00	0.99	1.00
Embedded Paragraph	1.00	0.95	1.00	0.77	0.68
Embedded Image	0.76	0.89	0.98	0.52	0.89
Embedded T. + Embedded I.	1.00	1.00	1.00	1.00	1.00
Embedded P. + Embedded I.	1.00	1.00	1.00	0.99	1.00

We will start with the results of the manual assessment setting and with one of the scenarios associated with the setting which is Scenario (a) while using the entire dataset in Table 5.1 and analyze the results.

In Non-embedded text, we saw 3 unrelated news for the War Ukraine topic. In Embedded Title, we saw 2 unrelated news for the War Ukraine topic and we saw 1 unrelated article for the Covid topic. In Embedded Paragraph, we saw 4 unrelated news for War Ukraine topic, 8 unrelated articles for Covid and 4 news related to presidents other than Trump. In Embedded Image, we saw 10 related to protests other than racial justice, 2 unrelated news for War Ukraine, 1 unrelated to Elections, 10 related to diseases other than Covid and 4 unrelated news to Trump. In Embedded T. + Embedded I., all the topics show 18 relevant news. In Embedded P. + Embedded I., we saw 1 unrelated news for Covid topic.

Table 5.2 shows the results in the same scenario but with the dataset reduced to the test split for the possible matches and without Non-Embedded text.

Table 5.2: Results using test split in Scenario (a) using average precision@18 as metric with 5 different topics on 15th November

	Racial Justice	War Ukraine	Elections	Covid	Trump
Embedded Title	1.00	1.00	1.00	1.00	0.99
Embedded Paragraph	0.93	1.00	1.00	0.99	0.78
Embedded Image	0.45	0.85	0.92	0.79	0.83
Embedded T. + Embedded I.	1.00	1.00	1.00	1.00	0.97
Embedded P. + Embedded I.	0.96	1.00	1.00	0.98	0.96

Table 5.3: Results using the entire dataset in Scenario (b) using precision@18 as metric with 5 different topics on 9th November

	Racial Justice	War Ukraine	Elections	Covid	Trump
Non-embedded text	0.94	1.00	1.00	1.00	1.00
Embedded Title	1.00	0.83	1.00	0.83	0.94
Embedded Paragraph	0.83	0.78	0.72	0.83	0.89
Embedded Image	0.61	0.72	0.94	0.61	0.78
Embedded T. + Embedded I.	1.00	1.00	1.00	0.94	1.00
Embedded P. + Embedded I.	1.00	1.00	1.00	0.94	1.00

After the first results, we could already see how the searches using the different features and different topics will result in better or worse scores. Apart from Non-Embedded text, which is a feature that only serves for testing and doesn't affect the study of the thesis whose focus is on similarity-based searches using embeddings, we can see how the multimodality filter feature shows signs of improvement when its complementing the title and paragraph matching features. The scores show how the image matching feature can be seen as the worse matching feature followed by the paragraph matching and finally the title matching.

The scenario that follows for manual assessment is Scenario (b) while using the entire dataset in Table 5.3 and here's the analysis of the results:

In Non-Embedded text, we saw 1 news related to general justice instead of racial justice. In Embedded Title, we saw 3 unrelated news for War Ukraine, 3 unrelated articles for Covid and 1 unrelated article for Trump. In Embedded Paragraph, we saw 3 unrelated articles for racial justice, 4 news related to other wars other than War Ukraine, 5 related to education instead of elections, 3 unrelated articles for Covid and 2 unrelated articles for Trump. In Embedded Image, we saw 7 related to protests other than racial justice, 5 unrelated news for War Ukraine, 1 unrelated news for elections, 7 related to diseases other than Covid and 4 unrelated news for Trump. In Embedded T. + Embedded I., we saw 1 unrelated article for Covid. In Embedded P. + Embedded I., we saw 1 unrelated article for Covid.

Table 5.4 shows the results in the same scenario but with the dataset reduced to the test split for the possible matches and without Non-Embedded Text.

Table 5.4: Results using test split in Scenario (b) using precision@18 as metric with 5 different topics on 15th November

	Racial Justice	**War Ukraine**	**Elections**	**Covid**	**Trump**
Embedded Title	0.61	1.00	0.94	1.00	0.94
Embedded Paragraph	0.72	1.00	0.94	0.89	0.44
Embedded Image	0.11	0.50	0.83	0.33	0.28
Embedded T. + Embedded I.	0.94	1.00	1.00	1.00	1.00
Embedded P. + Embedded I.	0.78	1.00	1.00	0.56	0.50

Similarly to the topic-based queries ordered by relevance, the results showed improvements when using the multimodality filter and image matching got the worst scores followed by paragraph matching and finally title matching. Even though we use different metrics to calculate both scenarios, we can see the matching features performing slightly better without the timeline sorting feature mostly because the relevance sorting feature is organizing the news based on relevance whereas the timeline feature is focusing on giving a sequence of events for the topic. Two examples of timeline structures can be seen in Figure 5.4.

5.2.2 Results using Automatic Assessment

Table 5.5: Results with text embeddings matching in Scenario (a) using average precision@100 as metric with 5 different topics on 10th November

	Racial Justice	**War Ukraine**	**Elections**	**Covid**	**Trump**
Embedded Title	0.48	0.90	0.58	0.83	0.91
Embedded Paragraph	0.34	0.74	0.68	0.71	0.82

The next results will be performed using an automatic assessment setting. The first scenario we will see is Scenario (a) but in this case, we will consider Embedded Title and Embedded Paragraph only, because we use a specific keyword available in the news that depicted the topic as our text query and that can't be done for images. The keywords were selected based on the relevance of that keyword according to the topic and based on the presence of the keyword in news related to the topic. We used the entire dataset in Table 5.5.

The keywords selected were the following:

1. "Discrimination" using Embedded Title we got 23 relevant news out of 100 and using Embedded Paragraph we got 6 relevant news out of 100.

2. "Putin, Vladimir V" using Embedded Title we got 93 relevant news out of 100 and using Embedded Paragraph we got 78 relevant news out of 100.

3. "Democratic Party" using Embedded Title we got 47 relevant news out of 100 and using Embedded Paragraph we got 54 relevant news out of 100.

Table 5.6: Results using test split in Scenario (a) using average precision@100 as metric with 5 different topics on 16th November

	Racial Justice	War Ukraine	Elections	Covid	Trump
Embedded Title	0.11	0.90	0.57	0.54	0.64
Embedded Paragraph	0.05	0.53	0.49	0.58	0.55

Table 5.7: Results with text embeddings matching in Scenario (c) using an average from Recall@5,10,50,100 as metric with 10 different queries for 5 different topics on 10th November

	Racial Justice	War Ukraine	Elections	Covid	Trump
Recall@5	0.10	0.10	0.00	0.20	0.40
Recall@10	0.10	0.10	0.20	0.20	0.40
Recall@50	0.50	0.30	0.30	0.40	0.50
Recall@100	0.50	0.30	0.40	0.40	0.50

4. "Coronavirus (2019-nCoV)" using Embedded Title we got 30 relevant news out of 100 and using Embedded Paragraph we got 37 relevant news out of 100.

5. "Trump, Donald J" using Embedded Title we got 77 relevant news out of 100 and using Embedded Paragraph we got 70 relevant news out of 100.

Table 5.6 shows the results in the same scenario but with the dataset reduced to the test split for the possible matches.

These results served to demonstrate that, when the size of candidate documents goes from 18 to 100 and the assessment of relevance for a news article is done only based on the existence of a keyword in its information, despite not being the most representative, is the most objective and efficient way to determine if a news article is relevant in an automatic setting with a large number of results. We also perceived that as the number of news retrieved on our searches increased, their relevancy dropped.

The final scenario we will be taking a look at for automatic assessment is Scenario (c) while using the entire dataset in Table 5.7 and analyzing its results.

This scenario is different from the rest because it uses 10 queries for each topic and those queries are either titles or paragraphs from the news that will be encoded and used with the paragraph or title matching feature respectively. For Recall@5 with a score of 0.1 as an example, it means that out of the 10 news whose title or paragraph was used as queries only 1 was seen in the top 5 news in the ranking.

We saw that even with 100 candidate documents to find the 10 news used as queries, it managed to find 5 or fewer news for all the topics. We can obtain almost the same results with 50 candidate documents.

Table 5.8: Results using test split in Scenario (c) using an average from Recall@5,10,50,100 as metric with 10 different queries for 5 different topics on 16th November

	Racial Justice	**War Ukraine**	**Elections**	**Covid**	**Trump**
Recall@5	0.00	0.30	0.00	0.40	0.00
Recall@10	0.10	0.40	0.00	0.40	0.10
Recall@50	0.40	0.60	0.10	0.60	0.10
Recall@100	0.50	0.60	0.20	0.60	0.10

Table 5.8 shows the results in the same scenario but with the dataset reduced to the test split for the possible matches and much like in previous scenarios, we can see for the topics of War Ukraine and Covid the scores improved because the candidate documents were picked from ranges of dates (2021 and 2022) that favored those topics and even without the knowledge from the CLIP model for these events, its diversified training allowed the similarity-based search to find similarities between the queries and the news articles.

5.3 Final Comparison of the System's Relevant Features

So far we've experimented with our system's different features with 5 different topics in different scenarios using manual and automatic assessment.

Looking at this final evaluation, we have yet to put all of the features created that we consider to be relevant according to previous results or general relevancy for this thesis and check how they perform in a scenario with a large number of different types of queries for different topics that occurred in different dates. This evaluation will be important to determine if the results seen so far have been biased toward certain queries and topics or if they were representative enough for a large portion of the topics available in our dataset.

5.3.1 Setting up Evaluation

This evaluation will be taking in a total of 50 queries that will explore 18 different topics in various time intervals that will take into account news that the model has more and less knowledge of and also explore different types of queries within the same topic such as **a)** simple queries whose text is composed of 3 or fewer words, **b)** complex queries whose text is composed of 4 or more words and **c)** diverse queries that explore themes associated with the topic. We will use 100 candidate documents for each query and determine how many of those are relevant.

We will be using the entire dataset given there are news articles from 2013 up until 2022 with the results extracted on 4th January 2023 and we will be using automatic assessment for a more objective evaluation using the keywords provided in Table 5.9. The sorting feature used will be the relevance sorting feature because we won't be paying attention to the visualization of the results. The considered features will be:

Relevance - The title matching feature showed the best results when compared to the other matching features and we will refer to this as *Relevance*.

Image - The image matching feature because it's an important feature to understand the limitations of matching text with images despite the weaker results we saw previously and we will refer to this as *Image*.

Multimodal - The title matching feature with multimodal filtering feature showed improvements in our results throughout the different evaluations we've seen thus far and we will refer to this as *Multimodal*.

Suggested - The suggestion feature that will use the best features includes trying to find relevant time intervals to improve the amount of relevant news found and we will refer to this as *Suggested*.

Specific - A bonus feature is the time interval filtering feature which requires the knowledge of the user for what time interval is relevant for the topic and we will refer to this as *Specific*.

We will be using different metrics to evaluate the results and the scores computed will be the average of the metrics for all the queries:

Recall@10,20,50,100 - We will check how many news articles are relevant given the query about a topic in all the relevant documents in our dataset or more specifically, we count the number of news with relevant keywords in the top-K documents and divided it by all the news with relevant keywords in the entire dataset.

Precision@10,20,50,100 - We will check how many of the 100 candidate documents were news that contained relevant keywords in the top-K.

MeanAveragePrecision@100 - We determined how many of the 100 candidate documents were news that contained relevant keywords but also how the relevant documents were positioned in the ranking.

Table 5.9: All the queries and keywords used for the final evaluation

Year	Query	Keyword(s)/Key phrases
2013	Edward Snowden releases classified documents	Surveillance of Citizens by Government
2014	Malaysian airline lost jet	Malaysia Airlines Flight 370
2014	Search for Malaysia Airlines missing jet	Malaysia Airlines Flight 370
2014	Families of missing frustration over missing jet	Malaysia Airlines Flight 370
2015	Mayweather Pacquiao Fight	Boxing/Mayweather, Floyd Jr/Pacquiao, Manny
2017	Terrorist attack Manchester	Manchester Arena Bombing (May, 2017)
2017	Ariana concert ends with terrorist attack	Manchester Arena Bombing (May, 2017)
2017	Police investigation on manchester bomber	Manchester Arena Bombing (May, 2017)
2017	Car rams into crowd of protesters Charlottesvile	Charlottesville, Va, Violence (August, 2017)
2017	White Nationalist March ends in death	Charlottesville, Va, Violence (August, 2017)
2017	Trump criticized by remarks on Charlottesville	Charlottesville, Va, Violence (August, 2017)
2018	Parkland Shooting	Parkland, Fla, Shooting (2018)
2018	Florida School Shooting suspect arrested	Parkland, Fla, Shooting (2018)
2018	Victims frustrated over lack of gun control	Parkland, Fla, Shooting (2018)
2018	Thailand cave Rescue	Chiang Rai Province (Thailand) / Tham Luang Nang Non Cave (Thailand)
2018	Rescuers save team in difficult thai cave	Chiang Rai Province (Thailand) / Tham Luang Nang Non Cave (Thailand)
2018	Thai boys condition after the cave rescue	Chiang Rai Province (Thailand) / Tham Luang Nang Non Cave (Thailand)
2019	Notre Dame Cathedral Fire	Notre Dame Cathedral (Paris, France)
2019	Aftermath of Notre Dame Fire	Notre Dame Cathedral (Paris, France)
2019	France focuses on rebuilding Notre Dame	Notre Dame Cathedral (Paris, France)
2019	Trump Impeachment	Trump-Ukraine Whistle-blower Complaint and Impeachment Inquiry
2019	Report on the Impeachment Inquiry	Trump-Ukraine Whistle-blower Complaint and Impeachment Inquiry
2019	Records show Giuliani pressure on Ukraine	Trump-Ukraine Whistle-blower Complaint and Impeachment Inquiry
2021	Covid	Coronavirus (2019-nCoV)
2019	Brexit	Great Britain Withdrawal from EU (Brexit)
2020	Presidential Election of 2020	Presidential Election of 2020
2021	Covid cases reach new peak	Coronavirus (2019-nCoV)
2019	Brexit will affect Europe	Great Britain Withdrawal from EU (Brexit)
2019	Brexit splits Britain	Great Britain Withdrawal from EU (Brexit)
2020	Trump and Biden Debate	Presidential Election of 2020
2020	Mail voting causes doubt during elections	Presidential Election of 2020
2021	India Covid-19 Crisis	Coronavirus (2019-nCoV)
2021	Storming of the US capitol	Storming of the US Capitol (Jan, 2021)
2021	Trump incites rioters to storm capitol	Storming of the US Capitol (Jan, 2021)
2021	Trump ends his era with violence	Storming of the US Capitol (Jan, 2021)
2022	War Ukraine	Russian Invasion of Ukraine (2022)
2022	COP27	United Nations Framework Convention on Climate Change
2022	World Cup 2022	World Cup 2022 (Soccer)
2022	Hurricane Ian	Hurricane Ian (2022)
2022	Midterm Elections 2022	Midterm Elections (2022)
2022	Military aid sent to help defend Ukraine	Russian Invasion of Ukraine (2022)
2022	Putin threats Nuclear War	Russian Invasion of Ukraine (2022)
2022	Richer countries pay for climate damage	United Nations Framework Convention on Climate Change
2022	Climate goals and promises are falling short	United Nations Framework Convention on Climate Change
2022	U.S. Team for the world cup 2022	World Cup 2022 (Soccer)
2022	Critics to Qatar and Fifa during world cup	World Cup 2022 (Soccer)
2022	Florida braces for Hurricane Ian	Hurricane Ian (2022)
2022	Destruction after Hurricane Ian	Hurricane Ian (2022)
2022	Democratic Party to keep the majority	Midterm Elections (2022)
2022	Trump and the Republican Party	Midterm Elections (2022)

5.3.2 Analyzing Diversity

Table 5.10: Average diversity evaluation for all 50 queries

	Time Diversity Relevant	Time Diversity All	Theme Diversity Relevant	Theme Diversity All
Relevance	10.5 ± 5	72.0 ± 8	5.7 ± 2	62.5 ± 7
Image	4.7 ± 2	26.7 ± 2	5.3 ± 2	83.1 ± 10
Multimodal	12.4 ± 6	72.1 ± 8	5.7 ± 2	62.6 ± 7
Suggested	5.3 ± 3	8.0 ± 2	6.0 ± 2	37.1 ± 9

Before proceeding to the analysis of the results for the final comparison of the features, we decided to analyze how the different features compare to each other when it comes to the diversity of results.

The diversity of results is associated with the number of different themes and time intervals that can be depicted in our candidate documents. For that reason, we can separate the diversity in time and theme diversity. The study of diversity told us with precision how much each feature is sacrificing in terms of the topic and temporal context in order to improve the accuracy of the results. Therefore, we need to analyze if the trade-off is worth it.

For this comparison, we took a look at each test's time diversity by counting the number of unique time intervals for relevant news articles (Time Diversity Relevant) and also counting the number of unique time intervals for all the candidate documents (Time Diversity All). Secondly, we checked each test's theme diversity by counting the number of unique keywords associated with relevant news articles (Theme Diversity Relevant) and also counting the number of unique keywords for all the candidate documents (Theme Diversity All). Table 5.10 shows the average of the 50 queries for the calculations mentioned before followed by the standard deviation.

The results demonstrated that there were divergences between the different tests with the Relevance test showing that from the 72 time intervals found in the candidate documents, only 10 contained relevant news to our query and for the 62 unique keywords found on our candidate documents only 5 were relevant to our query. The Image test only had 26 time intervals because the time intervals in this test started in 2006 when images became available with 4 considered relevant and 83 unique keywords with 5 considered relevant which we speculated were because of less relevant news in the candidate documents. The Multimodal test showed similar results to the Relevance test except for the relevant time intervals where it was able to find 12 relevant time intervals perhaps because the multimodal filter removed irrelevant time intervals and added relevant ones. The Suggested test picks what it thinks are the relevant time intervals which were 8, but only 5 were actually relevant and that was 7 less relevant time intervals compared to the Multimodal test. Despite that, looking at the theme diversity we can see 37 unique keywords where 6 are relevant which we speculated and later on confirmed that it directly correlated to more relevant news in the candidate documents given the query.

Table 5.11: Average diversity evaluation for all 50 queries on the Specific test

	Time Diversity Relevant	Time Diversity All	Theme Diversity Relevant	Theme Diversity All
Specific	1.0	1.0	5.8 ± 2	32.4 ± 5

The Specific test results can be seen in Table 5.11 which shows the 1 relevant time interval chosen by us and the 32 unique keywords where 5 are relevant for the query and even though this test shows promising results it's not as relevant because it requires very specific knowledge on the user of what date is the most relevant for the topic.

5.3.3 Analyzing the System's Features

Table 5.12: Average Recall@K for all 50 queries with K = 10,20,50,100

	Recall@10	Recall@20	Recall@50	Recall@100
Relevance	0.034	0.052	0.066	0.075
Image	0.020	0.024	0.033	0.040
Multimodal	0.034	0.053	0.067	0.08
Suggested	0.063	0.103	0.185	0.228

We started to analyze the results with the Recall@K metric using the different features, which can be checked in Table 5.12. We can see how the Image test continues to underperform compared to the other features and where the Recall@100 compared to the Recall@10 doubles the score. The Relevance test follows the Image test when it comes to performance and also more than doubles the score when comparing the Recall@10 to the Recall@100. The Multimodal test showed improvements compared to the Relevance test but not to the same extent as the previous evaluations with the Recall@10 having the same score for both and Recall@100 with a slightly higher score. The Suggested test showed the biggest improvements with Recall@10 and Recall@20 improving their score almost twice and the score of Recall@50 and Recall@100 improving almost thrice when compared to the Multimodal test. This result suggested how the time interval filter was having the most impact in picking relevant news articles for our candidate documents.

Table 5.13: Average Precision@K for all 50 queries with K = 10,20,50,100

	Precision@10	Precision@20	Precision@50	Precision@100
Relevance	0.047	0.079	0.129	0.160
Image	0.028	0.048	0.086	0.129
Multimodal	0.047	0.080	0.130	0.193
Suggested	0.050	0.100	0.222	0.364

We followed up with the results of the Precision@K metric in Table 5.13 which compared to the Recall@K, uses the number of candidate documents (100) consistently throughout the different topics and tests to assess relevancy instead of using the number of relevant documents in the dataset which varies with each topic and test. The Image test remains with the lowest score, the Multimodal test continues to improve slightly when compared to the Relevance test with the Precision@100 showing the only notable improvement and the Suggested test showed a progressively higher score from the Precision@10 to the Precision@100 which indicates the most consistent results considering the suggestion algorithm from Section 4.7. The suggestion algorithm uses the initial results obtained by the Relevance test to capture the relevant dates and uses them as a filter to improve the results that follow and that can be seen with the Precision@10 showing only a slight improvement when compared to the Multimodal test but the results that follow such as the Precision@100 can see an improvement of almost twice the score.

Table 5.14: Average MeanAveragePrecision@100 for all 50 queries

	Mean Average Precision@100
Relevance	0.472
Image	0.346
Multimodal	0.477
Suggested	0.506

The last results obtained were acquired using as metric the MeanAveragePrecision@100 in Table 5.14. This metric is important to not only understand the number of relevant documents in the candidate documents as seen with Precision@K but also how well the relevant documents are positioned in the ranking given that we needed the relevant documents to outdo the irrelevant news in the ranking. We perceived that when it came to the performance of the tests we created, its similar to the performances registered in the Precision@K with the Image test achieving the lowest score and the Multimodal test slightly improving the score compared to the Relevance test. The Suggested test obtained a better score but not to the same extent as the one observed with the Precision@100 when compared to the Multimodal test and that's because the improvement of results was observed for lower positions in the ranking which means lesser improvements considering this metric.

Table 5.15: Average results using the different metrics for all 50 queries on the Specific test

	@10	**@20**	**@50**	**@100**
Recall	0.088	0.152	0.264	0.327
Precision	0.070	0.130	0.256	0.366
MeanAveragePrecision	-	-	-	0.667

We decided to analyze the Specific test in the end because despite showing the best results for the different metrics in Table 5.15, it requires unintended knowledge from the user since to perform this test we had to make some investigation ourselves on what was the most relevant time interval for each given topic and we couldn't reach that specific interval consistently using our system. The Specific test would only have the news articles specific to 1 guaranteed relevant time interval to reach the candidate documents making it more likely to obtain relevant news articles. The results using the Specific test for

Precision@100 showed only a slight improvement when compared with the Suggested test because of the lack of time diversity when most topics have more than 1 relevant time interval as seen in Table 5.10 and the MeanAveragePrecision@100 showed a big improvement because it's not influenced by the initial results of the Relevance test unlike the Suggested test and because of that was able to identify more relevant news in the top of the ranking.

5.4 Use Case Scenarios

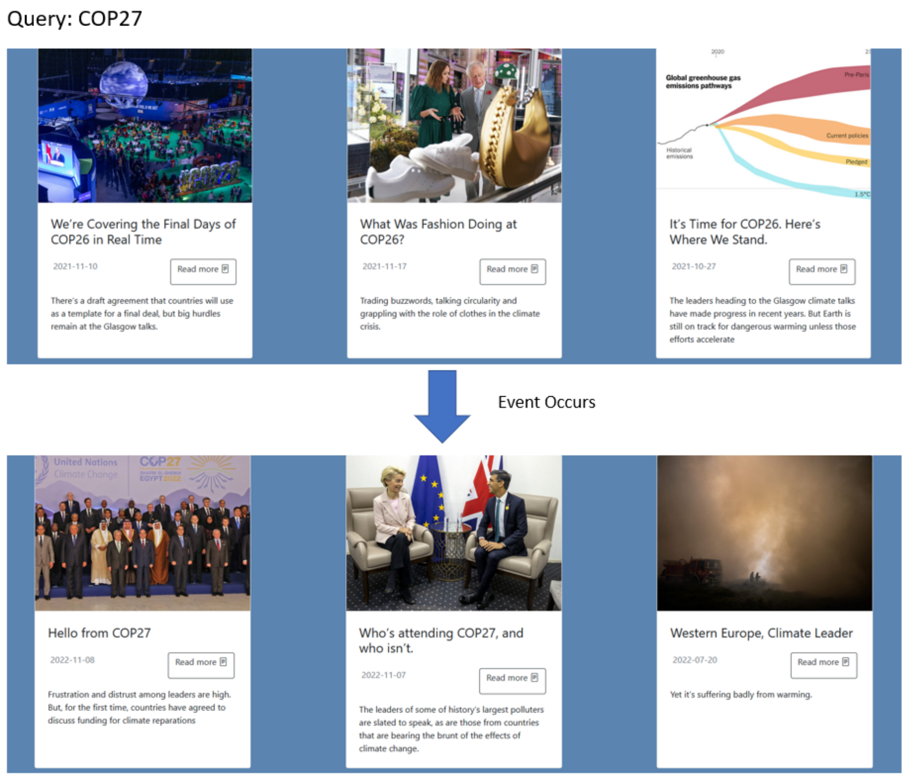
The previous experiments allowed us to test out all of our features. Specifically, all the matching features, the timeline sorting feature, the multimodal filtering feature and the suggestion feature in different scenarios. For the next experiments, we will be focused on evaluating the system in specific use case scenarios for recent topics using the best features we've found so far, while using manual assessment to check if the 18 results presented in the system's interface are relevant or not.

5.4.1 COP27 - An In-Depth Use Case Scenario

For the first use case scenario, we will take a look in depth at a topic while assuming the user has knowledge about the features of our system and that topic will be the event of COP27.

COP27 is a conference that involves the participation of countries from around the world with the objective of promoting climate technology solutions for developed countries. This event was explored by us on 16th November 2022 when the number of relevant documents counted for COP27 while using the keyword "United Nations Framework Convention on Climate Change" in 2022 was 44 (currently it has 57).

Firstly and using the test split, we wanted to test this topic in a scenario where the user inputs the query "COP27" and uses the title-matching feature before the event occurs. To figure out the results of a search before the event happened we need to use a filter to remove the news after October 2022. The 18 documents presented were 4 pieces of news about the "COP26" event, followed by 7 news about "climate change" and 7 unrelated news which inform the user about the conference preceding the COP27 conference and about the topic prominent to the event which is climate change. This demonstrated that the model will still provide contextualized results to the topic even when dealing with unseen data. In Figure 5.2 we can see the interface shown to the user before and after the event happened without any manual interference in the system.



Query: COP27

We're Covering the Final Days of COP26 in Real Time

2021-11-10 Read more

There's a draft agreement that countries will use as a template for a final deal, but big hurdles remain at the Glasgow talks.

What Was Fashion Doing at COP26?

2021-11-17 Read more

Trading buzzwords, talking circularity and grappling with the role of clothes in the climate crisis.

It's Time for COP26. Here's Where We Stand.

2021-10-27 Read more

The leaders heading to the Glasgow climate talks have made progress in recent years. But Earth is still on track for dangerous warming unless those efforts accelerate

Event Occurs

Hello from COP27

2022-11-08 Read more

Frustration and distrust among leaders are high. But, for the first time, countries have agreed to discuss funding for climate reparations

Who's attending COP27, and who isn't.

2022-11-07 Read more

The leaders of some of history's largest polluters are slated to speak, as are those from countries that are bearing the brunt of the effects of climate change.

Western Europe, Climate Leader

2022-07-20 Read more

Yet it's suffering badly from warming.

Figure 5.2: List of news before and after "COP27" using our system

After the event happened and using the same evaluation settings as before, the 18 documents presented were 2 news about COP27, 13 about COP26 and other climate-related news and 3 unrelated news. The first 2 documents got a cosine-similarity score higher than 0.9 and the rest was below 0.9 which tells us that the embedding matching recognized and singled out the relevant news well but missed out on some relevant news since we still have 42 documents about COP27 that weren't suggested for our list. We also simulated the addition of a relevant image about COP27 by the user to use the multimodal filtering feature and got 6 more relevant news, making it a total of 8 news related to COP27 out of 18 in positions 1,2,11,13,14,16,17,18 of the list presented to the user.

Next, we wanted to see the experience of the user using the same settings as before but using the timeline sorting feature and the entire news collection to know more about how this event came to be and the overall sequence of events associated with this event. The sequence starts with news about climate change, followed by more news about climate change and summits over the years, 1 news article about COP26 and 1 news article about COP27 at the end of the list. The number of unrelated news was 5 but with the addition of an image by the user and with the multimodal filtering feature enabled, we got all the news relevant to the topic of climate change.

Following that experiment, we simulated a user that didn't know the name of the conference and instead tried to reach news related to COP27 through context. We used the same settings as the previous scenarios but used the query "Climate International Convention" and for the 18 documents presented, we only saw 1 about COP27 and the rest were all related to climate conferences and climate change. When using an additional image and the multimodal filtering feature we acquired 5 more relevant news, resulting in a total of 6 relevant news out of 18 in positions 3,5,6,7,8,10. Comparing the news related to COP27, the scenario where we specified the event identified 2 more news than when we didn't specify it.

In the final use scenario, we decided to reproduce a user that was attempting to use a query that explored a specific theme within the topic of COP27. Under the same settings but using the query "Richer countries pay for climate damage", the user will identify 8 related news to COP27 in the top of the list where 7 of those were also related to the specified theme and when an additional image is added for the multimodal filtering feature we get a total of 14 related news to COP27 in 18 with news about the specified theme remaining at 7. We were able to not only find news related to a theme within the topic but also, by searching for that theme, we got the best results when it comes to reaching relevant news for the topic of COP27 which shows how the added context can improve the results given to the user.

5.4.2 Suggestion Use Case Scenario

The second use scenario we created was to evaluate the user experience assuming the user neither had knowledge about the topic and wanted to learn more about it nor had knowledge about the system's features and just wanted to get the best possible experience using the text, an image or both.

Previously in Section 4.7, we introduced the suggestion feature as a solution to automate the selection of the best features from the matching features (Section 4.3), the sorting features (Section 4.4) and the filtering features (Section 4.6). Furthermore, the suggestion feature will be used to evaluate the user experience on the topics of "COVID" and "War in Ukraine" with the results extracted on 10th December 2022. We also saw in this use scenario an opportunity to evaluate the time interval sorting feature which is an essential part of the suggestion feature.

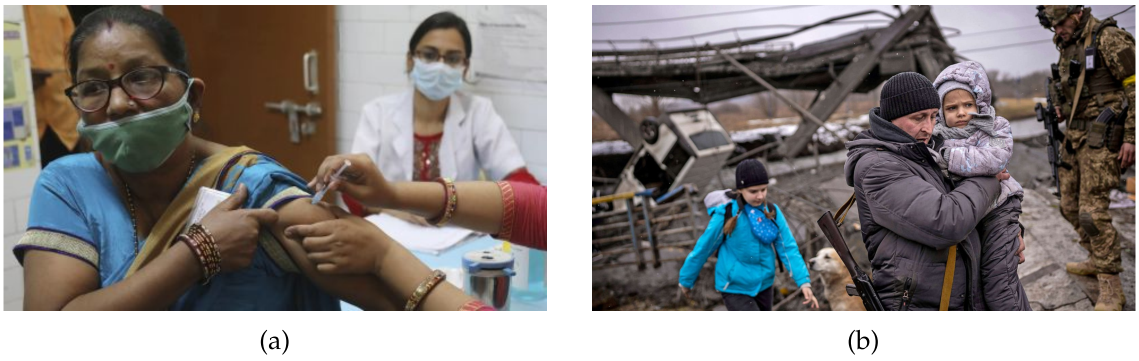
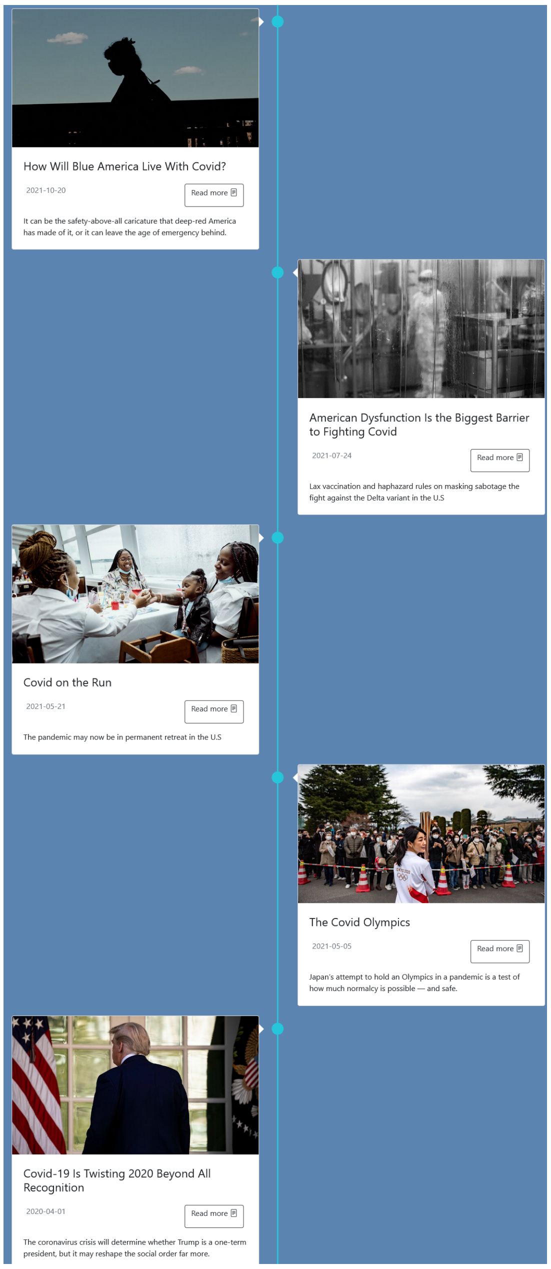


Figure 5.3: Selected images for the image query (a) COVID and (b) War Ukraine

In the use-case scenario where the user utilizes the text query "Covid", the system uses the title-matching feature, timeline sorting feature because there are enough relevant time intervals to make a timeline, and the time interval filtering feature because we didn't provide a text and an image to do the multimodal filtering feature. The user is presented with 18 news articles all related to the topic of COVID from 10 time intervals. Figure 5.4a shows an example of the results presented to the user in this use scenario. Compared to the results of the embedded title experiment in timeline format in Section 5.2.1, the user would be able to identify 3 more relevant news. When the user provides the image query "Covid" (Figure 5.3), the user will be presented with 18 news articles from 9 time intervals where 12 news are related to COVID and 6 to other diseases. Compared to the results of the embedded image experiment in timeline format in Section 5.2.1, the user would be able to identify 1 more relevant news. Finally, with the text and image query "Covid", the system will use the same features with the addition of the multimodal filtering feature resulting in 18 news articles related to Covid presented to the user from 10 time intervals, which compared to the results of the embedded title and image experiment in timeline format in Section 5.2.1, the user would be able to identify 1 more relevant news.

For the next topic, War Ukraine, the system uses the title-matching feature, timeline sorting feature and time interval filtering feature. The user is presented with 18 news articles all related to the topic of War Ukraine from 10 time intervals. Figure 5.4b shows an example of the results presented to the user in this use scenario. Compared to the results of the embedded title experiment in timeline format in Section 5.2.1, the user would be able to identify 3 more relevant news. When the user provides the image query "War Ukraine" (Figure 5.3), the user will be presented with 18 news articles all related to the topic of War Ukraine from 7 time intervals. Compared to the results of the embedded image experiment in timeline format in Section 5.2.1, the user would be able to identify 5 more relevant news. Finally, with the text and image query "War Ukraine", the system will use the same features with the addition of the multimodal filtering feature, resulting in 18 news articles related to "War Ukraine" presented to the user from 10 time intervals, which compared to the results of the embedded title and image experiment in timeline format in Section 5.2.1, the user would have the same amount of correct results shown.



How Will Blue America Live With Covid?

2021-10-20

Read more

It can be the safety-above-all caricature that deep-red America has made of it, or it can leave the age of emergency behind.

American Dysfunction Is the Biggest Barrier to Fighting Covid

2021-07-24

Read more

Lax vaccination and haphazard rules on masking sabotage the fight against the Delta variant in the U.S

Covid on the Run

2021-05-21

Read more

The pandemic may now be in permanent retreat in the U.S

The Covid Olympics

2021-05-05

Read more

Japan's attempt to hold an Olympics in a pandemic is a test of how much normalcy is possible — and safe.

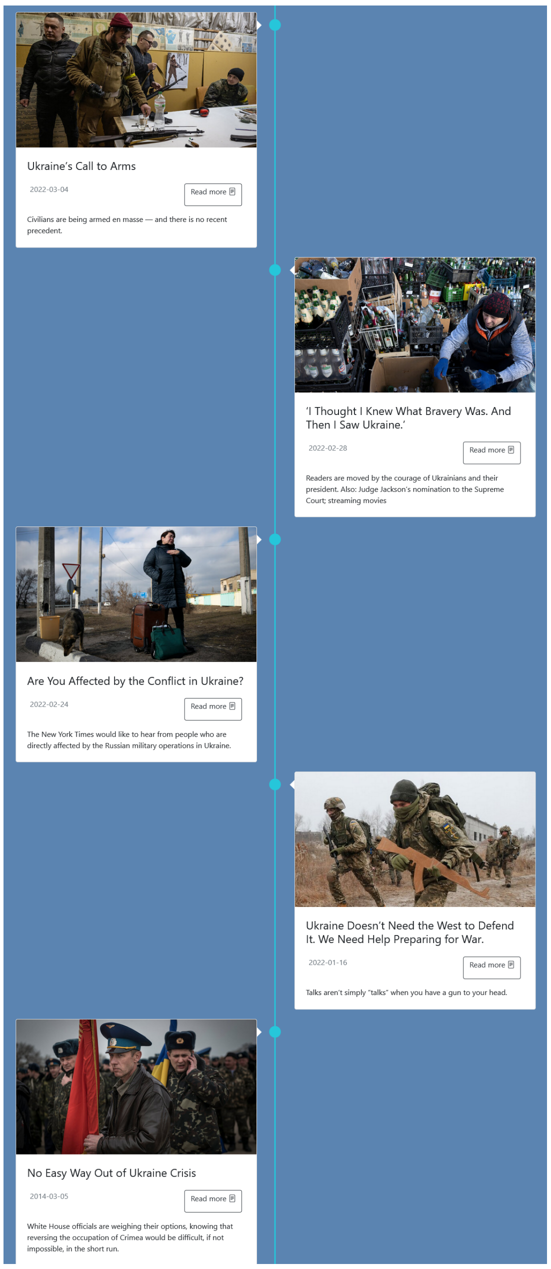
Covid-19 Is Twisting 2020 Beyond All Recognition

2020-04-01

Read more

The coronavirus crisis will determine whether Trump is a one-term president, but it may reshape the social order far more.

(a)



Ukraine's Call to Arms

2022-03-04

Read more

Civilians are being armed en masse — and there is no recent precedent.

'I Thought I Knew What Bravery Was. And Then I Saw Ukraine.'

2022-02-28

Read more

Readers are moved by the courage of Ukrainians and their president. Also: Judge Jackson's nomination to the Supreme Court; streaming movies

Are You Affected by the Conflict in Ukraine?

2022-02-24

Read more

The New York Times would like to hear from people who are directly affected by the Russian military operations in Ukraine.

Ukraine Doesn't Need the West to Defend It. We Need Help Preparing for War.

2022-01-16

Read more

Talks aren't simply "talks" when you have a gun to your head.

No Easy Way Out of Ukraine Crisis

2014-03-05

Read more

White House officials are weighing their options, knowing that reversing the occupation of Crimea would be difficult, if not impossible, in the short run.

(b)

Figure 5.4: User experience using recommendation feature on topics (a) Covid and (b) War Ukraine

5.5 Prompt Engineering

The suggestion use-case scenario indicated that the user experience was getting better resulting in better visualization of the results. However, we found some particular queries that brought irregular results to the visualization.

One of the queries we found that was causing irregular results even when using the suggestion feature was the query "earthquake". The results were mostly decontextualized with only a few depicting the topic at hand. When we added just a bit of context to the query such as "earthquake Syria" the results were much better achieving a wide range of correct results within the topic. These and the results we observed in Section 5.4.1, led us to the conclusion that the complexity of the query had a positive impact on the results visualized. However, we can't add unwanted context to a user-provided query. A solution we found to this problem was using prompt engineering.

In the CLIP paper [37] (model used for our embedded representations), the prompt engineering technique was mentioned as a technique that allowed them to better associate the images with the text during the training steps of the model by adding a prompt to the text such as "This is a picture of". This prompt was, rather than providing unwanted context, adding length to the text making it a more likely match for longer texts.

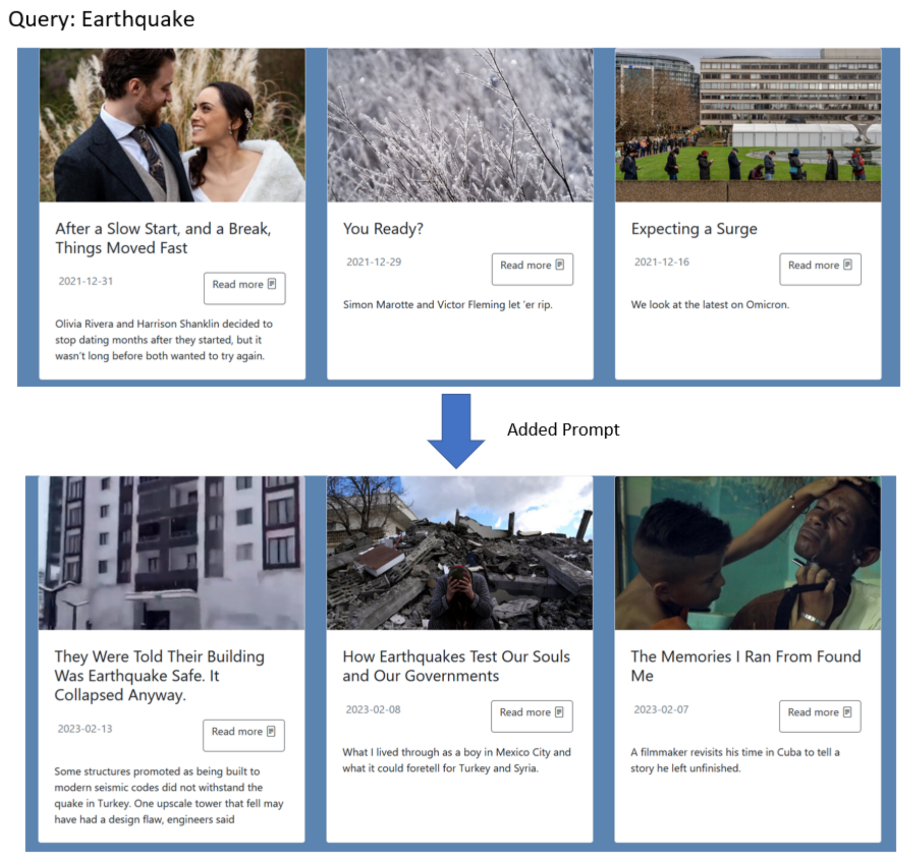
We decided to apply this solution for smaller and less specific queries that could benefit from a neutral context while keeping the bigger and more complex queries in their regular format because, from our experiments, these queries didn't benefit from the added prompt. Figure 5.5 shows the results before and after adding a prompt to the text query in our system.

5.6 Results Summary

Having evaluated all of our system's features in multiple scenarios, we will provide an overview of the key findings and outcomes of our analysis.

In general, the standard embedding-based search resulted in reasonable aggregations of relevant news even when compared to the results of the non-embedded search. We found out that some of the matching features resulted in better aggregations of relevant news for the topic. The title matching was overall the best feature in aggregating relevant news, followed by the paragraph matching feature and finally the image matching feature that showed how the context provided by encoded images was underperforming when trying to reach relevant results.

The multimodal feature, in which we used the title matching feature with the image to filter out the irrelevant results, showed promising results in the initial evaluations and was also able to slightly improve the results in the final comparison of the relevant features. This feature overall demonstrated that the image can be considered an important asset to achieve relevant aggregations of news when it provides additional context to our text query.



After a Slow Start, and a Break, Things Moved Fast
2021-12-31
Read more
Olivia Rivera and Harrison Shanklin decided to stop dating months after they started, but it wasn't long before both wanted to try again.

You Ready?
2021-12-29
Read more
Simon Marotte and Victor Fleming let 'er rip.

Expecting a Surge
2021-12-16
Read more
We look at the latest on Omicron.

They Were Told Their Building Was Earthquake Safe. It Collapsed Anyway.
2023-02-13
Read more
Some structures promoted as being built to modern seismic codes did not withstand the quake in Turkey. One upscale tower that fell may have had a design flaw, engineers said

How Earthquakes Test Our Souls and Our Governments
2023-02-08
Read more
What I lived through as a boy in Mexico City and what it could foretell for Turkey and Syria.

The Memories I Ran From Found Me
2023-02-07
Read more
A filmmaker revisits his time in Cuba to tell a story he left unfinished.

Figure 5.5: List of news before and after using a prompt on "earthquake" using our system

The suggestion feature used multiple relevant time intervals as a filter that were acquired through an initial query using the title matching feature. We could see from the results in the final comparison of the relevant features that there was an improvement in the ranking of relevant news compared to the multimodal feature. This feature overall demonstrated that recognizing relevant time intervals and restricting the results to those intervals leads to more relevant aggregations of news.

The use-case scenarios helped us understand how accurate the results were for users with less or more understanding of our system. For the recent topics chosen for these scenarios, we continued to observe relevant aggregations of news using the multimodal and the suggestion feature even though the model wasn't trained with information about those topics. We also noticed how more complex queries would positively impact the user experience since they resulted in better rankings of news.

Finally, we saw that prompt engineering was able to mitigate the poor performance of results observed in simpler and smaller queries without affecting the context of the query.

6 Conclusions and Future Work

This chapter will focus on making conclusions about what we were able to culminate throughout the different stages of this project and how those conclusions will affect the study of similarity-based search using embeddings in the context of gathering information from events through news going forward. We will be looking at all the evidence and results we've acquired and questioning whether we were able to accomplish the goals we proposed in the introduction of this thesis. We will start by revisiting the challenges we assumed in the introduction of the thesis; reviewing the objectives proposed; finalizing thoughts on the overall project and future work.

6.1 Revisiting Intentions and Challenges

In this section, we will be revisiting the intentions and challenges identified at the beginning of this thesis. For the challenges, how they actually imposed during the creation of our dataset, the development of the system and the evaluation of the system and how we solved or mitigated their impact on our system. The ideas we followed through were the following:

- Deliver up-to-date and rich multimodal news browsing.
- Contextualize and let readers "witness" the event with their own eyes, by providing visual cues.
- Zero-shot deep multimodal (vision and language) approaches for the news domain, to analyze and aggregate news pieces on-the-fly.
- Collect and then semantically organize news streams' content in a way that events become structured according to topics unfolding.

Here are the challenges we overcame:

- **What information is worth extracting from the news articles? -** For this challenge, we ended up extracting most of the news articles' information and made different evaluations to compare the impact of the different news informative fields. This allowed us to find that the title field is the most relevant information field for similarity-based search using embeddings. The challenge we found in extracting news was rather the fact that not all news had all the information we needed and we ended up having to discard some news for their lack of information, or using a web crawler to extract additional relevant information.

- **What model would be used to achieve rich representations of data but that would also enable the zero-shot classification? -** We ended up discussing a variety of models in Section 2.6.3, models that either were able to achieve great representations but lacked in the zero-shot scenario, or models trained to perform in zero-shot scenarios but the representations achieved were inadequate. Based on the results achieved, we observed that CLIP is capable of attaining the best representations in the zero-shot scenario, and that made it more apparent that CLIP was the ideal model to obtain the embeddings for our embedding similarity search.

- **How can we store embeddings that can be accessed through other embeddings? -** We found OpenSearch to work well in this scenario because not only were we able to store the embeddings in the form of vectors and other relevant information from news, but we could also use those vectors to respond directly to queries done using embeddings by using the approximate k-NN search plugin and the cosine-similarity for embedding similarity searches.

- **What should be done to automate the process of extraction, processing and indexing of the news articles? -** The solution found was to run a script every day at a specific time that would execute the news extraction pipeline making the news readily available to be used from the day before instead of requesting news as soon as a new article was published.

- **How are we presenting the different features of our system using a simple interface? -** We ended up designing a simple interface with an area specified for data forms to be used in our queries and an area for visualizing the results of the query but to distinguish between the different features we identified different endpoints that were associated with each feature and that way we could perform queries on the different features and visualize the results all in the same space.

- **How are we going to check if a news article is relevant or not? -** We manually observed some of the news information to check if it was relevant to the topic and when it came to the automatic evaluation, we were fortunate to have keywords for each news article associated to a specific topic which allowed us to assess the relevance more objectively.

- **How should we account for mistakes made on the rankings and how should we organize the resulting news? -** This challenge is also partially correlated to our objective of finding the best possible results for news related to a topic and structuring the news in relevant formats and will be further seen in Section 6.2. The solution we found to account for mistakes made on the rankings was by using filtering features that would filter out the non-relevant news articles from the results while adding relevant ones to the candidate documents. To organize the results we used sorting features that helped to visualize the news in significant formats.

6.2 Objective Review and Final Thoughts

Finally, we reach the final stage where we assess whether we achieved the goal or objective proposed. This objective can be separated into different smaller objectives that were a constant point of focus during the elaboration of the thesis.

1) **The creation of an ever-evolving dataset updated with the latest events published by the news media and their relevant information processed into embeddings** was an essential means to reach our objective and was accomplished in this work. The use of a news API in the NewYorkTimesAPI with all the relevant information from news including the images readily available when they are published on the website. We leveraged a pre-trained model capable of processing text and images into rich visual representations and optimized for zero-shot text and image embedding extraction in CLIP, and a script that makes requests to the API daily to extract and process the news in an automatic fashion. This allowed us to achieve the step of providing users an article collection updated with the latest news.

2) **The similarity-based search using embeddings** was an essential feature where we needed to perform searches directly from the encoded query provided by the user on the embeddings processed during the creation of the dataset which we were able to accomplish in this work. The creation of special indexes that allowed the search for vectors was a feature available in OpenSearch which allowed us to achieve another step toward our goal. The similarity-based search using embeddings or matching features as we've mentioned in this document was evaluated in different scenarios. We concluded that the use of images as a query or as matching for the query resulted in less than optimal results, and the use of text as a query or as matching for the query was performing much better given the number of relevant news identified in our candidate documents. However, there

was room to improve when certain irrelevant news articles were outranking relevant news articles in our candidate documents given the topic.

3) **The aggregation of news to contextualize the user on the sequence of events** was an important step in order to accomplish our goal. We designed our own sorting features such as the timeline feature that was able to inform the user of an event from the earliest published news article to the latest while keeping the sequence of news relevant to the given the topic. The evaluations done using the timeline feature allowed us to conclude that despite showing less relevant news than the relevance sorting feature which is ordered based on similarity score, it was able to contextualize the user on the different stages of the event while showing relevant news for the chosen event, which we consider to be another successful step to accomplish our goal.

4) **The development of features to improve the number of relevant news in the candidate documents to prevent misinformation** was a key stride concerning our goal. We designed our filtering features which apply filters to the results attained from the similarity-based search using embeddings that remove irrelevant news articles from the candidate documents while inserting relevant ones. The multimodal filter, based on our evaluations, was able to use the context of the image to identify more relevant news in specific scenarios and present them to the user. While the feature showed improvements in specific scenarios, when we tested it in a more diverse set of topics, the results were not as significant. Therefore, we concluded that the feature generally improved results, but its impact varied depending on the specific scenario, and could potentially lead to a significant enhancement in the user experience. The time interval filter, based on our evaluations, was able to obtain more relevant news in the candidate documents by restraining the possible candidate documents to a given time interval. However, this would also mean that the diversity of news would be compromised for the improvement in accuracy which also meant less contextualization for the user. We believe that with the suggestion feature, we were able to achieve a good balance of diversity and accuracy and this theory was supported by the results. It was able to achieve good sequences of events to contextualize the user and great improvements in the number of relevant news articles to prevent a restricted view of the events and that contributed to reaching our goal.

It was apparent to us that the suggestion feature resulted in the best searching feature because it not only provided the best overall experience for the user, but also reached the best scores for the different metrics in the final evaluation with the other features proving to be possible alternatives in specific scenarios. In the end with all the steps towards our goal mentioned previously resulting in successful developments, we can say more evidently that our objective of **providing a system capable of browsing the latest events published on the news with embedding similarity search using both textual and visual modalities and aggregate relevant news in appropriate structures to prevent restricted views of events and contextualize the user on the sequence of events given a topic** was accomplished.

6.3 Future Work

This last section will mention different ideas that weren't explored in this work but could lead to new discoveries in the area of browsing on-the-fly multimodal news under a zero-shot setting.

Long Text Tokens - The entire body of the news for each news article was extracted to the news collection but we couldn't encode information longer than 200 characters (limit defined by the CLIP model) which limited our options of matching fields. The context provided by the full body could've led to better results in the embedding-based search and is something worth exploring in the future.

News Sources - We centered our study around the news articles published by the New York Times but more sources of news could be worth exploring to see how much influence different types of news have on the embedding-based search or if the embedding-based search is more biased towards certain formats of news.

Model Comparison - We used the model CLIP because it showed the most promise in the zero-shot setting but it's impossible to say confidently that CLIP is the best model for this setting without studying the behavior of other models in the same circumstances. Thus, studying the behavior of other models that can perform in the zero-shot setting can be an opportunity for further investigation.

Multimodal Embeddings - The use of embeddings that combined both textual and visual modalities into one representation is a possible idea to add to this work because it could lead to interesting results when it comes to matching which in our case was made by either using an image or a text and the only time we combined both modalities was when we used text matching and an image as a filter.

Domain Adaptation - There is also the possibility of feeding pairs of text and images from news articles to the CLIP model so the model can train and adapt more to the domain of the news rather than the generic domain approach done by the creators of the CLIP model. This requires not only retraining the model but also processing and storing the new embeddings.

Interface Complexity - Finally, the interface shown in our work allowed us to visualize news using different structures but there's room to complexify our simple interface with more visualizations for the results, better visual indicators and more interactability with the results. These can be designed with the objective of guiding the user in more complicated browsing scenarios.

Bibliography

[1] URL: https://www.nytimes.com/ (cit. on pp. 9, 10, 12).

[2] URL: https://sh-tsang.medium.com/review-cmc-contrastive-multiview-coding-68e1563bc3d8 (cit. on p. 21).

[3] URL: https://opensearch.org/ (cit. on pp. 27, 36).

[4] URL: https://ewn.co.za/2020/11/08/trump-joins-two-other-former-us-leaders-as-one-term-president (cit. on p. 29).

[5] URL: https://newsapi.org/s/google-news-api (cit. on p. 29).

[6] URL: https://developer.nytimes.com/apis (cit. on p. 32).

[7] URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (cit. on p. 34).

[8] URL: https://opensearch.org/docs/latest/search-plugins/knn/approximate-knn/ (cit. on pp. 43, 46).

[9] URL: https://flask.palletsprojects.com/en/2.2.x/ (cit. on p. 44).

[10] S. Atito. "SiT: Self-supervised vision Transformer". In: 2020 (cit. on p. 18).

[11] C. Bartolomeu, R. Nóbrega, and D. Semedo. "Understanding News Text and Images Connection with Context-enriched Multimodal Transformers". In: *Proceedings of the 30th ACM International Conference on Multimedia*. Lisbon, Portugal: Association for Computing Machinery, 2022 (cit. on pp. 2, 3, 8, 10, 23, 30, 31).

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. "Language models are few-shot learners". In: vol. 2020-December. 2020 (cit. on p. 12).

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: vol. PartF168147-3. 2020 (cit. on p. 20).

[14] S. Chowdhury, X. Dong, and X. Li. "Recurrent Neural Network Based Feature Selection for High Dimensional and Low Sample Size Micro-array Data". In: 2019. DOI: 10.1109/BigData47090.2019.9006432 (cit. on p. 13).

[15] A. M. Dai and Q. V. Le. "Semi-supervised sequence learning". In: vol. 2015-January. 2015 (cit. on p. 11).

[16] Z. Dai and R. Huang. "A Joint Model for Structure-based News Genre Classification with Application to Text Summarization". In: 2021. DOI: 10.18653/v1/2021.findings-acl.295 (cit. on p. 8).

[17] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: vol. 1. 2019 (cit. on pp. 11, 14, 15).

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: 2020 (cit. on p. 17).

[19] J. Ferreira, D. Lavado, R. Gonçalves, M. Knorr, L. Krippahl, and J. Leite. "Faster Than LASER - Towards Stream Reasoning with Deep Neural Networks". In: vol. 12981 LNAI. 2021. DOI: 10.1007/978-3-030-86230-5_29 (cit. on p. 13).

[20] R. Ferreira, C. Lopes, R. Gonçalves, M. Knorr, L. Krippahl, and J. Leite. "Deep Neural Networks for Approximating Stream Reasoning with C-SPARQL". In: vol. 12981 LNAI. 2021. DOI: 10.1007/978-3-030-86230-5_27 (cit. on p. 13).

[21] F. Hamborg, C. Breitinger, and B. Gipp. "GiveMe5W1H: A universal system for extracting main events from news articles". In: vol. 2554. 2020 (cit. on p. 25).

[22] K. He, X. Zhang, S. Ren, and J. Sun. "ResNet V1". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem (2015). ISSN: 10636919 (cit. on p. 17).

[23] J. Howard and S. Ruder. "Universal language model fine-tuning for text classification". In: vol. 1. 2018. DOI: 10.18653/v1/p18-1031 (cit. on p. 11).

[24] C. H. Lampert, H. Nickisch, and S. Harmeling. "Learning to detect unseen object classes by between-class attribute transfer". In: 2009. DOI: 10.1109/CVPRW.2009.5206594 (cit. on p. 22).

[25] A. Li, A. Jabri, A. Joulin, and L. V. D. Maaten. "Learning Visual N-Grams from Web Data". In: vol. 2017-October. 2017. DOI: 10.1109/ICCV.2017.449 (cit. on pp. 22, 23).

[26] F. Liu, Y. Wang, T. Wang, and V. Ordonez. "Visual News: Benchmark and Challenges in News Image Captioning". In: 2021. DOI: 10.18653/v1/2021.emnlp-main.542 (cit. on pp. 19, 24).

[27] J. Liu, T. Liu, and C. Yu. "NewsEmbed: Modeling News through Pre-trained Document Representations". In: 2021. DOI: `10.1145/3447548.3467392` (cit. on pp. 21, 27).

[28] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: `https://github.com/joaomlourenco/novathesis/raw/master/template.pdf` (cit. on p. ii).

[29] J. Lu, D. Batra, D. Parikh, and S. Lee. "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks". In: 32 (2019). Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. A. Buc, E. Fox, and R. Garnett. URL: `https://proceedings.neurips.cc/paper/2019/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf` (cit. on pp. 18, 31).

[30] G. Marcelino, R. Pinto, and J. Magalhães. "Ranking news-quality multimedia". In: 2018. DOI: `10.1145/3206025.3206053` (cit. on p. 9).

[31] G. Marcelino, D. Semedo, A. Mourão, S. Blasi, J. Magalhães, and M. Mrak. "Assisting News Media Editors with Cohesive Visual Storylines". In: 2021. DOI: `10.1145/3474085.3475476` (cit. on p. 9).

[32] V. V. Nakkirtha, L. Jiang, C. Swierczewski, and J. Mazanec. "Build K-Nearest Neighbor (k-NN) Similarity Search Engine with Elasticsearch". In: 2020 (cit. on p. 27).

[33] N. Oostdijk, H. van Halteren, E. Baar, and M. Larson. "The connection between the text and images of news articles: New insights for multimedia analysis". In: 2020 (cit. on p. 9).

[34] A. Patel, L. Cheung, N. Khatod, I. Matijosaitiene, A. Arteaga, and J. W. Gilkey. "Revealing the unknown: Real-time recognition of galápagos snake species using deep learning". In: *Animals* 10 (5 2020). ISSN: 20762615. DOI: `10.3390/ani10050806` (cit. on p. 16).

[35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. "Deep contextualized word representations". In: vol. 1. 2018. DOI: `10.18653/v1/n18-1202` (cit. on p. 12).

[36] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. "Improving Language Understanding by Generative Pre-Training". In: *Homology, Homotopy and Applications* 9 (1 2007). ISSN: 15320081 (cit. on p. 11).

[37] A. Radford, J. Wook, K. Chris, H. Aditya, R. Gabriel, G. Sandhini, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. *CLIP: Learning Transferable Visual Models From Natural Language Supervision*. 2019 (cit. on pp. 3, 18–22, 24, 25, 34, 74).

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of Machine Learning Research* 21 (2020). ISSN: 15337928 (cit. on p. 11).

[39] H. Tan and M. Bansal. "LXMert: Learning cross-modality encoder representations from transformers". In: 2019. DOI: 10.18653/v1/d19-1514 (cit. on pp. 3, 18, 23, 31).

[40] Y. Tian, D. Krishnan, and P. Isola. "Contrastive Multiview Coding". In: vol. 12356 LNCS. 2020. DOI: 10.1007/978-3-030-58621-8_45 (cit. on p. 20).

[41] M. Tong, S. Wang, Y. Cao, B. Xu, J. Li, L. Hou, and T. S. Chua. "Image enhanced event detection in news articles". In: 2020. DOI: 10.1609/aaai.v34i05.6437 (cit. on p. 26).

[42] A. Tran, A. Mathews, and L. Xie. "Transform and tell: Entity-aware news image captioning". In: 2020. DOI: 10.1109/CVPR42600.2020.01305 (cit. on pp. 15, 18, 24, 30).

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is all you need". In: vol. 2017-December. 2017 (cit. on pp. 14, 15).

[44] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, B. van Merrienboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-farley, J. Chorowski, Y. Bengio, C. Szegedy, S. Reed, P. Sermanet, V. Vanhoucke, A. Rabinovich, J. Pan, C. J. Ng, A. J. Teoh, Z. C. Lipton, K. Konda, X. Bouthillier, P. Vincent, R. Memisevic, J. Schmidhuber, G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, H. C. H. Su, Y. Gong, Q. Ke, M. Isard, S. Lazebnik, E. Denton, S. Chintala, A. Szlam, R. Fergus, Y. N. Dauphin, Y. Bengio, and X. Z.
bibinitperiod S. K. D. Povey. "A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv : 1506 . 00019v2 [cs . LG] 29 Jun 2015". In: *International Journal of Computer Vision* 106 (March 2013 2015). ISSN: 18792782 (cit. on p. 13).

[45] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao. "VinVL: Revisiting Visual Representations in Vision-Language Models". In: 2021. DOI: 10.1109/CVPR46437.2021.00553 (cit. on p. 18).

2023 Multimodal On-the-fly News Media Exploration Ricardo Barqueira

NOVA SCHOOL OF SCIENCE & TECHNOLOGY