Meme Search: Design and implementation of a Search Engine for internet memes

Héctor Martel
Department of Computer Science and Technology
Tsinghua University
`hkt20@mails.tsinghua.edu.cn`

Abstract

This paper presents a Search Engine for internet memes called Meme Search. The task of searching in a collection of memes presents some challenges regarding the data modalities, the diversity of messages, and its true intention. The problem is approached using various pre-trained machine learning models that have been used successfully in text representation learning and image classification tasks. A unified framework is used to perform searches in multiple data modalities. An efficient method to perform nearest neighbors search in a high-dimensional vector space is implemented using KD-trees to retrieve the results. In addition, the system design and deployment are discussed attending to the use of this Search Engine in a real-world scenario. The full code is available on GitHub[1].

1 Motivation and background

Internet memes are a form of user-generated content that is becoming increasingly popular, often used in social media sites and messaging apps. The main purpose of a meme is to make fun of something using the combination of an image and a text caption.

The true meaning of a meme is relatively is easy to understand for humans, while it presents many challenges for computer systems. Despite being just images, memes are multi-modal in nature, containing 2 modalities of data: visual and text. It is common for a meme to present visual content and a text caption that do not match in a logical sense for the purpose of the joke. Therefore, users may want to search using text or images as the query.

There are many potential use-case scenarios in which a Search Engine for memes can be useful. First, it allows users to access a big unstructured collection of memes in a friendly manner. Second, it can provide a global perspective of how the most popular memes evolve over time. Finally, it can be used to categorize memes by extracting relations of similarity between them and lets users discover new ones.

2 Related work

The purpose of this section is to present existing solutions that are related to Meme Search and analyze their core functionalities. 4 mainstream commercial meme hosting websites are considered.

Reddit.com is a discussion website containing many topics, including memes, that are organized by forum-like threads called subreddits. The content of a subreddit is a infinite scroll feed with posts in reverse chronological order (newest frist), usually containing 1 meme per post. Contents inside a subreddit can be searched using text and various filters such as publication date or number of likes.

[1]Meme Search, on GitHub: `https://github.com/hmartelb/meme-search`

Imgflip.com is a website dedicated specifically to memes that lets users create their own from template images. There is a search functionality to find memes and templates by text or category tags.

Me.me is another website dedicated to memes, with search by text input and category tags. Additionally, the page of a particular meme displays recommendations of other similar memes below.

9gag.com offers memes and other fun content in an infinite scroll feed, and also in "trending" or "new" categories. The search functionality covers text queries and category tags.

The above pages have input suggestions as well as other types of recommendation and advanced filters based on text. However, to the best of the author's knowledge, there is no reverse image search or "similar images" function available in any of the consulted websites. Only the text information is considered, which is available beforehand and manually annotated by the users in most cases.

3 System design overview

The design choices were made to facilitate the ease of integration of machine learning into the technology stack. All the system is written in Python, using the Flask framework[2] for the frontend web server and FastAPI[3] for the backend Search API. These 2 components run live on the server and have differentiated tasks: the frontend server communicates with the user, providing it with the HTML and other assets to visualize the website. The backend API handles the connection with the search index and solves the incoming queries, using JSON as input and output data format. The crawling and feature extraction procedures are performed off-line on a GPU-enabled machine, since the latter is very computationally intensive. The live search index in the server can be updated without causing any downtime when the offline computation finishes. The overall system design diagram is shown in Figure 1.

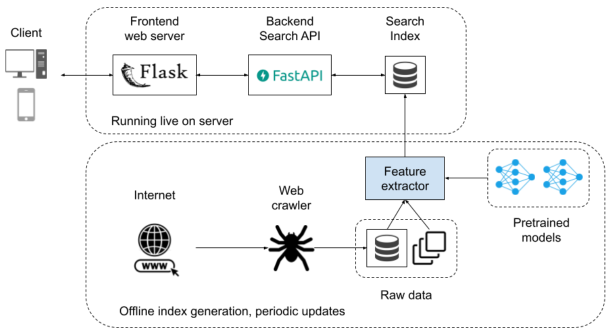


Figure 1: Application architecture diagram. The top block contains the Frontend and API compoents that are run live on the server, while the bottom block contains the more resource-intensive data collection and feature extraction processs.

Some features from Figure 1 have been designed with a real production environment in mind. To save compute resources in the Search API, a cache system is implemented between the frontend and the backend servers. The motivation is to avoid sending repeated requests to the API if the results for the same query have been returned previously within a certain time-frame. By this cache mechanism, users can get results for frequent or trending queries faster while reducing the load of the backend. Moreover, the separation of the modules allows for horizontal scaling of frontend and backend independently according to the application demands.

[2]Flask framework: `https://flask.palletsprojects.com/en/2.0.x/`
[3]FastAPI framework: `https://fastapi.tiangolo.com/`

4 Technical details

The queries can be divided into the following 2 types: query by text input and query by image. In the second case, a user can query similar memes using an existing one in the search index or via an external image URL. The image is downloaded to compute its individual features at query time.

Although the first intuition is to resolve each query type with different methods, a unified framework is employed by observing that both modalities can be stored using a dense vector representation. The approach relies on high-level features that are extracted using Deep Learning techniques, in which the vector space captures semantically relevant relationships. Under this assumption, another problem becomes to efficiently handle the high-dimensional data and retrieve the results without significant delays. The feature extraction methods are presented in subsection 4.1 and the results retrieval methods are presented in subsection 4.2.

4.1 Feature extraction

As mentioned in section 1, the data is composed by visual and text modalities, which requires the combination of Natural Language Processing and Computer Vision.

The text is written in the image, so the first step is to use an Optical Character Recognition (OCR) model for text extraction. This step can be skipped if the original post contains the metadata of the text caption, but this is not always guaranteed, hence the need for OCR to complete the search index. Image-to-text conversion is a Computer Vision task that can be performed by CNN image classifiers (for single symbols) or more sophisticated Neural Network systems (for long and complex text). Meme Search integrates the EasyOCR framework[4] because of its flexibility and the SOTA pre-trained models that it provides. The code is based in Pytorch[5] and builds on top of CRAFT [7] and CRNN [4] (which integrates a ResNet [3] feature extractor) for the detection and recognition parts respectively.

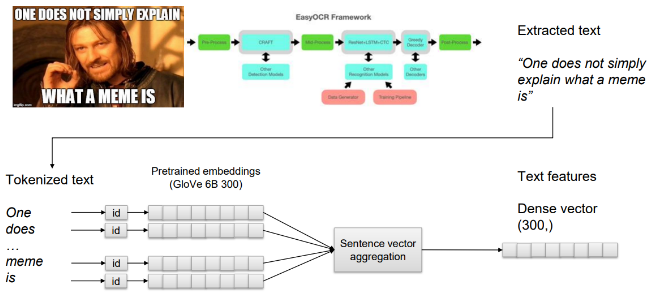


Figure 2: Overview of the text extraction pipeline. The EasyOCR framework is followed by GloVe word embeddings for dense feature extraction.

The ResNet achitecture [3] has proven successful in the ImageNet classification challenge [2]. In fact, it is a popular pre-trained model for image recognition or classification. The predictions of class probabilities are not directly used for high-level similarity. It has been found experimentally that taking an intermediate representation of the last fully connected layers is more effective. In Meme Search, the selected representation is a dense vector of 512 dimensions.

A Python implementation of Locality Sensitive Hashing (LSH)[6] was considered at first for image similarity, but discarded in favor of ResNet features. LSH is very sensitive to pixel value changes and works with low-level information. In contrast, ResNet features are more robust to changes in spatial locations, subject pose, etc. of images that are still similar on a higher-level structure.

The text is converted to a dense vector of 300 dimensions by tokenizing each word, obtaining their GloVe embeddings[7], and aggregating all the word vectors. Although many aggregation methods can be used, such as LSTM encoder[5] or a Sentence Transformer[8], the arithmetic mean across all the

[4]EasyOCR by JaidedAI, on GitHub: `https://github.com/JaidedAI/EasyOCR`
[5]Pytorch Deep Learning framework: `https://pytorch.org/`
[6]LSH for near-duplicate image detection, on GitHub: `https://github.com/mendesk/image-ndd-lsh`
[7]GloVe, Stanford NLP: `https://nlp.stanford.edu/projects/glove/`

word vectors was found to work well in practice while being much faster to compute. The reason is that typically the text captions are short and may not contain grammatically correct sentences, thus, decreasing the effectiveness of sentence embedding methods. The complete text processing pipeline is shown in Figure 2.

Each meme is analyzed using the text and vision models to obtain dense feature vectors. In the case of text, there are 3 feature vectors corresponding to "title", "inner text" and "all text", each having 300 dimensions. The image features are a single dense vector of 512 dimensions.

The feature extraction procedure is repeated for a total of 3200 images that conform the search index. This number has been obtained by crawling the website `imgflip.com` using the topics "computer science", "maths" and "finance", and performing a duplicate removal operation. Two memes are considered duplicates if their Euclidean distance in image feature space is smaller than a predefined threshold, or if the combined text and title are an exact match.

4.2 Results retrieval

When a query is sent Meme Search, it is processed according to its data modality to obtain a dense feature vector, hereafter called query vector. The query vector is then compared against all the feature vectors that are already present in the search index to find the best candidate results. In feature space, the distance is correlated with the high-level similarity of the image contents or the sentence meanings, implying that the closest instances to the query vector are the most relevant results. Ranking the results is trivial, since sorting them by distance in ascending order provides an appropriate relevance ranking.

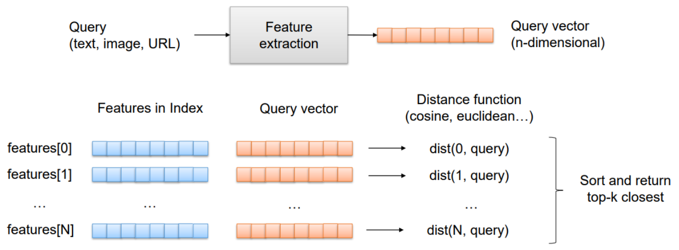


Figure 3: Illustration of the linear search over the index. The query vector (orange) is compared with each feature vector (blue) in the index, resulting in N comparisons. The distance metric is Euclidean for non-normalized features and Cosine for normalized features.

First, a naive solution to obtain the results of a query is to compute the distance between every instance in the search index and the query vector. This results in a linear runtime $O(n)$ since the index needs to be scanned from the beginning to the end. After that, the distances need to be sorted to obtain the ranking to finally select the top-k elements. The sorting operation takes $O(nlog(n))$ and imposes a performance bound in the search. However, this strategy performs too many non-relevant comparisons because the only important results are the top-k. The remaining elements do not need to be sorted. An optimization is to use a min-heap with capacity k, to store at most k elements in sorted order, keeping the ones with minimum distance while traversing the search index. Again, this results in linear runtime complexity $O(n)$. An illustration of this linear search can be found in Figure 3.

A linear runtime is generally a bad choice for search. The performance impact may not be critical in this case with thousands of elements, but it is definitely problematic when the search index contains of millions of elements [6]. Large technology companies such as Facebook and Spotify have faced this problem and open-sourced FAISS[8] and Annoy[9] respectively to tackle dense vector similarity search. Both solutions are based on k-dimensional trees combined with other strategies to obtain *approximate* nearest neighbors on very large-scale datasets. These algorithms are too advanced for the purpose of Meme Search, which implements the vanilla version of the k-dimensional tree from the Scipy spatial library[10].

[8]Faiss, on GitHub: `https://github.com/facebookresearch/faiss`
[9]Annoy, on GitHub: `https://github.com/spotify/annoy`
[10]Scipy spatial: `https://docs.scipy.org/doc/scipy/reference/spatial.html`

The k-dimensonal tree (KD-tree [1]) is a special type of search tree, similar to a Binary Search Tree (BST), that employs a clever approach to turn the high dimensionality of the data into an advantage. The tree is constructed by alternating axes to obtain a binary decision boundary, one dimension at a time, using the median value of the data along that dimension as threshold. The axis is changed to the another dimension in the next level of depth of the tree. The region of interest is reduced significantly, delimited by multiple high-dimensional hyper planes. When the region of interest is small enough, the leaf level of the tree is reached and the distances between the remaining elements and the query vector are computed using a brute force linear scan. For ease of visualization, a 2D example is shown in Figure 4. The post-processing steps are detailed in Figure 5.

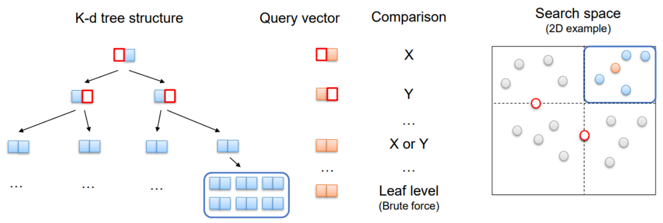


Figure 4: Illustration in 2D of the search over the index using a KD-tree. The query vector (orange) is compared with each feature vector (blue) in the index, resulting in $log_d(N) + C$ comparisons, where d is the number of dimensions and C a constant at the leaf level. The distance metric is Euclidean for non-normalized features and Cosine similarity for normalized features.

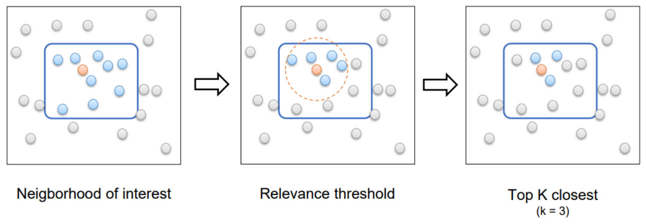


Figure 5: Filtering steps after the search space reduction in the KD-tree. The points in the neighborhood of interest need to be post-processed to eliminate non-relevant results. For that, a relevance threshold found empirically is applied to the data. In case of more matches than the top-k results, the k-closest are selected.

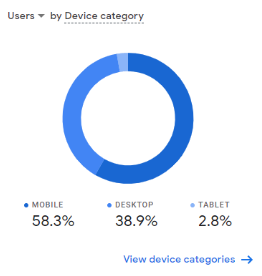
We can further speed-up the computation by applying dimensionality reduction, such as PCA or t-SNE, before constructing the KD-tree. The reduced feature vectors are stored in the search index and the same transformation is applied to the query vector before the comparisons. Since dimensionality reduction is a way of lossy compression, it provides speed improvements while degrading the accuracy of the retrieved results. Instead of exact distances, the results become an approximation of the true values. For Meme Search, reducing the dimensions to less than 100 showed significant accuracy degradation, with many results being irrelevant to the query. From 250 dimensions and higher, results were found to be acceptable. However, the current size of the search index allowed to resolve queries in reasonable time without dimensionality reduction.

5 Experiments and results

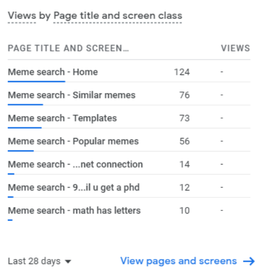
5.1 Usage statistics

The tools provided by Google Analytics[11] were used to study the user behavior on the website during 2 weeks of early testing. The analysis revealed some valuable insights about what devices technologies they use, from which country they access, and an the navigation flow between the different screens. This information is shown in Figure 6. The observations are presented together with the necessary changes (already implemented or future) to improve the user experience.

[11]Google Analytics: `https://analytics.google.com/`

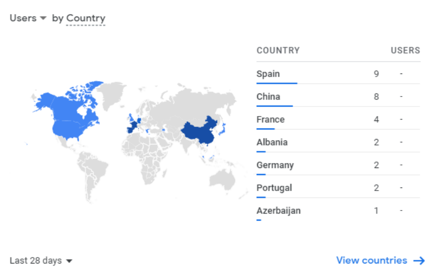


(a)


Views by Page title and screen class

PAGE TITLE AND SCREEN…	VIEWS	
Meme search - Home	124	-
Meme search - Similar memes	76	-
Meme search - Templates	73	-
Meme search - Popular memes	56	-
Meme search - ...net connection	14	-
Meme search - 9...il u get a phd	12	-
Meme search - math has letters	10	-

(b)



COUNTRY	USERS	
Spain	9	-
China	8	-
France	4	-
Albania	2	-
Germany	2	-
Portugal	2	-
Azerbaijan	1	-

(c)

Figure 6: Usage statistics from the launch of Meme Search up to the present day. (a) Relative use of devices used to access the website. (b) Number of views per page. (c) Number of users by country.

Figure 6 (a), shows that the majority of users prefer to access the website using a mobile phone, with a share of 58.3%, followed by 38.9% of desktop users. Therefore, it is necessary to optimize the User Interface for mobile devices. Screen sizes and interactions differ from a desktop computer, and also, mobile devices may have slower network connections. To alleviate this last issue, the website assets are compressed using Flask-Assets[12] and all the memes are hot-linked to their original source, which is likely to serve them using a Content Delivery Network (CDN).

Figure 6 (b) reveals that the home page is the most viewed, which is expected. However, the second page in number of views is "Similar memes". This page is triggered when a user performs a similarity search with an existing instance in the search index. Interestingly, it has been used more times than all the text input queries combined. The popularity of this feature, even surpassing fixed sections like "Templates" or "Popular memes" indicates that a promising direction is to build more recommendation functionalities on top of the Search Engine, to let users discover new content.

Figure 6 (c) shows that users from different places in the world have accessed the website in its current English version. The number of active users can be increased by adding support (translations and contents) in Spanish, Chinese and other European languages such as French, German or Portuguese. Word vectors and OCR pre-trained models are available in many of these languages, and translations can be incorporated using Flask-Babel[13]. Therefore, this enhancement can be easily accomplished in the future.

5.2 User feedback

The early testing phase of the Search Engine provided some opportunities for the users to express their direct feedback and suggest improvements. Detailed discussions with 5 users over a video call have originated some of the current features, such as the "Popular memes" screen (see Figure 7 (d)) and "Input suggestions" (see Figure 9).

6 Conclusions and future work

A functional Search Engine for internet memes, called Meme Search, has been presented. The main differentiation factor with respect to the existing solutions presented in section 2 is the image similarity search (see Figure 8), currently lacking in all 4 websites under study. The choices regarding the Search Engine design and implementation, discussed in section 3, have been made considering the current state of the application prototype and, more importantly, its scalability into a real-world web application in the near future. The usage data from section 5 pointed out some areas to be improved regarding usability. Further directions to turn Meme Search into a viable web service are discussed below.

[12]Flask-Assets: `https://flask-assets.readthedocs.io/`
[13]Flask-Babel: `https://flask-babel.tkte.ch/`

Model fine tuning. It must be acknowledged that there is still room for improvement in the Machine Learning part of this search engine. The pre-trained models used for feature extraction have not been fine tuned for this particular task. Some types of memes or queries can be challenging because they modify words for the purpose of the joke, resulting in terms that do not exist in the word embedding vocabulary. For now, these out-of-vocabulary terms are ignored, but this behavior is not ideal.

Inappropriate content detection. The current search index has been built by selecting 3 topics that, in the author's views, are less prone to containing inappropriate content. In the real world, however, the results containing sexual, violent or hateful content should be detected and filtered to protect users from certain audiences. The detection can serve to completely hide these contents from the search engine or, at least, let the users decide whether or not they want to see these contents. Frameworks such as KAREN[14] can be used to detect hate speech in memes and, therefore, alleviate some of these issues. This is an important concern, as shown by the proposal of the Hateful Memes challenge [9].

Multi-modal joint representation. While the representation of a meme in the search index is multi-modal, the different modalities are considered separately. A future research direction can be to combine all the available information. The search ranking could be further improved by designing a neural network architecture that performs feature fusion of all the different modalities to learn a joint representation of the memes. Incorporating multi-modal data relationships can be beneficial to provide search results that potentially capture the true meaning of the memes, becoming more consistent and relevant to human users.

Acknowledgments

I would like to thank my company[15], for providing the web server, subdomain and GPU resources to make this Search Engine avaliable to the public internet. Thank you to my classmates from the Web Information Retrieval course and other colleagues for their valuable feedback during the early testing.

References

[1] Songrit Maneewongvatana and David M. Mount. "On the Efficiency of Nearest Neighbor Searching with Data Clustered in Lower Dimensions". In: *Proceedings of the International Conference on Computational Sciences-Part I*. ICCS '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 842–851. ISBN: 3540422323.

[2] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[3] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[4] Baoguang Shi, Xiang Bai, and Cong Yao. *An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition*. 2015. arXiv: 1507.05717 [cs.CV].

[5] Hamid Palangi et al. "Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.4 (Apr. 2016), pp. 694–707. ISSN: 2329-9304. DOI: 10.1109/taslp.2016.2520371. URL: http://dx.doi.org/10.1109/TASLP.2016.2520371.

[6] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs". In: *arXiv preprint arXiv:1702.08734* (2017).

[7] Youngmin Baek et al. *Character Region Awareness for Text Detection*. 2019. arXiv: 1904.01941 [cs.CV].

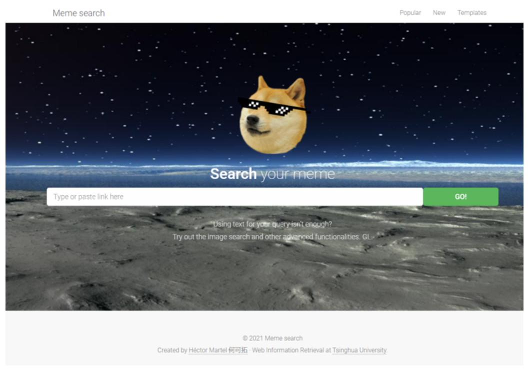
[8] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[9] Douwe Kiela et al. *The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes*. 2021. arXiv: 2005.04790 [cs.AI].

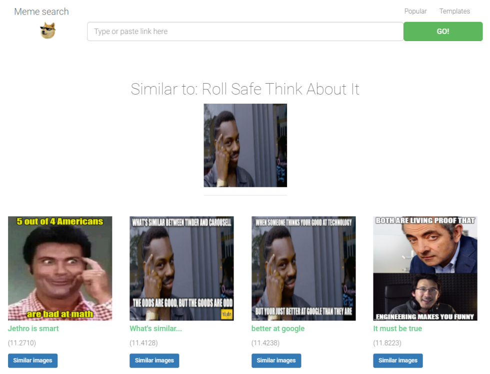
[14] KAREN: Hatespeech Detection and Benchmarking: https://github.com/TiagoMAntunes/KAREN

[15] thekaosmusic.com - AI for Music Production: https://www.thekaosmusic.com

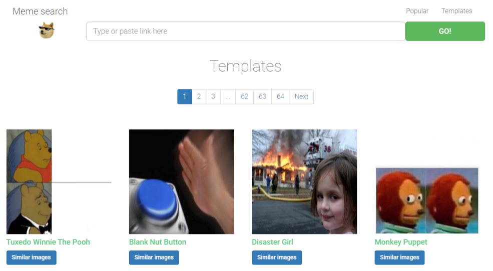
A Screenshots


Meme search
Popular
New
Templates
Search your meme
Type or paste link here
GO!
Using text for your query isn't enough?
Try out the image search and other advanced functionalities. GL
© 2021 Meme search
Created by Héctor Martel 何可拓 · Web Information Retrieval at Tsinghua University

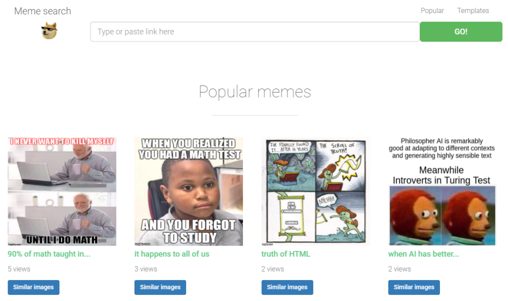
(a)


Meme search
Popular
Templates
Type or paste link here
GO!
Similar to: Roll Safe Think About It
5 out of 4 Americans
are bad at math
WHAT'S SIMILAR BETWEEN TINDER AND CAROUSELL
THE ODDS ARE GOOD, BUT THE GOODS ARE ODD
WHEN SOMEONE THINKS YOUR GOOD AT TECHNOLOGY
BUT YOUR JUST BETTER AT GOOGLE THAN THEY ARE
BOTH ARE LIVING PROOF THAT
ENGINEERING MAKES YOU FUNNY
Jethro is smart
(11.2710)
What's similar...
(11.4128)
better at google
(11.4238)
It must be true
(11.8223)
Similar images

(b)



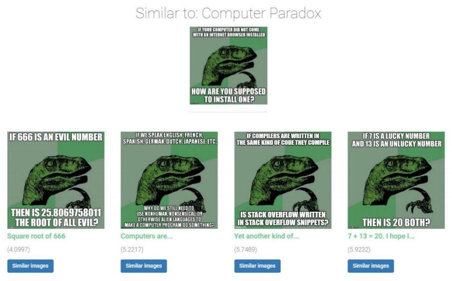
(c)



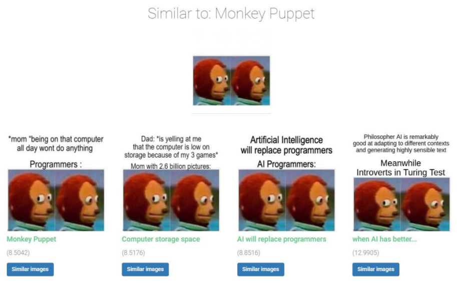
(d)

Figure 7: Main screens of the web application. (a) Home page. (b) Search results page. (c) Templates page. (d) Popular memes page.

B Features


Similar to: Computer Paradox
IF YOUR COMPUTER DID NOT COME WITH AN INTERNET BROWSER INSTALLED
HOW ARE YOU SUPPOSED TO INSTALL ONE?
IF 666 IS AN EVIL NUMBER
THEN IS 25.8069758011 THE ROOT OF ALL EVIL?
IF WE SPEAK ENGLISH, FRENCH, SPANISH, GERMAN, DUTCH, JAPANESE, ETC,
WHY DO WE STILL NEED TO USE NONHUMAN, NONSENSICAL, OR OTHERWISE ALIEN LANGUAGES TO MAKE A COMPUTER PROGRAM DO SOMETHING?
IF COMPILERS ARE WRITTEN IN THE SAME KIND OF CODE THEY COMPILE
IS STACK OVERFLOW WRITTEN IN STACK OVERFLOW SNIPPETS?
IF 7 IS A LUCKY NUMBER AND 13 IS AN UNLUCKY NUMBER
THEN IS 20 BOTH?
Square root of 666
(4.0997)
Computers are...
(5.2217)
Yet another kind of...
(5.7489)
7 + 13 = 20. I hope I...
(5.9232)
Similar images

(a)


Similar to: Monkey Puppet
*mom "being on that computer all day wont do anything
Programmers :
Dad: *is yelling at me that the computer is low on storage because of my 3 games*
Mom with 2.6 billion pictures:
Artificial Intelligence will replace programmers
AI Programmers:
Philosopher AI is remarkably good at adapting to different contexts and generating highly sensible text
Meanwhile Introverts in Turing Test
Monkey Puppet
(8.5042)
Computer storage space
(8.5176)
AI will replace programmers
(8.8516)
when AI has better...
(12.9905)
Similar images

(b)

Figure 8: Similar images feature. (a) Similar to query meme (image + text). (b) Similar to query template (image without text). In both cases, the top image is used to as a query to retrieve others with similar visual content and different text captions. The same function is adapted to memes and templates by setting different relevance thresholds.

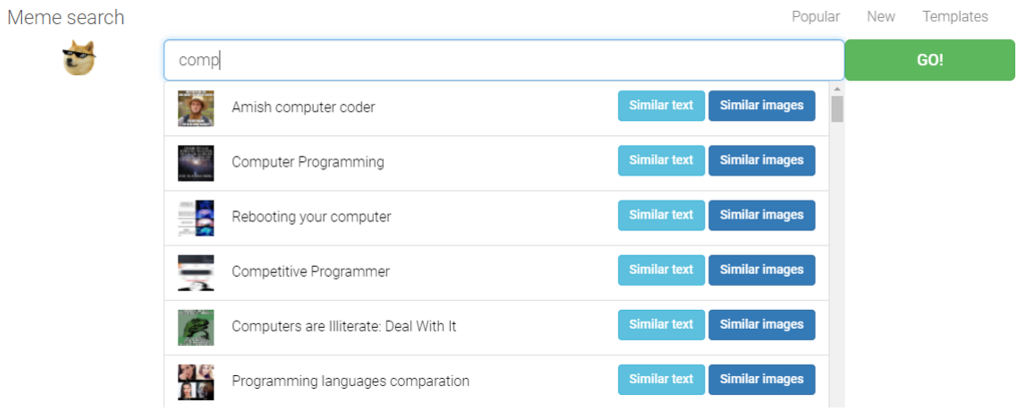


Figure 9: Input suggestion feature. On each suggestion, clicking on the title redirects to the detail page of that particular meme. Additionally, memes containing similar text or similar images can be obtained with the buttons on the right.