Golden Retriever: A Real-Time Multi-Modal Text-Image Retrieval System with the Ability to Focus

Florian Schneider
Chris Biemann
florian.schneider-1@uni-hamburg.de
christian.biemann@uni-hamburg.de
Universität Hamburg
Hamburg, Germany

ABSTRACT

In this work, we present the Golden Retriever, a system leveraging state-of-the-art visio-linguistic models (VLMs) for real-time text-image retrieval. The unique feature of our system is that it can focus on words contained in the textual query, i.e., locate and highlight them within retrieved images. An efficient two-stage process implements real-time capability and the ability to focus. Therefore, we first drastically reduce the number of images processed by a VLM. Then, in the second stage, we rank the images and highlight the focussed word using the outputs of a VLM. Further, we introduce a new and efficient algorithm based on the idea of TF-IDF to retrieve images for short textual queries. One of multiple use cases where we employ the Golden Retriever is a language learner scenario, where visual cues for "difficult" words within sentences are provided to improve a user's reading comprehension. However, since the backend is completely decoupled from the frontend, the system can be integrated into any other application where images must be retrieved fast. We demonstrate the Golden Retriever with screenshots of a minimalistic user interface.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Image search**.

KEYWORDS

multi-modal; text-image retrieval system; visio-linguistic models

ACM Reference Format:
Florian Schneider and Chris Biemann. 2022. Golden Retriever: A Real-Time Multi-Modal Text-Image Retrieval System with the Ability to Focus. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531666

1 INTRODUCTION

The famous adage *"A picture is worth a thousand words."* can be interpreted in various ways. One way is to see this as a motivation and problem statement for multi-modal text-image retrieval systems that enable searching images with words, i.e., textual queries. While current solutions, e.g., from popular search engines, work astonishingly well, they lack the ability to focus single words of the query and locate them within the retrieved images. That is, they rank images according only to the whole textual query.

With the Golden Retriever presented in this paper, we propose a solution to put particular focus on a word within the query when retrieving images. Further, we locate and highlight the denoted image for the focused word within the retrieved images.

This feature enables multiple use cases, for example, a multi-modal language learner scenario, where visual cues for difficult words can support a user's reading comprehension. To implement the ability to retrieve and rank images not only for the whole textual query but additionally incorporate focused words within the query, we leverage state-of-the-art multi-modal models. However, these models are computationally heavy, challenging real-time critical applications when searching through a large pool of images. With our system, we propose a solution for this by implementing an efficient preprocessing stage that drastically reduces the number of images processed by the multi-modal retrieval model. As a part of this preprocessing stage, we further introduce a fast new algorithm based on TF-IDF [11] to retrieve images for textual queries.

2 RELATED WORK

There were significant breakthroughs in various computer vision and natural language processing tasks during the last few years [3, 7, 8, 21]. This progress of uni-modal models also led to a great leap forward in multi-modal visio-linguistic models (VLMs), leveraging the power of transformers to work with text and images simultaneously [5, 12, 14, 16]. For content-based text-image retrieval [6, 22], these VLMs learn a metric function $\Phi(Q, I) : \mathbb{R}^{|Q| \times |I|} \to [0, 1]$ that measures the similarity of a textual query Q and image I. The objective is to find the best matching image $I_k = \underset{i \in P}{\operatorname{argmax}}\, \Phi(Q, I_i)$ for the query text Q from a pool of images P.

There are two major differences in the architecture of current VLMs, affecting how the text-image similarity is computed. In so-called early-fusion VLMs, a single transformer stack is employed that simultaneously processes textual and visual token embeddings and computes the text-image similarity from the outputs of the self-attention heads of the last layer. In VLMs referred to as late-fusion models, there are two transformer stacks, one for the textual input and one for the visual input. Late-fusion VLMs calculate the cosine-similarity from the textual and visual CLS tokens or from

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531666

an aggregation of the other token embeddings of the last layer to compute the text-image similarity. Because the complexity of self-attention is quadratic in the number of input tokens, early-fusion models require less computational power or execution time than late-fusion models for inference. However, even with late-fusion models, "real-time" critical applications become challenging to implement when retrieving the best matching images according to a textual query from a large pool of images.

3 MOTIVATION AND CHALLENGES

There are two primary challenges the Golden Retriever system solves, briefly outlined in the following subsections.

3.1 VLMs in "real-time" critical Retrieval Systems

State-of-the-art visio-linguistic models (VLMs) require much computational power to retrieve the best matching images for a textual query from a large pool of images. Hence, it becomes challenging to leverage those VLMs for real-time critical retrieval systems. To solve this issue, the Golden Retriever system implements a sophisticated pre-selection stage that drastically reduces the number of candidate images processed by the VLMs.

3.2 Extending queries by Focus Words

The second motivation of the Golden Retriever is to extend the textual query used in standard text-image retrieval, which comprises a sequence of words by a focus word contained within the sequence. In the following, we refer to the sequence of words in the query as the *context* and the focus word as the *focus*. Then, we retrieve the best matching images according to the context and pay particular attention to the focus word in a re-ranking stage. Further, we locate and highlight the image region where the focus word is best represented in the retrieved images.

4 VISUALLY WEIGHTED TF-IDF

This section introduces an efficient method to retrieve images for textual queries consisting of short noun phrases. Our algorithm is based on TF-IDF [11], but is applied to images instead of textual documents. Hence, we refer to it as Visually-Weighted TF-IDF or VW-TF-IDF. In Section 5.3.2, we describe how we utilize this method to retrieve images relevant to the *focus*.

For the VW-TF-IDF, we interpret images as visual documents with "terms" that are classified Region-Of-Interests (ROIs) in the image predicted by an object detection and classification network, e.g., Faster-R-CNN [18]. In the current Golden Retriever version, we use a pre-trained network [1, 24] with about 1400 unique objects and attributes labels. The set of labels is what we refer to as "visual vocabulary" and each element is called a term in the following.

To compute the VW-TF-IDF score, the classical formula of TF-IDF is extended by a weighting scheme based on visual properties. The motivation is that the score should be higher if the region with the respective label is prominent in the image and the classifier is confident. Hence, the confidence scores and the ROI areas are incorporated in addition to the counts of the terms from the traditional TF-IDF formula.

Formally, we define the VW-TF-IDF of a term t and an image document d as

$$\text{vw\_tf\_idf}(t, d) = \text{vw\_tf}(t, d) \cdot \log\left(\frac{\text{num}_{\text{docs}}}{\text{df}(t) + 1}\right) \tag{1}$$

where the logarithmic term is standard inverse document frequency (IDF) with simple additive Laplace-Smoothing for numerical stability. The visually weighted term frequency (VW-TF) is defined as

$$\text{vw\_tf}(t, d) = \frac{\text{cnt}(t, d) \cdot \text{weight(t, d)}}{\text{num\_terms}(d)} \tag{2}$$

where $\text{cnt}(t, d)$ is the number of times term t appears in document d and $\text{num\_terms}(d)$ is the total number of terms in the document. The weight of the term t in d is defined as

$$\text{weight(t, d)} = \alpha\ \text{conf}(t, d) + (1 - \alpha)\ \text{area}(t, d) \tag{3}$$

$$\text{conf}(t, d) = \frac{1}{\text{cnt}(t, d)} \sum_{t^{(i)} \in t} t^{(i)}_{conf} \tag{4}$$

$$\text{area}(t, d) = \frac{1}{d_{area}} \sum_{t^{(i)} \in t} t^{(i)}_{area} \tag{5}$$

where $t^{(i)}_{conf}$ is the accumulated confidence score, $t^{(i)}_{area}$ is the accumulated area of the ROIs of term $t_{(i)} \in d$, and d_{area} is the total area of the image document d. The parameter α is used to control the importance of the confidence or area of a term in the final weight of t.

To efficiently retrieve the most relevant images for a query, we first compute a VW-TF-IDF index for every term in the visual vocabulary and every image in the set of images to be searched in an offline setting. Then, in the online setting, the most relevant images have the highest VW-TF-IDF for the query and can be retrieved via simple dictionary lookups in the pre-computed index.

One major drawback of our method – and in general TF-IDF – is that the query can only contain terms from the limited visual vocabulary, i.e., the method lacks proper out-of-vocabulary handling. We overcome this issue with a pre-processing step that transforms arbitrary queries to queries that only contain terms contained in the vocabulary. More on this pre-processing step is detailed in Section 5.3.2.

5 SYSTEM ARCHITECTURE

This section describes the Golden Retriever system to solve the main challenges introduced in the previous section. Auxiliary components like, e.g., a static file server for images or components to generate images with highlighted *focus* are not described here.

5.1 User Interface

The minimalistic user interface presented in Section 7 communicates with the Golden Retriever backend via HTTP calls to a REST API. It is implemented as a simple browser plugin to mimic a search engine-like environment using HTML, CSS, and plain JavaScript. However, since the frontend is decoupled from the backend, the Golden Retriever can be easily integrated within other applications.

5.2 Backend Summary

The Golden Retriever backend implements the two-stage retrieval process schematically sketched in Figure 1. The first pre-selection stage (c.f. Section 5.4) reduces the image pool P to a significantly smaller candidate image set that the VLM processes. Note that the image pool comprises images along with their corresponding textual captions, i.e., it contains multi-modal text-image data. The second fine-selection stage (c.f. Section 5.3) leverages a VLM to retrieve the best matching images from the candidate image set according to the extended query and locate the image region that best matches the *focus* word.

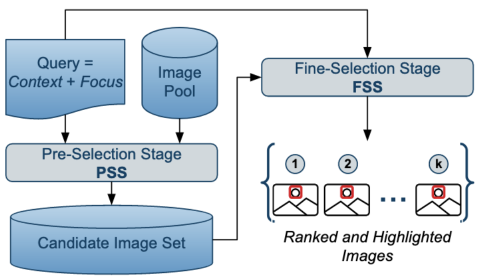


Figure 1: Schematic overview of the Golden Retriever backend system.

In the current version of the Golden Retriever, we use three different multi-modal datasets as image pools: MS COCO [13], Flickr30k [23], and a Wikipedia-based dataset collected by us for other work [20]. Further, we currently employ only TERAN [14] models trained on different datasets in the presented proof-of-concept application. However, we successfully experimented with UNITER [5] models but did not yet implement them in the demonstrated system. Furthermore, in general, every VLM that can compute text-image similarities can be integrated into the Golden Retriever system.

In an extensive experiment described in Section 6 to measure the Golden Retriever backend execution time per request, we found that the average system response is around 2.10 seconds, agnostic to the size of the image pool and the length of the query.

5.3 Pre-Selection Stage

In this stage of the Golden Retriever backend, the image pool is drastically reduced to the candidate image set. Therefore, two efficient sub-procedures are implemented: One selects images relevant to the *context*, and the other selects images relevant to the *focus*. After that, the two resulting sets are merged to obtain the final candidate image set. We first apply the intersection of the *context*-relevant and *focus*-relevant images as the merging operation. If the size of the resulting set is too small, we merge the two sets via union. This size parameter is defaulted to 5000 but can be set by the system administrator. In the following, we briefly describe the two sub-procedures.

5.3.1 Context-based Pre-Selection. To select the *context*-relevant images, we first computed sentence embeddings for every caption of the images in the image pool with an SBert [17] model for semantic textual similarity [4]. Secondly, we clustered the embeddings for efficient searching using FAISS [10] with a quantized Approximate Nearest Neighbor index. Both of these steps are done in an offline setting. Then, in the online setting, we compute the *context* embedding and retrieve the most similar captions in the cluster via cosine similarity. The associated images to the captions are considered *context*-relevant.

5.3.2 Focus-based Pre-Selection. We apply the VW-TF-IDF algorithm introduced in Section 4 to select *focus*-relevant images from the image pool. Since the *focus* part of the query can contain arbitrary words, we need to transform it so that it only consists of terms in the vocabulary of the VW-TF-IDF index. Therefore, we first use a spaCy [9] model for tokenizing the *focus* and obtaining the lemmata of the surface form of the *focus*. Note that the *focus* can comprise more than one token, e.g., for compound nouns or nouns described by adjectives. Next, we retrieve the top-k similar terms from the vocabulary for every *focus* token not contained in the vocabulary. To do this efficiently, we utilize Magnitude [15] with FastText [2] embeddings. The default value for k is set to 10, but can be adjusted per request by the user. In the final step to select the set of *focus*-relevant images, we retrieve the best matching images to the transformed query, i.e., the top-k similar terms, from the pre-computed VW-TF-IDF index.

5.4 Fine-Selection Stage

In this stage of the Golden Retriever backend, we forward the images in the candidate set through a VLM to rank them according to the twofold query. Further, we locate the region that best matches the *focus* part of the query and highlight it with a bounding box. In the current version of the Golden Retriever, we use TERAN, a late-fusion VLM designed for efficient text-image retrieval. The textual input to TERAN are token embeddings computed by a pre-trained BERT [7] tokenizer model. The visual inputs are ROI features extracted with a pre-trained Faster R-CNN [1, 18, 24]. Following the authors, we limit the number of visual tokens per image to 36. Since the query consists of two parts, i.e., the *context* and the *focus*, we compute a score for both parts and apply a weighted average in a re-ranking stage to retrieve the best matching images from the candidate set.

TERAN calculates the global similarity between an image and a textual query by computing a fine-grained word-region-alignment (WRA) matrix $\mathbf{A}$. The cells of $\mathbf{A}$, are the cosine-similarities of the visual regions of the image I and textual tokens of the *context* C are defined as

$$\mathbf{A}_{i,j} = \frac{\mathbf{v}_i^T \mathbf{t}_j}{|\mathbf{v}_i||\mathbf{t}_j|} \tag{6}$$

where $\mathbf{v}_i \in I$ and $\mathbf{t}_j \in C$.

The global similarity, i.e., the *context*-score $s_I^{(C)}$, of an image I and a *context* C is defined as

$$s_I^{(C)} = \sum_{j \in |C|} \max_{i \in |I|} \mathbf{A}_{ij} \tag{7}$$

To specially attend to the *focus* F, we compute a *focus*-score $s_I^{(F)}$ based on the WRA matrix A.

$$s_I^{(F)} = \frac{1}{N * (f_e - f_s + 1)} \sum_{i=0}^{N} \sum_{j=f_s}^{f_e} \mathbf{A}_{ij} \tag{8}$$

where N is the number of regions per image; f_s and f_e are the starting and ending indices of $F \in C$, respectively.

After that, we first normalize and then combine the global similarity (the *context*-score) with the *focus*-score by a weighted average to obtain the final score for the image s_I.

$$s_I = \alpha \cdot s'^{(C)}_I + (1 - \alpha) \cdot s'^{(F)}_I \tag{9}$$

where $\alpha \in [0, 1]$ controls the weighted average and $s'^{(C)}_I$ and $s'^{(F)}_I$ are the normalized *context*-score and *focus*-score, respectively. The default for α is set to 0.5 but can be adjusted by the user per request.

Finally, we sort the images according to their score to rank the candidate image set with respect to the *context* as well as the *focus* part of the query. To locate the region where the *focus* is represented best, we select the ROI with the maximum *focus*-score.

6 "REAL-TIME" CAPABILITY EXPERIMENT

In the following, timings of the Golden Retriever backend system and its sub-components are reported to assess the system's "real-time" capability. Note that "real-time" in the context of our system is always in parentheses because it must not be confused with "true" real-time systems as defined in the context of robotics or real-time operating systems like RTOS[1]. However, there exists a loose definition of "near-real-time" systems, according to which there must not be "significant delays"[2]. As stated in the corresponding Wikipedia article, this "delay in near real-time is typically in a range of 1-10 seconds"[3].

Multiple factors have varying influence on the system's response time. To find how much these factors weigh, the "real-time" assessment test reported in this section was conducted as follows: The system was used with different parameter, query, and dataset combinations. Each of the three queries $Q1$, $Q2$, $Q3$, with 827, 124, 67 characters in *context* length, respectively, was combined with four different modes with the COCO [13], Flickr30k [23], and WISMIR [19] datasets. This results in a set of $3 * 4 * 3 = 36$ different parameter combinations, for which the average system response time over 10 consecutive runs was measured. As it can be observed from the results presented in Figure 2, the length of the context part of the query affects the system's response time the most. This is an expected result since the similarity of an image is based on pooling the word-region-alignment (WRA) matrix, representing the fine-grained similarity of each textual and visual token. Hence, the longer the context, the larger the WRA matrix and the more time the retrieval model needs to generate and pool the matrix.

Further, the effect of the Preselection Stage (PSS) can be noticed: The larger the dataset is, from which the system retrieves the top-k images, the longer the PSS takes, whereas the average response time of the Fineselection Stage (FSS) remains almost across different datasets. Flickr30k has about 31K, COCO about 123K, and WISMIR v2 about 395K images, and the corresponding average PSS response times are 0.09s, 0.27s, and 0.52s, respectively. This increase of time of the PSS is almost linearly proportional to the number of unique images in datasets. These results also highlight the effectiveness of the two-stage retrieval approach of the system.

[1] https://www.freertos.org

[2] https://www.its.bldrdoc.gov/fs-1037/dir-024/_3492.htm

[3] https://en.wikipedia.org/wiki/Real-time_computing#Near_real-time

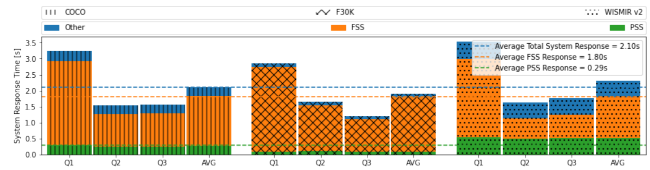


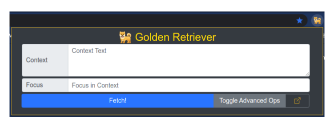
Figure 2: Averaged timing measurements of the system response time for multiple queries $Q1$, $Q2$, and $Q3$ on different datasets. Each bar represents the total system response time, which comprises the response times of different subcomponents. Best viewed digitally with zoom and color.

As depicted in Figure 2, the overall average system response time across all datasets, queries, and modes evaluated in this "real-time" suitability test of the Golden Retriever is 2.10s. Hence, in conclusion, it is considered as an acceptable result.

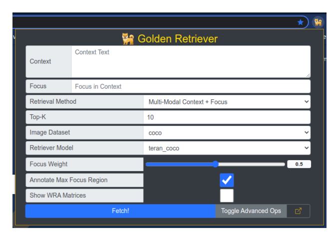
7 SYSTEM DEMONSTRATION

In this section, the Golden Retriever is demonstrated with screenshots of various retrieval examples with different queries using different views of the minimalist user interface.

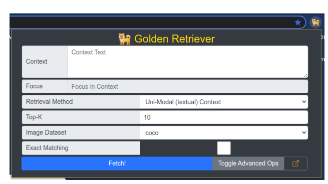
There are four views for different text-image retrieval types supported by the Golden Retriever user interface, shown in Figure 3. The available options and parameters are described in detail on our GitHub page[4]. When the plugin is opened, it shows



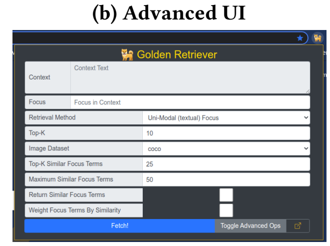
(a) Simple UI



(b) Advanced UI



(c) Advanced UI to trigger a *context* only retrieval



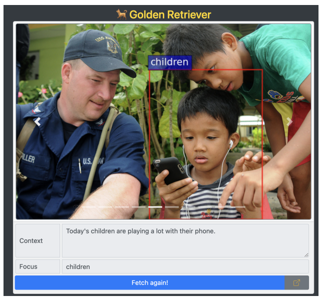
(d) Advanced UI to trigger a *focus* only retrieval

Figure 3: Different views of the minimalistic Golden Retriever user interface. Best viewed digitally with zoom and color.

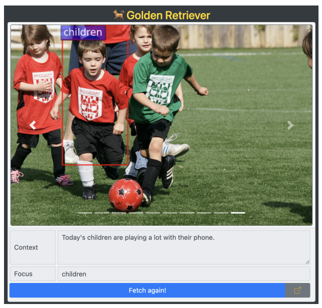
a straightforward interface presented in Figure 3a to retrieve the most similar images for a query consisting of the *context* and *focus* for non-technical users. For research purposes or advanced users, the plugin also offers an interface shown in Figure 3b with more

[4] https://github.com/floschne/MMIRS

options that can be toggled by a button. To retrieve images solely for the *context* (c.f. Section 5.3.1), the UI as shown in Figure 3c is provided. Similarly, if a user wants to retrieve images only for the *focus* (c.f. Section 5.3.2), the UI as shown in Figure 3d is used. Once the top-k images are retrieved, they are presented by an interactive slideshow to the user. The image in full resolution is opened in a new tab by clicking on an image. Figure 4 shows different Golden Retriever results for queries comprising a *context* and a *focus*. In

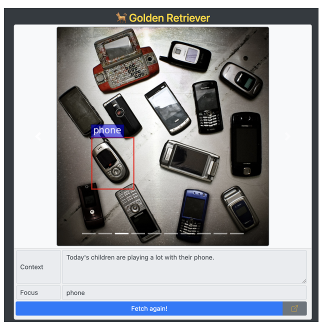


(a) ***focus*** **=** ***children*****;** $\alpha = 0.1$



(b) ***focus*** **=** ***children*****;** $\alpha = 0.9$

(c) ***focus*** **=** ***phone*****;** $\alpha = 0.1$



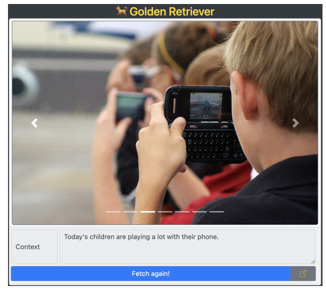
(d) ***focus*** **=** ***phone*****;** $\alpha = 0.9$

Figure 4: Example Golden Retriever results with highlighted ***focus*** **regions for queries with** ***context*** **=** ***"Today's children are playing a lot with their phone."*** **but different** ***focus*** **and** α **values.**

Figure 5 different Golden Retriever results for *context*-only queries (c.f. Section 5.3.1) are shown. In Figure 6 different Golden Retriever results for *context*-only queries (c.f. Section 5.3.2) are shown.

8 CONCLUSION

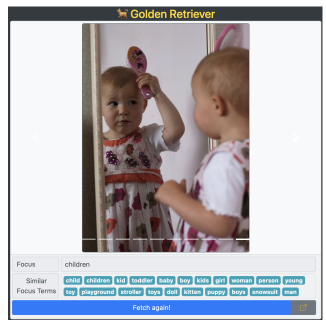
This paper presented the Golden Retriever, a system leveraging state-of-the-art visio-linguistic models for real-time text-image retrieval. The unique feature of our system is that it can focus on words contained in the textual query. To enable real-time capability and the ability to focus, we sketched a two-stage process implemented in the Golden Retriever. Further, we introduced an efficient algorithm based on TF-IDF to find images for short textual queries. To test the "real-time" capability of the system, we conducted an extensive experiment, where we found that the average system response time is in an acceptable range. Finally, we demonstrated the Golden Retriever with screenshots of a minimalistic user interface.



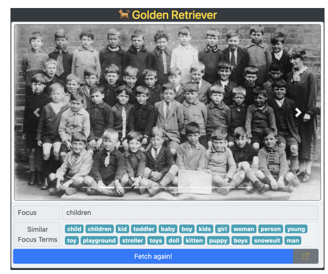
(a)

(b)

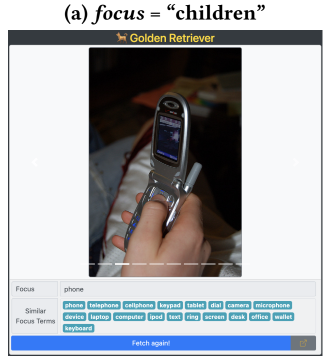
Figure 5: Example Golden Retriever results for queries with ***context*** **=** ***"Today's children are playing a lot with their phone."*** **and no** ***focus***



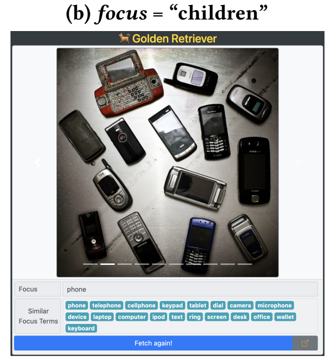
(a) ***focus*** **= "children"**



(b) ***focus*** **= "children"**



(c) ***focus*** **= "phone"**



(d) ***focus*** **= "phone"**

Figure 6: Example Golden Retriever results for queries with different ***focus*** **words but no** ***context*****.**

ACKNOWLEDGMENTS

This research was partially funded by the German Research Foundation – DFG Transregio SFB 169: Cross-Modal Learning.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA, 6077–6086.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics (TACL)* 5 (2017), 135–146.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Virtual, 1877–1901.

[4] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada, 1–14. https://doi.org/10.18653/v1/S17-2001

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *European Conference on Computer Vision (ECCV)*. Online, 104–120.

[6] Paul Clough, Henning Müller, and Mark Sanderson. 2004. The CLEF 2004 cross-language image retrieval track. In *Proceedings of the 5th conference on Cross-Language Evaluation Forum: multilingual Information Access for Text, Speech and Images*. 597–613.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, MN, USA, 4171–4186.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[9] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. https://spacy.io/.

[10] Johnson, Jeff and Douze, Matthijs and Jégou, Hervé. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* (2019).

[11] Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (1972).

[12] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-and-Language Tasks. In *European Conference on Computer Vision (ECCV)*. Online, 121–137.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*. Zurich, Switzerland, 740–755.

[14] Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. 2021. Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 4 (2021), 1–23.

[15] Ajay Patel, Alexander Sands, Chris Callison-Burch, and Marianna Apidianaki. 2018. Magnitude: A Fast, Efficient Universal Vector Embedding Utility Package. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Brussels, Belgium, 120.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning (ICML)*. Online, 8748–8763.

[17] Reimers, Nils and Gurevych, Iryna. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 3973–3983.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39, 6 (2016), 1137–1149.

[19] Florian Schneider. 2021. *Self-Supervised Multi-Modal Text-Image Retrieval Methods to Improve Human Reading*. Master's thesis. University of Hamburg.

[20] Florian Schneider, Özge Alaçam, Xintong Wang, and Chris Biemann. 2021. Towards Multi-Modal Text-Image Retrieval to improve Human Reading. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Mexico City, Mexico (online).

[21] Mingxing Tan and Quoc Le. 2021. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning (ICML)*. Online, 10096–10106.

[22] Christopher Phillip Town. 2004. *Ontology based Visual Information Processing*. Ph. D. Dissertation. University of Cambridge.

[23] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2 (2014), 67–78.

[24] Zhou Yu, Jing Li, Tongan Luo, and Jun Yu. 2020. A PyTorch Implementation of Bottom-Up-Attention. https://github.com/MILVLG/bottom-up-attention.pytorch.