Information Retrieval and Search Engines Project 2020–2021

Aristotelis Chrysakis, Katrien Laenen, Christos Theodoropoulos, Marie-Francine Moens
Department of Computer Science
KU Leuven
firstname.lastname@kuleuven.be

0 Preliminaries

This assignment is only relevant for students following the 6 ECTS version of the course (i.e., with the code B-KUL-H02C8B). The first part of the assignment is due on April 2, 2021, and the second part on May 20, 2021. Note that the submission date of the first part of the assignment is different from what is communicated in the slides of the introduction of the course in order to provide more time to finish the first part. The assignment may be completed in groups of 1 to 3 students.

1 Introduction

Many modern search applications aim to provide relevant results given a query from the user. However, the types of information that can be found online are increasingly diverse — for instance, text, imagery, video, and audio. Thus, an increasingly important task is *cross-modal retrieval*, where the user query and the returned information belong to different modalities. An example could be the retrieval of images for a set of tags or requirements, as might be the case in social networks. Another example is the increasing demand for audio retrieval, where a retrieved song needs to match a user-specified description or genre. In this project, we will study the problem of cross-modal retrieval in practice, using state-of-the-art techniques to achieve the retrieval of relevant results.

In the following, we discuss the problem of cross-modal retrieval in more detail (Section 2), and describe your objectives for the first part (Section 3) and the second part (Section 4) of this project (which will be added here in the coming days). The final two sections provide information on how to submit your work and how your work will be evaluated.

2 Cross-Modal Retrieval

Let us assume we have a collection of images

$$\mathcal{D}_{\mathrm{I}} = \left\{(\boldsymbol{x}_i, \boldsymbol{v}_i)\right\}_{i=1}^{m} \tag{1}$$

of size m, where $\boldsymbol{x}_i$ is an actual image and $\boldsymbol{v}_i$ is a binary vector that describes the image in terms of the presence or absence of certain semantic categories (e.g., the presence of people, or whether the image contains buildings). Similarly, we have a collection of textual captions

$$\mathcal{D}_{\mathrm{T}} = \left\{(\boldsymbol{z}_j, \boldsymbol{w}_j)\right\}_{j=1}^{n} \tag{2}$$

of size n, where $\boldsymbol{z}_j$ is a caption and $\boldsymbol{w}_j$ is a binary category vector that describes the caption. Note that it can be $m \neq n$, or, in other words, we do not assume that image and caption collections are of equal size.

The *cross-modal retrieval* task can be roughly described as follows: given a query in one modality (e.g. text, image, audio, etc.), retrieve the most suitable sample(s) from a different modality. In this project, we specifically focus on the retrieval of images given textual captions. Hence, our problem description can be summarized by the following sentence: Given a textual query $\boldsymbol{z}_j$, retrieve the most similar image(s) $\boldsymbol{x}_i$. This goal can be achieved by learning *projections* $f : \mathcal{I} \rightarrow \mathcal{S}$ and $g : \mathcal{T} \rightarrow \mathcal{S}$ to a common *latent subspace* $\mathcal{S}$, where $\mathcal{I}$ and $\mathcal{T}$ represent the image and text modalities respectively. The main difficulty of this approach is finding an appropriate subspace $\mathcal{S}$. This subspace-learning

task can be viewed as *dimensionality reduction*, which is traditionally achieved using linear methods such as Principal Component Analysis (PCA), Latent Semantic Indexing (LSI), or Canonical Correlation Analysis (CCA). In the context of this project, we will use a different approach which utilizes *neural network* models in order to learn to map the input modalities $\mathcal{I}, \mathcal{T}$ to a shared latent space $\mathcal{S}$ of lower dimensionality (Wang et al., 2016). In the following section, we describe how we aim to do that.

3 Cross-Modal Retrieval via Metric Learning

We will train a neural network model to learn nonlinear projections f and g onto a shared latent space $\mathcal{S}$ that captures the semantics of images and texts. In other words, an image and a similar text description should be embedded closely together in the shared space $\mathcal{S}$, while, on the other hand, an image and a dissimilar text description should lie far apart in $\mathcal{S}$. Such projections can be achieved via neural networks that are optimized with an appropriate loss function.

We will assume without loss of generality that both f and g return d-dimensional real vectors of unit length,[1] or, in other words, $|f(\boldsymbol{x}_i)|_2 = |g(\boldsymbol{z}_j)|_2 = 1$ for all i and j. The loss function we will use is defined as

$$\ell_{\text{IT}} = \sum_{i=1}^{m} \sum_{j=1}^{n} s_{i,j} \left| f(\boldsymbol{x}_i) - g(\boldsymbol{z}_j) \right|_2^2, \tag{3}$$

where $s_{i,j}$ is the cosine similarity of the category vectors $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$ corresponding to the image with index i and the caption with index j

$$s_{i,j} = \frac{\boldsymbol{v}_i^{\mathsf{T}} \boldsymbol{w}_j}{|\boldsymbol{v}_i|_2 |\boldsymbol{w}_j|_2}. \tag{4}$$

Intuitively, minimizing the loss function described by Equation 3 will promote smaller distances between image-caption pairs with more similar category vectors. However, the minimization of this loss function does not promote larger distances between pairs with more dissimilar category vectors. In fact, if the two projections learn to map any of their inputs to exactly the same point in $\mathcal{S}$, the loss function will be zero but, in reality, the network will not have learned anything useful.

To address this shortcoming of Equation 3, we would like to enforce the following constraint:

$$\left| f(\boldsymbol{x}_i) - g(\boldsymbol{z}_j) \right|_2^2 \geq c, \quad \text{when } s_{i,j} = 0, \tag{5}$$

where $c > 0$ is a hyperparameter called the *margin*. In order to incorporate the previous constraint into Equation 3, we rewrite it as follows

$$\ell_{\text{IT}} = \sum_{i=1}^{m} \sum_{j=1}^{n} a_1 s_{i,j} \left| f(\boldsymbol{x}_i) - g(\boldsymbol{z}_j) \right|_2^2 + a_2 \max\left\{0,\ \mathbb{1}[s_{i,j} = 0] \left(c - \left| f(\boldsymbol{x}_i) - g(\boldsymbol{z}_j) \right|_2^2 \right) \right\}, \tag{6}$$

where $\mathbb{1}[A]$ is the *indicator* function, which takes the value one when its argument A is true, or the value zero in the opposite case. Note the addition of the *trade-off* hyperparameters a_1, a_2 between the similarity and dissimilarity terms. The trade-off hyperparameters must satisfy $a_1, a_2 \geq 0$ and $a_1 + a_2 = 1$, so only one of them needs to be explicitly tuned (in the interval $[0, 1]$).

For this project, instead of the raw images, you have access to their corresponding image feature vectors. Moreover, you will use the bag-of-words model to transform raw text descriptions into numerical vectors. The representations of the two input modalities should then be projected into the shared latent space $\mathcal{S}$, using a neural network that implements two projection functions (i.e., one for the image feature vectors and another for the bag-of-words vectors). We will provide you with the image features but you will have to implement the bag-of-words approach yourselves. You may also design the neural network architectures yourself, but we recommend that you start with relatively small and simple architectures. It is very important that you add a nonlinear *activation* function after every layer of a neural network,[2] in order to ensure that it is able to learn a nonlinear transformation.[3]

[1] It suffices to use a regular network followed by an normalization with the L_2 norm of the output vector.

[2] The only exception being the final layer, which does not need an activation function.

[3] For more information on this topic, please refer to 6.4.1 of Goodfellow et al. (2016).

3.1 Dataset and Platform

To train your model you should use the NUS-WIDE-LITE dataset which can be downloaded from the official webpage of the NUS-WIDE datasets (Chua et al., 2009)[4]. The NUS-WIDE-LITE dataset consists of 55,615 images. These images and their features are provided by us. Each image is associated with:

- Raw text tags which constitute the caption for the image.
- A ground truth vector of 81 elements which contains the category labels for the image-caption pair.

Of the images, 27,807 are used as the training set, and the rest as the test set. You can extract a randomly selected small subset from the training set to use as a validation set (i.e., for tuning hyperparameters). Please use the train and test dataset splits exactly as defined in $\{Train,Test\}$*_imageOutPutFileList.txt*. The README file of the data loader describes which dataset files you need to extract the image features, raw text tags and ground truth vectors from. Note that the data loader returns batches of size 16. This means that within one batch, for each image there is one corresponding caption and 15 non-corresponding captions that should be used to calculate Equation 6.

In case you would like to avoid running your code on your personal hardware, we recommend using the Google Colab[5] platform, which supports the PyTorch[6] and TensorFlow[7] deep learning frameworks. If you do use it, please make sure to properly configure your code to run on the GPU, and also to choose the correct hardware accelerator in the Colab notebook (Edit/Notebook settings/Hardware accelerator/GPU).

3.2 First Implementation

Your first goal is to implement the previously described cross-modal retrieval system and evaluate it on the test set of the NUS-WIDE-LITE dataset. There are several ways to evaluate the performance of such a retrieval system. In this project, you are asked to compute the *Mean Reciprocal Rank* (MRR) of your retrieval system, optionally, with a cutoff at 10 (i.e., if the rank of the ground truth image is larger than 10, the reciprocal rank for this query is set to zero). MRR should be computed with respect to the test set—that is, you should rank all test set images for each test set caption query, and subsequently compute the MRR averaged over all test set queries.

3.3 Extending the Loss Function

To further improve the performance of our cross-modal system, we would like to extend the previously-described loss function. The loss function defined in Equation 6 only takes into account inter-modal distances—that is to say, distances between instances of different modalities. Your objective here is to add to the definition of Equation 6 two additional loss components that take into account intra-modal distances—that is, distances between instances of the same modality. These loss components should have the same form as Equation 6, and the same values for a_1, a_2 you used earlier. Your final loss function should look like

$$\ell = \beta_1 \ell_{\text{IT}} + \beta_2 \ell_{\text{I}} + \beta_3 \ell_{\text{T}}, \tag{7}$$

where ℓ_{I} and ℓ_{T} take into account the intra-modal distances for the image and text modalities respectively, and $\beta_1, \beta_2, \beta_3$ are hyperparameters that determine the relative contribution of each loss component to the joint loss. These hyperparameters should satisfy $\beta_1, \beta_2, \beta_3 \geq 0$ and $\beta_1+\beta_2+\beta_3 = 1$. We propose the following greedy strategy for tuning them: Vary only the value of β_1 and set the other two under the assumption that both intra-model loss components carry exactly the same importance (i.e., $\beta_2 = \beta_3$). You should once again calculate the MRR for the cross-modal retrieval system

[4] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

[5] https://colab.research.google.com

[6] https://pytorch.org/

[7] https://www.tensorflow.org/

that uses the extended loss function, and comment on the difference you observe in relation to the performance observed previously.

3.4 Summary

In short, you are expected to:

1. Implement a neural network model that performs projections of the two input modalities onto an appropriate latent space $\mathcal{S}$ (one for the image and one for the text modality).
2. Implement the loss function described in Equation 6 and use the NUS-WIDE-LITE dataset to tune all hyperparameters (i.e., learning rate, the margin m, and the trade-off hyperparameters a_1, a_2).
3. Using the optimal hyperparameter values you found, train your system using the training set and report the MRR metric on the test set (as described in Subsection 3.2).
4. Extend your loss function as described in Subsection 3.3, re-train your system from scratch, and report the new MRR (calculated the same way as previously). Discuss the difference between this MRR and the one you reported earlier.
5. Specifically for the project defense session, you should prepare a script that uses your best-performing model to retrieve and depict the 10 most relevant pictures from the test set, given a caption you will receive at the time.

Part I is due on April 2, 2021. Please provide your code and a short report (no more than one page of text, discounting figures, tables, references, etc.).

4 Cross-Modal Retrieval Using Triplet Losses

4.1 A Short Introduction

Consider a baseline point $\boldsymbol{a}$, called the *anchor*, a *positive* point $\boldsymbol{a}^+$ that is similar to $\boldsymbol{a}$ in some sense, and a *negative* point $\boldsymbol{a}^-$ that is dissimilar to $\boldsymbol{a}$. The general idea behind triplet losses is to enforce that the distance between anchor and positive be minimized and the distance between anchor and negative be maximized. There are many mathematical formulations of triplet losses that try to achieve this goal. In this second part of the project, we will examine three of them, and apply them in the problem of cross-modal retrieval.

4.2 A Basic Cross-Modal Triplet Loss

Let $\boldsymbol{x}$ be an arbitrary anchor image. Then the positive $\boldsymbol{z}^+$ will be the corresponding ground truth caption, and the negative $\boldsymbol{z}^-$ will be another randomly selected caption. Similarly, if the anchor is an arbitrarily selected caption $\boldsymbol{z}$, the positive $\boldsymbol{x}^+$ will be the corresponding ground truth image and the negative $\boldsymbol{x}^-$ another randomly selected image.[8] Note that if the anchor is an image, both the positive and negative will be captions; conversely, if the anchor is a caption, the positive and negative are going to be images. Thus, the cross-modal triplet loss for an anchor pair $(\boldsymbol{x}, \boldsymbol{z})$ can be defined as

$$\begin{aligned}
\ell^*_{\text{tr}} &= \max\left\{0,\ \left|f(\boldsymbol{x}) - g(\boldsymbol{z}^+)\right|_2^2 - \left|f(\boldsymbol{x}) - g(\boldsymbol{z}^-)\right|_2^2 + c\right\} \\
&\quad + \max\left\{0,\ \left|g(\boldsymbol{z}) - f(\boldsymbol{x}^+)\right|_2^2 - \left|g(\boldsymbol{z}) - f(\boldsymbol{x}^-)\right|_2^2 + c\right\} \\
&= \max\left\{0,\ \left|f(\boldsymbol{x}) - g(\boldsymbol{z})\right|_2^2 - \left|f(\boldsymbol{x}) - g(\boldsymbol{z}^-)\right|_2^2 + c\right\} \\
&\quad + \max\left\{0,\ \left|g(\boldsymbol{z}) - f(\boldsymbol{x})\right|_2^2 - \left|g(\boldsymbol{z}) - f(\boldsymbol{x}^-)\right|_2^2 + c\right\},
\end{aligned} \tag{8}$$

[8] A reasonable approach would be to select the negative at random from the mini-batch you are currently processing.

where, as before, we denote the margin hyperparameter by c. Of course, the loss should be computed over all m ground-truth image-caption pairs (here we assume that $m = n$), so its complete form is

$$\begin{aligned}\ell_{\text{tr}} = \sum_{i=1}^{m} &\max\left\{0,\ \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}_i)\right|_2^2 - \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}^-)\right|_2^2 + c\right\} \\ &+ \max\left\{0,\ \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}_i)\right|_2^2 - \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}^-)\right|_2^2 + c\right\}.\end{aligned} \tag{9}$$

4.3 Hard-Negative Sampling

A simple technique to improve retrieval performance when using triplet losses is sampling *hard-negatives*—that is, negative examples that are "close" to be considered positives. Here, you should try to find negatives within your mini-batch that have the smallest latent-space distance to the other modality of the anchor. You should sample hard-negatives both in the image-to-text and in the text-to-image directions.

4.4 Soft-Weighted Triplet Loss

In this subsection, and the next, we will try to exploit the category vectors to further improve retrieval performance. We would like to weight the loss term for each anchor pair by how difficult it is to correctly perform retrieval with this pair. Pairs that are more challenging will receive a larger weight, hence guiding the model to "focus" on them more. The weight of a loss term is computed as the cosine similarity between the anchor and negative category vectors

$$\begin{aligned}\ell_{\text{sw}} = \sum_{i=1}^{m} s_{i,-} &\max\left\{0,\ \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}_i)\right|_2^2 - \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}^-)\right|_2^2 + c\right\} \\ &+ s_{-,i} \max\left\{0,\ \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}_i)\right|_2^2 - \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}^-)\right|_2^2 + c\right\},\end{aligned} \tag{10}$$

where $s_{i,-}$ is the cosine similarity between the category vectors of $\boldsymbol{x}_i$ and $\boldsymbol{z}^-$ respectively, and $s_{-,i}$ is the cosine similarity between the category vectors of $\boldsymbol{x}^-$ and $\boldsymbol{z}_i$ respectively (see Equation 4).

4.5 Soft-Margin Triplet Loss

Finally, we would like to use an adaptive margin instead of a constant one. This choice is motivated by the fact that not all triplets are equally difficult given the current state of the projection networks, thus should not be treated in exactly the same way. The adaptive margin will be computed by once again exploiting category information

$$\begin{aligned}\ell_{\text{sm}} = \sum_{i=1}^{m} &\max\left\{0,\ \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}_i)\right|_2^2 - \left|f(\boldsymbol{x}_i) - g(\boldsymbol{z}^-)\right|_2^2 + c_0 \ln\left(1 + s_{i,-}\right)\right\} \\ &+ \max\left\{0,\ \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}_i)\right|_2^2 - \left|g(\boldsymbol{z}_i) - f(\boldsymbol{x}^-)\right|_2^2 + c_0 \ln\left(1 + s_{-,i}\right)\right\},\end{aligned} \tag{11}$$

where c_0 is a hyperparameter to be tuned, and $s_{i,-}$, $s_{-,i}$ are, as in the previous subsection, the cosine similarities between the category vectors of the anchors and their negatives.

4.6 Additional Evaluation

In addition to the MRR on the full test set, you should also compute the MRR on the small subset of the test set we provide you with (use the 256 images with filenames in the file `NUS_Test_Subset.txt` and their corresponding captions). This subset is approximately 100 times smaller than the full test set so you should be getting relatively better results. Note, however, that it is also a much less realistic evaluation of your system since, in real-life retrieval applications, the answer set typically contains massive amounts of possible answers.

If you wish to further assess whether your system learns a "good" latent space, you could compare its performance with that of a naive system that returns a random image out of the answer set without taking the input caption into account. You could also qualitatively compare the images that your system retrieves to those retrieved by the naive baseline for various captions of your choice.

4.7 SUMMARY

To summarize, using the same neural network architecture implemented in the first part of the project, you are expected to:

1. Integrate in your source code the new dataloader we provide you with.[9]
2. Implement a basic cross-modal triplet loss (Equation 9).
3. Train your model using this loss only and compute the MRR on the test set and its specified subset (see Subsection 4.6).
4. Integrate hard-negative sampling in your training process, re-train, and re-compute the MRR on the test set and its specified subset.
5. Implement the soft-weighted triplet loss (Equation 10), re-train using hard-negative sampling, and re-compute the MRR on the test set and its specified subset.
6. Implement the soft-margin triplet loss (Equation 11), re-train using hard-negative sampling, and re-compute the MRR on the test set and its specified subset.
7. Provide some comments on the differences in retrieval performance between the previous approaches.

Part II is due on May 20, 2021. Please provide your code and a short report (no more than one page of text, discounting figures, tables, references, etc.).

5 SUBMISSION

Source code (in Python, Matlab, C++, Java, etc.) must be submitted in Toledo by May 20, 2021. Make sure to:

- Add comments that explain your code.
- Add a detailed description of how to run your software.
- Add your test examples.
- If asked in the assignment, add documentation with the actual results and comparisons of different models.

6 GRADING

The grade of the assignment will be computed as the average of the grades of its two parts, and it will count for 1/3 of the final course grade.

Your work will be evaluated and graded on the basis of originality, elegance, efficiency (scalability), functionality and performance. Please refrain from fine-tuning on the test set (we will not give you any extra points for that!). On the day of the demo, you will be provided with a set of new queries to test your implementation (of both part I and part II) on. Make sure you create a script, which, given a specific caption (i.e., a sentence that the user inputs), retrieves the 10 most relevant images with respect to that caption.

REFERENCES

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., 2009.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[9] This dataloader generates a representation of the caption by summing up fastText pre-trained word embeddings. For more information on these embeddings please go to https://fasttext.cc/.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5005–5013, 2016.