Master in
Computer Vision
Barcelona

Cross Modal Retrieval

Module: C5

Group: 7

Students: Cristian Gutiérrez
Iñaki Lacunza
Marco Cordón
Merlès Subirà

Index

a) Implement basic Image-to-text retrieval.

b) Implement basic Text-to-image retrieval.

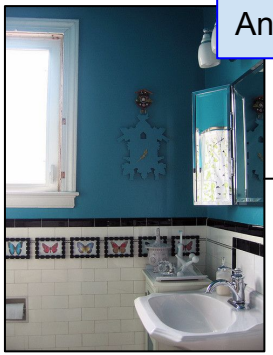
c) Use BERT embedding as Text feature extractor.
   i) Task a) using BERT
   ii) Task b) using BERT

d) Summary slide

Task (a): Implement basic Image-to-text retrieval.

Training instances

Model Employed



Anchor img

'A very clean and well decorated empty bathroom'

Positive caption

'A panoramic view of a kitchen and all of its appliances.'

Negative caption

Faster R-CNN (4096)

Activation ReLU

Embedding Layer (4096,1000)

Fast Text (N,300)

Aggregate N using mean

Activation ReLU

Embedding Layer (300,1000)

Fast Text (N,300)

Aggregate N using mean

Activation ReLU

Embedding Layer (300,1000)

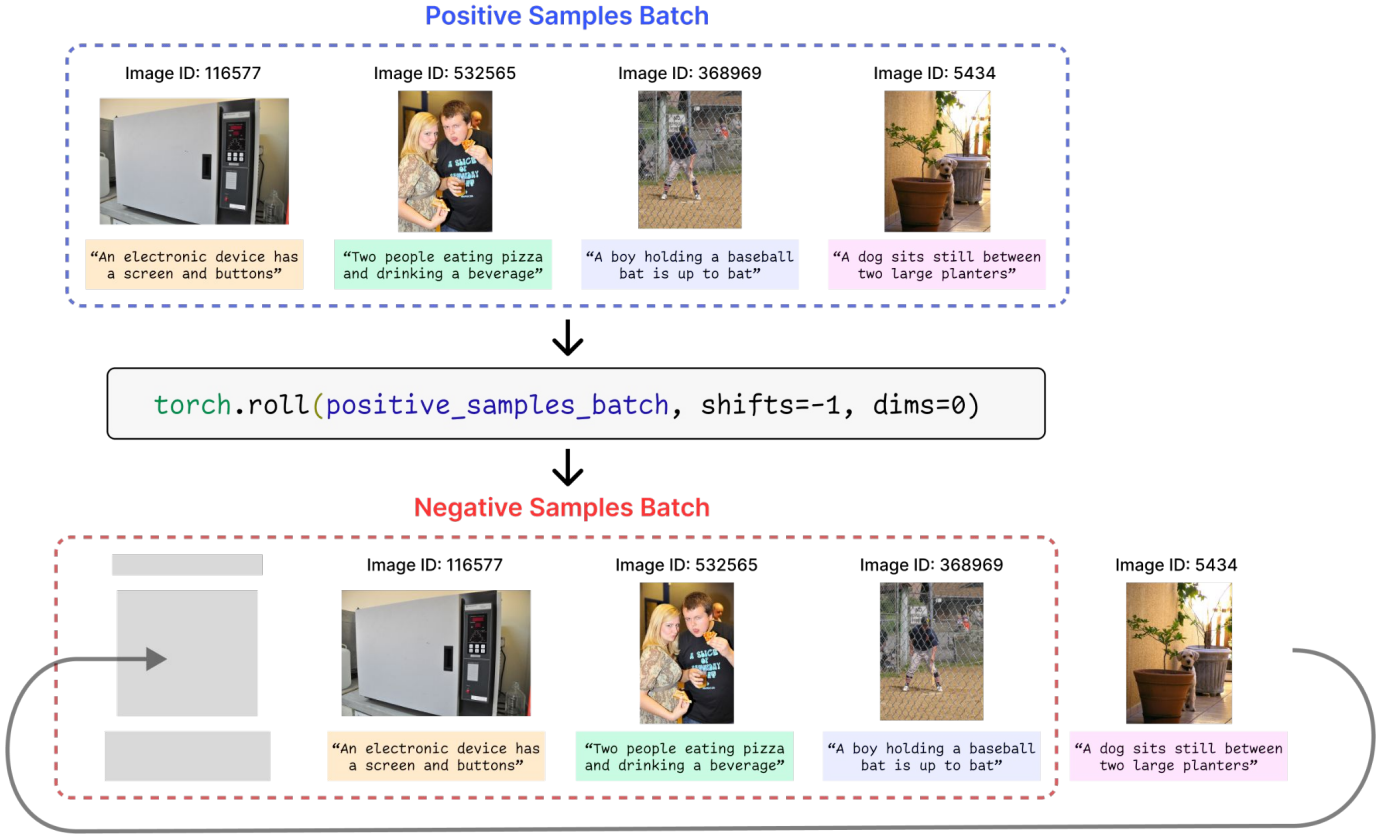
Triplet Loss

Task (a): Implement basic Image-to-text retrieval.

Through all this week work we implemented a simple and yet efficient solution to perform **negative batch-wise sampling** by shifting the batch.

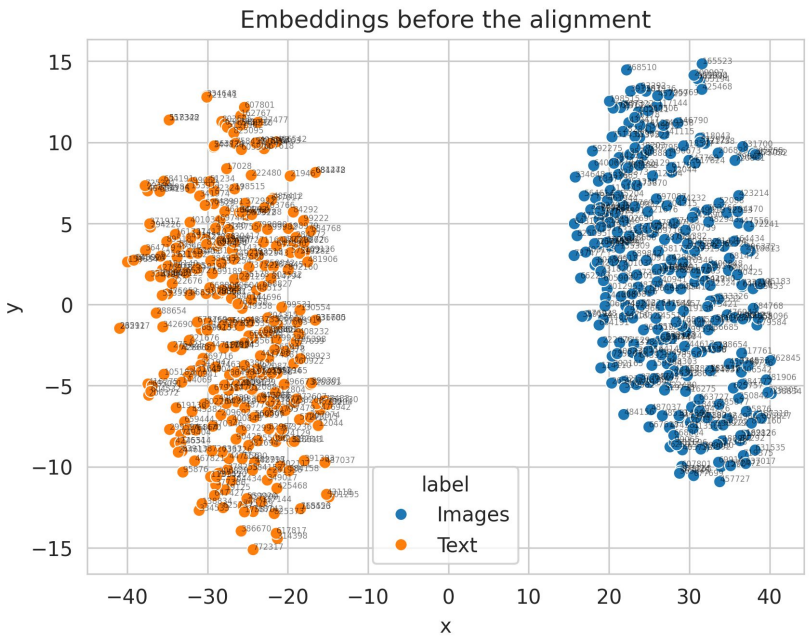
Because it is highly improbable that any subsequent captions are the same, this approach works as expected.

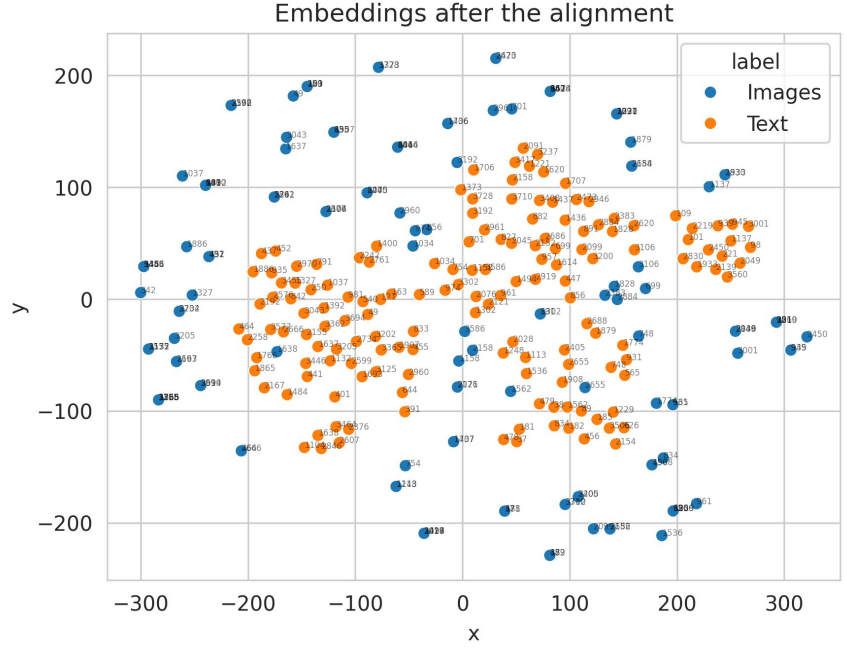
1:1 pos-neg ratio



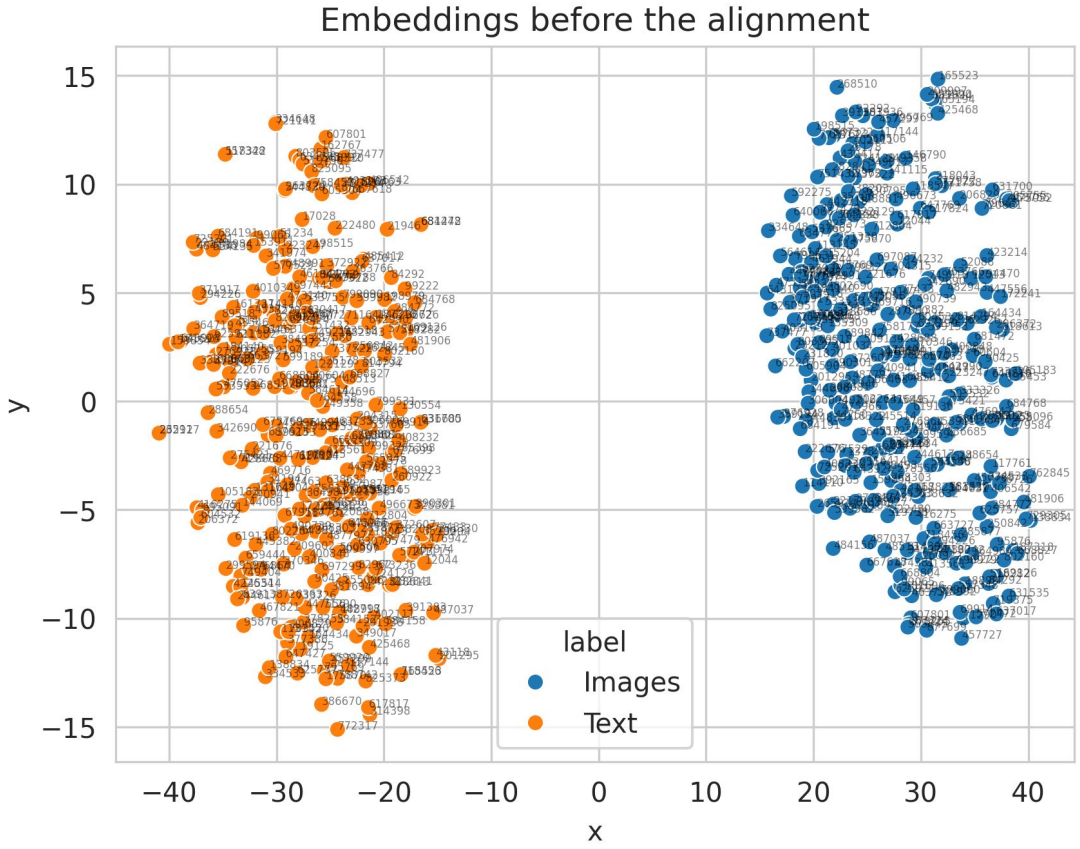
Task (a): Implement basic Image-to-text retrieval.

The embedding layers were trained in 4 epochs, using the entire train set. The used batch size was 32, and *Adam* was employed as optimizer. The triplet loss' parameters were the following: *margin=0.5 , p=2.*
Train time ⏳: 14 hours and 57 mins, using a RTX 3090. Final loss value: 0.079





Task (a): Implement basic Image-to-text retrieval.



TSNE plot BEFORE the alignment:

The caption and image embeddings are clearly differentiated before the alignment.

Obviously, before the alignment, there is no relation between the image and caption of the same instance. Just two large groups differentiating image and caption embeddings.

(300 instances picked randomly from the database were used to draw the TSNE plot)

Task (a): Implement basic Image-to-text retrieval.

TSNE plot AFTER the alignment:

After training the image and text embedding layers, the obtained embeddings are much more related.

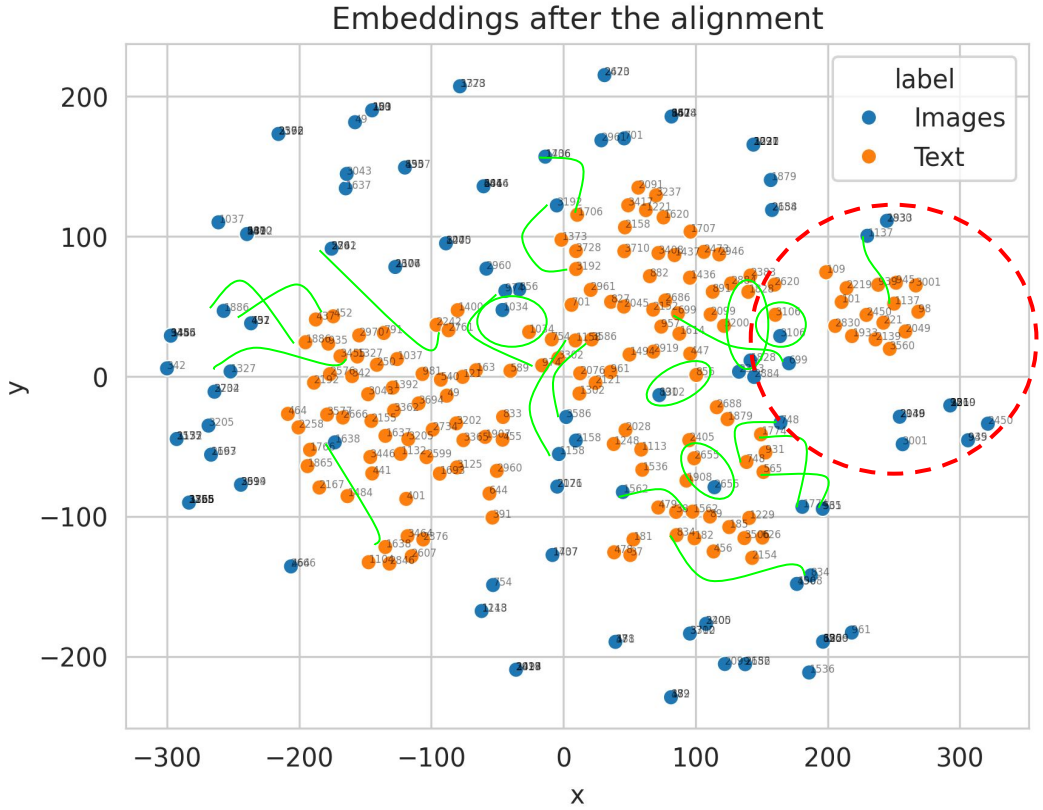
Most of the same indexed embeds (images and captions) are very close, but not overlapping (as shown by the green lines and circles) → Not overfitting!!!

NOTE: The distances shown in the TSNE plot are quite representative but are not the real ones, since the dimensionality has been reduced from 1000 to 2 in order to visualize it.

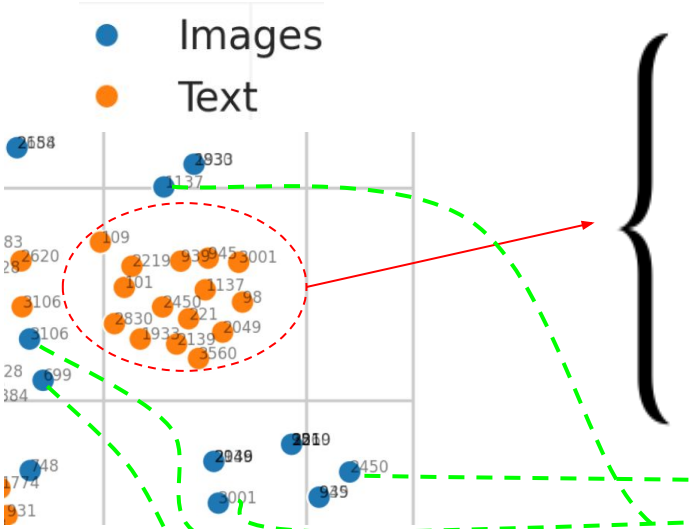
Related instances have ended up nearby

(a closer look will be given to the red circle in the next slide)

(150 instances picked randomly from the database were used to draw the TSNE plot)



Task (a): Implement basic Image-to-text retrieval.



- ID: 2219 → 'An red and white **airplane** is in the **cloudy sky**.'
- ID: 2450 → ' A large U.S Air Force **plain** sits on an asphalt ramp.'
- ID: 101 → 'There is a GOL **plane taking off** in a partly **cloudy sky**.'
- ID: 2139 → 'A **plane** that is **taking off** at an airport.'
- ID: 221 → 'An **airplane** that is, either, landing or just **taking off**.'
- ID: 939 → 'A random **plane** in the **sky flying** alone.'
- ID: 1137 → '**Jet liner flying off** into the distance on an **overcast day**'
- ID: 945 → 'An **airplane flying** high in the blue **sky**.'
- ID: 98 → 'A large passenger **airplane flying** through the **air**.'
- ID: 2049 → 'An **airplane** with its landing wheels out landing.'
- ID: 3560 →'A passenger **plane taking off** into the **sky**. '

ID → 699

sky*, *cloudy

ID → 3106

sky*, *airplane

ID → 1137

sky*, *airplane*, *cloudy

ID → 3001

sky*, *airplane

ID → 2450

sky*, *airplane*, *cloudy

Task (a): Implement basic Image-to-text retrieval.

Image with ID 564133

Ground Truth Caption

Street signs on a pole above traffic lights.

Predicted Captions

A graffiti-ed stop sign across the street from a red car

A vandalized stop sign and a red beetle on the road

A red stop sign with a Bush bumper sticker under the word stop.

The parachutes fly through the sky next to each other.

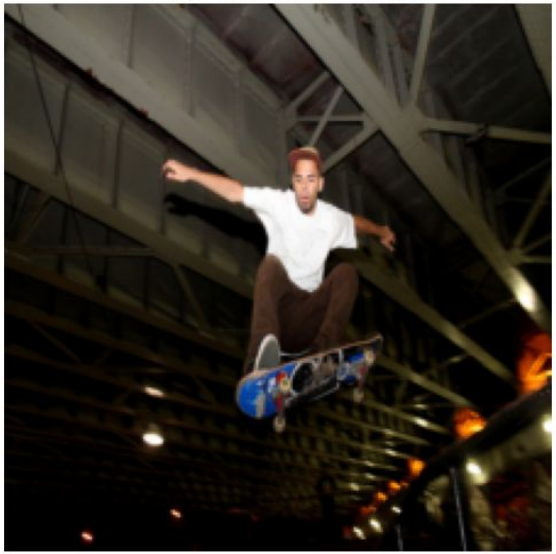
A bike on the platform of a public transportation stop at night.

It is important to remark that like in this task we want the algorithm to return captions when we introduce an image, we have to **fit the KNN using the captions.** Due to memory problems we have employed a database of only 200 images, selected randomly using a Data Loader. So, this could make that the obtained captions won't be very precise due to the **small amount of possibilities.**

As we can see the predicted captions are not 100% accurate, but have a lot of sense because the first three captions talks about road signals and a red object and in our test image there are some road signals and a traffic light that is red.

Task (a): Implement basic Image-to-text retrieval.

Image with ID 506029



Ground Truth Caption

A man in the air on his skateboard doing a trick.

Predicted Captions

A girl trusts another girl to shave her while sitting back to back.

A boy holding his arm around the shoulders of the girl sitting next to him.

A boy places his arm around a friend.

A female student is talking on her phone while walking.

A woman is shaving her face while sitting on a wooden bench.

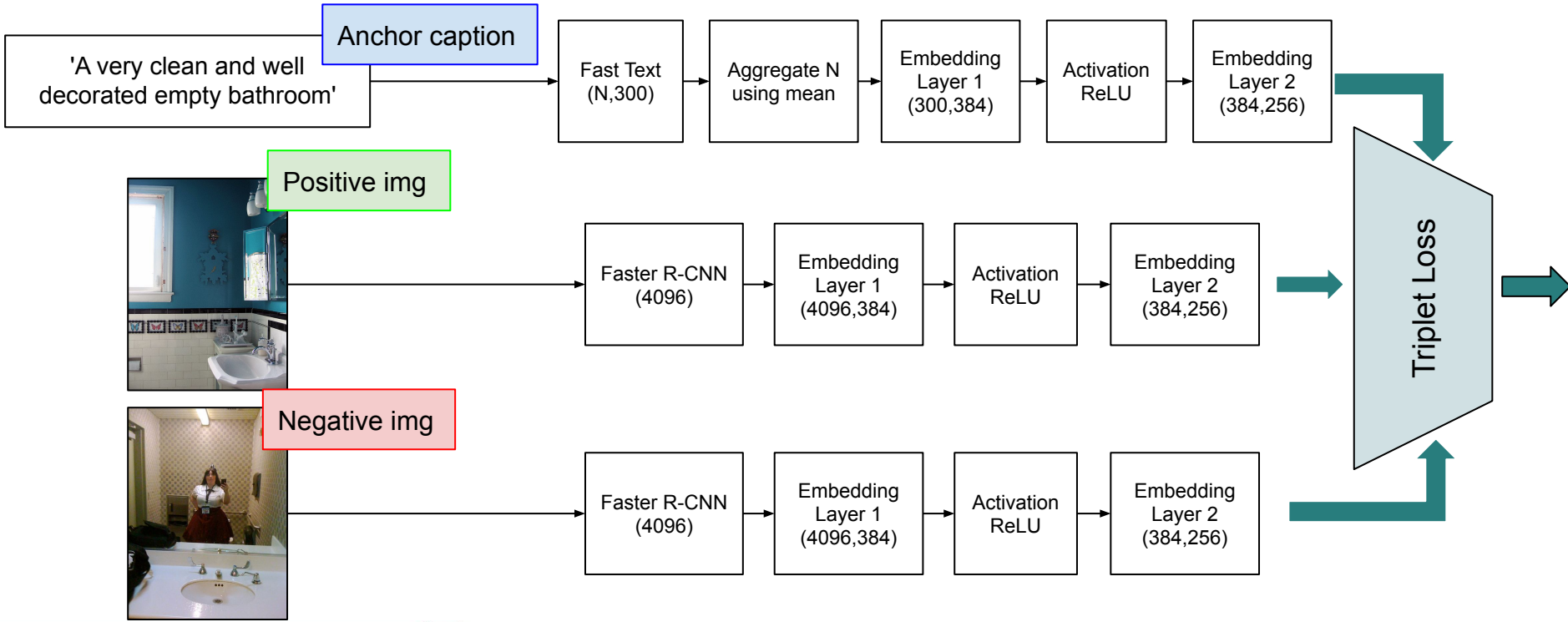
In this case all the top-5 captions talk about actions made by people, so the model is recognizing a person but it is not recognizing well which action is making in the picture.

Task (b): Implement basic Text-to-image retrieval.

Training instances

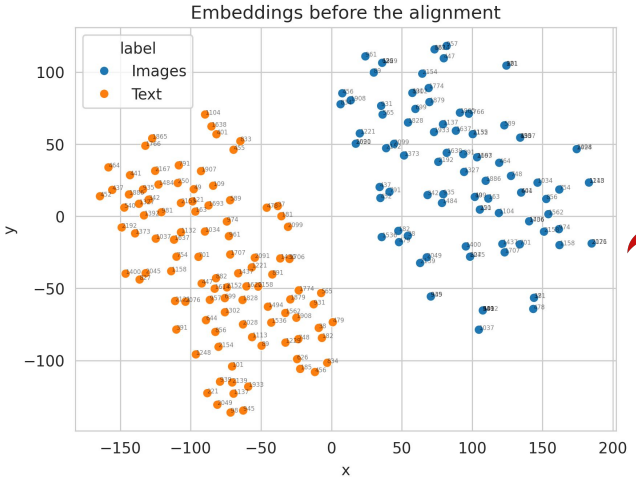
Model Employed

The model employed is different to the previous one since we want to try different approaches, but the training time was very high to try all in all tasks.

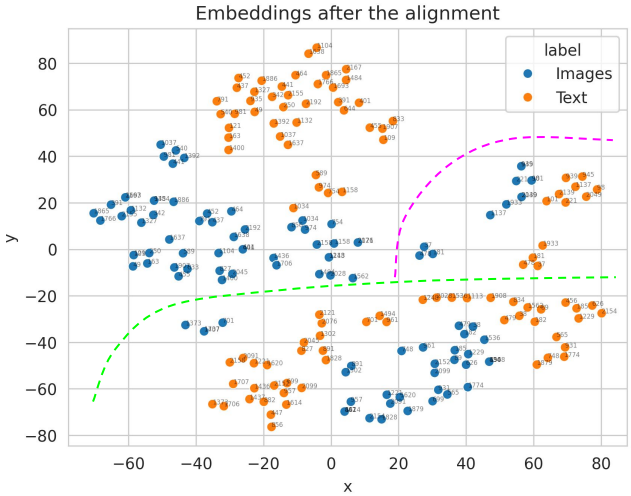


Task (b): Implement basic Text-to-image retrieval.

The embedding layers were trained in 4 epochs, using the entire train set. The used batch size was 32, and *Adam* was employed as optimizer. The triplet loss' parameters were the following: *margin=0.5 , p=2.*
Train time ⏳: 13 hours and 32 mins, using a RTX 3090). FInal loss value: 0.075



(100 instances picked randomly from the database)



(100 instances picked randomly from the database)

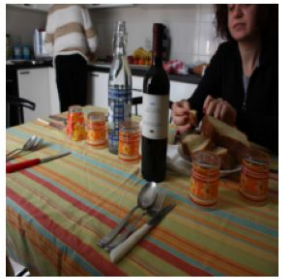
3 main groups have been formed.

Still, it can be seen how even though the embeddings of the images and their corresponding captions can be found very close (taking into account that the TSNE plots does not show the real distance, just a 2D representation), there exists some grouping between embeddings of the same color: a image embedding is found close to its corresponding caption embedding but also to many different image embeddings, and vice versa.8

Task (b): Implement basic Text-to-image retrieval.

Old picture of woman cooking together in the kitchen.

GT Image



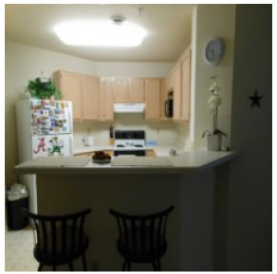
Now, since we want to return images we **have to fit the KNN using images.** Again due to memory problems we only could use 200 images to fit the KNN, so again the results will be a bit inaccurate.

The retrieval are all images of a people group in a kitchen like the caption says, so we can take these results as correct.

Task (b): Implement basic Text-to-image retrieval.

two ladies holding a large bowl filled with doughnuts

GT Image



In this case the retrieval is not so good as the previous, but at least the given images are from a kitchen or food and the caption talks about food. We have to take into account the limited database that we have so it is difficult to find images that are related with that caption.

Task (c): Use BERT embedding as Text feature extractor.

FastText **vs** **BERT**

- **Use of n-grams**
  Reliable for capturing morphological and syntactic information, but not much contextual understanding.

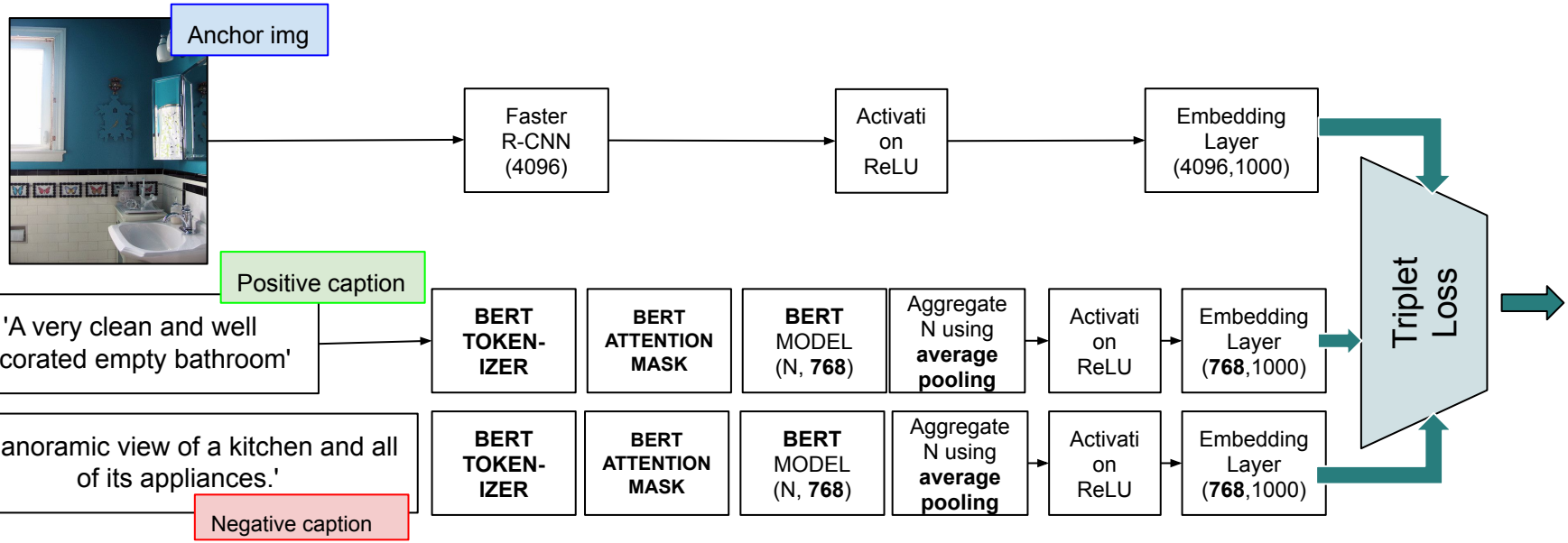
- **Fast and lightweight**
  High efficiency and smaller memory requirement

- **Bidirectional architecture**
  Take into account both preceding and following context of each word
- **Multi-layer transformer architecture**
  Process input sequences of variable length to analyse effectively contextual information.
- **Heavier architecture than FastText**
  Slower and bigger
  memory requirement

Task (c): Use BERT embedding as Text feature extractor.

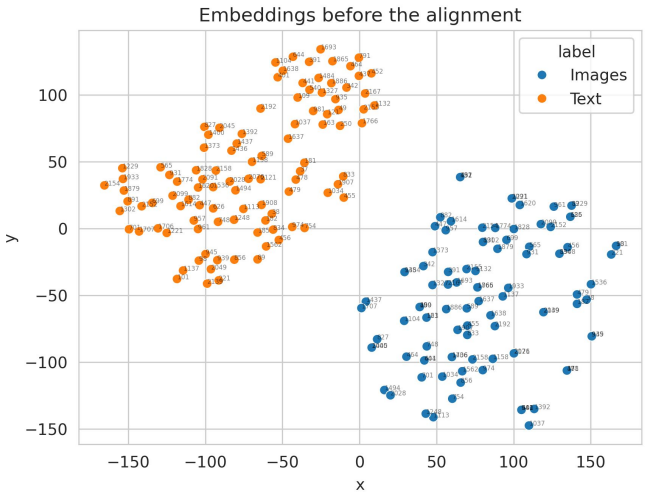
We repeated tasks a) and b) but using BERT instead of FastText.
The followed process was almost the same as earlier, but using some slight changes, which are marked using **bold** letters.
Before using BERT it was necessary to tokenize the caption and apply the given attention mask by the BERT tokenizer.
On the other hand, BERT outputted a embedding of 768 dimensions (instead of 300) for each token (instead of for each word).

TASK a) using BERT:

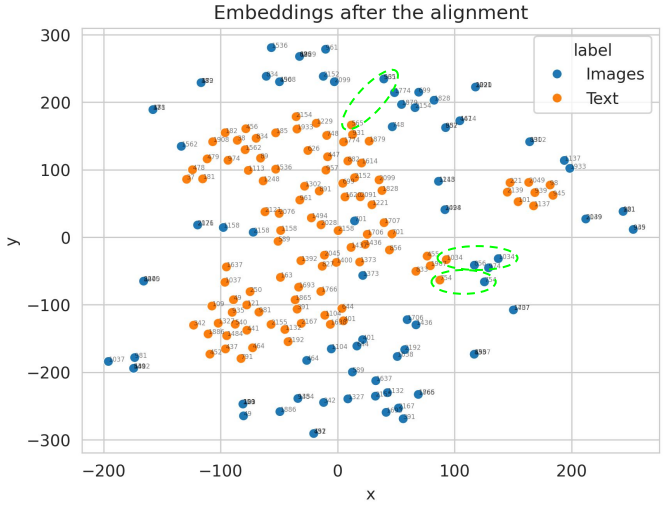


Task (c): Use BERT embedding as Text feature extractor.

The embedding layers were trained in **only 1 epoch**, using the entire train set. The used batch size was 32, and *Adam* was employed as optimizer. The triplet loss' parameters were the following: *margin=0.5 , p=2.*
Train time ⏳: 6 hours and 46 minutes, using a RTX 3090). Final loss value: 0.089 (0.079 in task *a)* using 4 epochs).



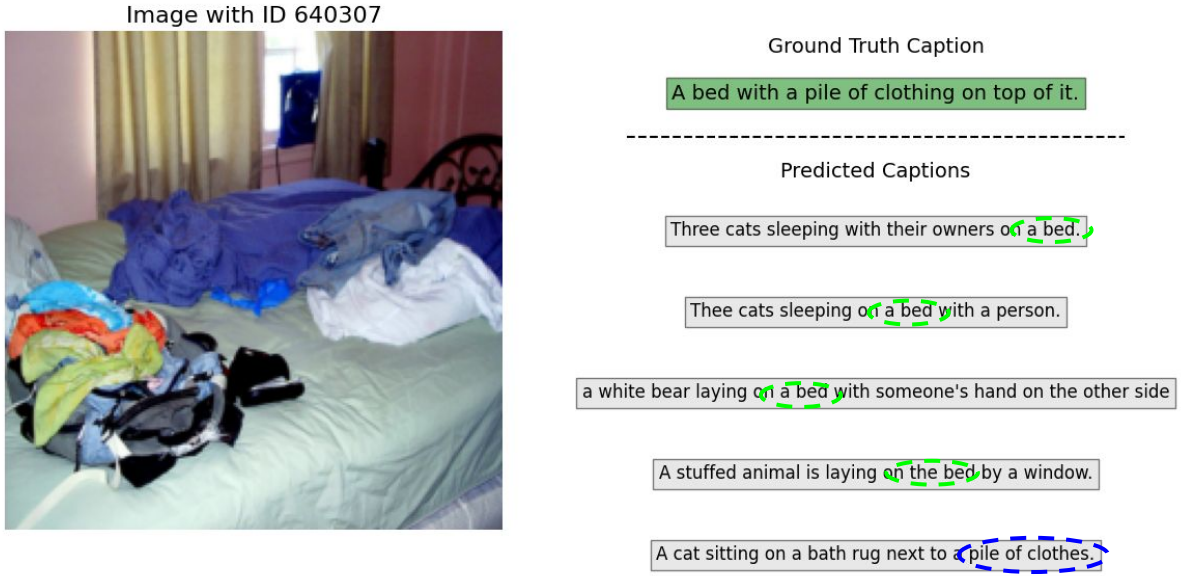
(100 instances picked randomly from the database)



(100 instances picked randomly from the database)

As we can see in the left graphs the results are very similar to task a. After the training the two different nuclei have been mixed in a try to **bring closer the img and the captions with the same id**.

Task (c): Use BERT embedding as Text feature extractor.



The system works very well as we can see in this example. In the image there are a bed and a pile of clothing and all the retrieval captions talk about a bed or a pile of clothes.

Task (c): Use BERT embedding as Text feature extractor.

Image with ID 255866

Ground Truth Caption

a bathroom with a toilet and a sink

Predicted Captions

A wooden toilet seat sits open in an empty bathroom.

White pedestal sink and toilet located in a poorly lit bathroom.

A toilet sits next to a sink in a bathroom.

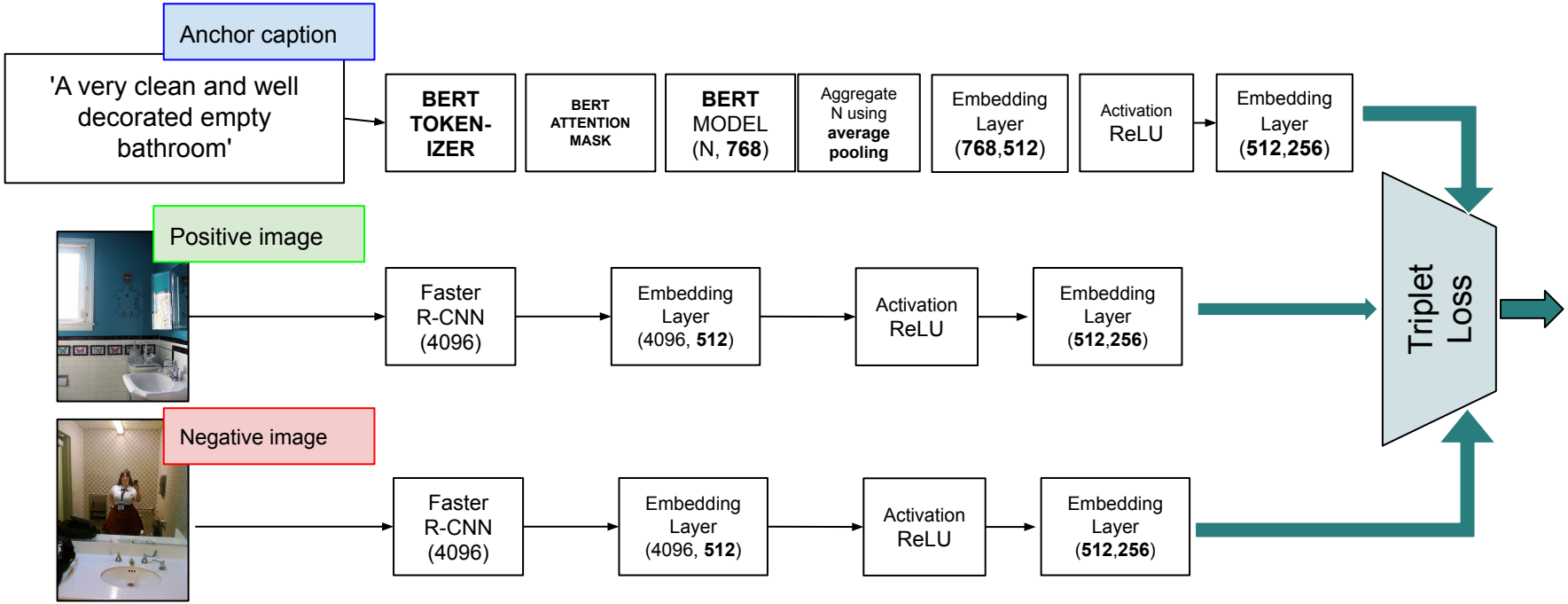
a blue bathroom with a sink and toilet

There is a clean bathroom counter and sink.

Again the five captions are related with the original image. All the phrases talk about a bathroom, a toilet or a sink and those are the unique elements in the image.

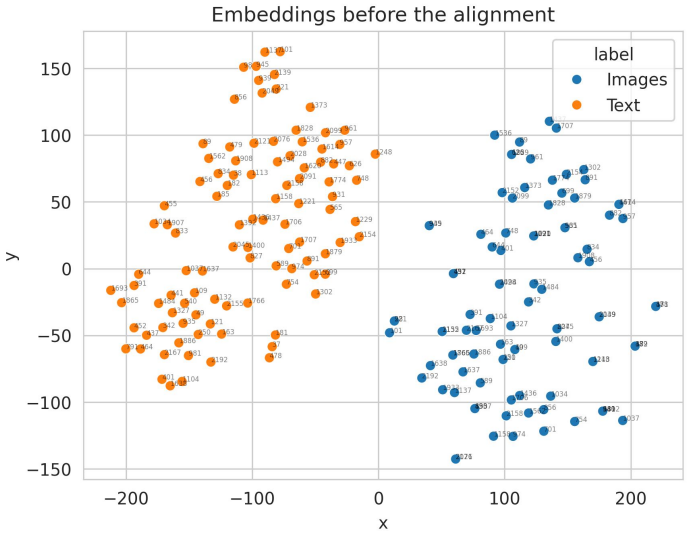
Task (c): Use BERT embedding as Text feature extractor.

TASK b) using BERT:



Task (c): Use BERT embedding as Text feature extractor.

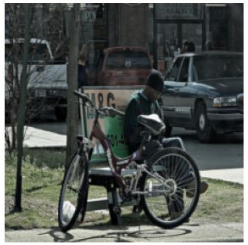
The embedding layers were trained in 1 epoch, using the entire train set. The used batch size was 32, and *Adam* was employed as optimizer. The triplet loss' parameters were the following: *margin=0.5 , p=2.*
Train time ⏳**:** 6 hours and 14 minutes, using a RTX 3090). Final loss value: 0.091 (0.075 in task *b)* using 4 epochs).



Task (c): Use BERT embedding as Text feature extractor.

A group of people on public transportation stare at their phones.

GT Image



Not so good results when working with *task b)*.

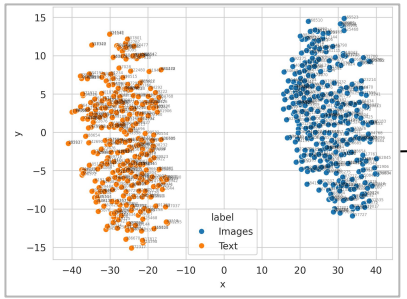
The image on the left contains a man **staring at the phone.**
The image on the middle contains **a group of people**.
The image on the right contains **electronics**, which is related to phones.

But even though it has some slight coincidences we do not think that the obtained results are good enough.

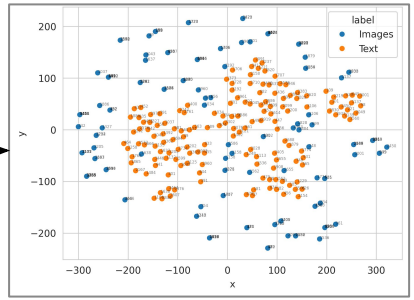
Summary Slide

- This week we have to do a cross modal retrieval algorithm.
- We have **trained** the models using the **400k images of train_COCO.**
- In order to try different things, **2 different approaches** in task a and b:

Before Alignment



After Alignment



Task A: Retrieval by Image

Activation + linear (image 4096 -> 1000) (text 300 -> 1000) (1 step algorithm)

Task B: Retrieval by Text

1 linear + activation + 1 linear (image 4096 -> 384 -> 200) (text 300 -> 384 -> 200) (2 steps algorithm)

- In the second part of the work we have implement **BERT embedding instead of FastText,** obtaining similar but a bit best results than before. Again we have implemented 2 different approaches for the 2 different types of retrievals. (Same as a and b)

Conclusions:

- Even if we train the embedding layers with a large amount of images, the amount of instances we use for fitting the KNN has a very large importance, and because of memory issues we could not use as much as we should.
- The BERT model uses the attention mechanism, so we should get a better contextual understanding. But the obtained results have not been so good.