Exercise 2 - To Embed or Not to Embed ...

Building Word Embeddings with PyTorch/TensorFlow

Deadlines

Deadline for Exercise 2 is **30.10.2022, 23:59 (Zurich Time)**.
Deadline for the peer review is **07.11.2022, 23:59 (Zurich Time).** You will find instructions for the peer review process at the end of this document.
Deadline for feedback to your peer reviewers is **12.11.2022, 23:59 (Zurich Time).**

Learning goals

This exercise introduces you to PyTorch and how we can use it to create our own corpus-specific word embeddings. By completing this exercise you should …

- … understand the basic building blocks for training models in PyTorch.
- … understand what word embeddings are and how you train them.
- … think about how one can evaluate word embeddings.

Please keep in mind that you can always consult and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

We encourage you to hand in your solutions as a Colab-Notebook. **Download your notebook as a .ipynb file**. That way your reviewers can view and execute your code. Or can view your already executed code.

Please hand in your code and your lab report. Hand in the following files and name them exactly in the following fashion:

- ex02_wordembeddings.ipynb
- ex02_labreport.pdf

zip it and name the zip-folder *ex02_ml4nlp1.zip*. The .ipynb files should contain your well documented AND EXECUTABLE code.

We recommend you use Google's Colaboratory, where you have access to GPU time. You can try to solve the exercise on your own computer. However, be aware that training without a GPU may take a long time.

We assume that the data files are in the same folder as the scripts, e.g.

- ex02_wordembeddings.ipynb
- scifi.txt
- tripadvisor_hotel_reviews.csv

Please submit the lab report in PDF format. The lab report should contain a detailed description of the approaches you have used to solve this exercise. Please also include results. In this exercise description, we highlight places in green where we expect a statement about an issue in your lab report.

Please note:

- Your peers need to be able to run your code. If it does not work, you will not be able to obtain the maximum number of points.
- DO NOT submit the data files!

Data

You will work with the complete "Trip Advisor hotel reviews" and "Sci-fi stories" for this exercise. Download the datasets from the material folder in the exercise section of OLAT. The folder contains the files "tripadvisor_hotel_reviews.csv" and "scifi.txt".
The files are also hosted under the following links:

- tripadvisor_hotel_reviews.csv
  https://drive.google.com/file/d/1ihP1HZ8YHVGGIEp1RHxXdt3PPIi12xvL/view?usp=sharing
- scifi.txt
  https://drive.google.com/file/d/10ehW4jZND3QA29v9aNboYUett5-swuNe/view?usp=sharing

The above URL can be used to load the data directly in your Colab notebook - as you already did in Exercise 1.

Part 1 - Train your CBOW embeddings for both datasets

Go to this Colaboratory Python Notebook (homepage from where we took it), create a copy of it and complete the missing code in the exercise section - which means implementing a CBOW model in PyTorch. Then, change the code so that it takes the Trip advisor hotel reviews text as input and produces word embeddings for the hotel reviews.

We strongly recommend (but do not require) an OOP-oriented approach e.g. as presented in Chapter 3 from Rao and McMahan. If you follow Rao and McMahan, that means building the classes for the vectorizer, data loader, etc. Make sure that the code is understandable by either using comments for classes and methods or by explaining the code in text cells of the notebook.

It is up to you to decide on the specific preprocessing steps (removing punctuation, lower-casing, numbers etc.).

1. Describe your decisions (preprocessing, class structure) in the lab report.

You will train two models:

- CBOW2 with a context width of 2 (in both directions) for the **Hotel Reviews dataset**
- and CBOW2 with a context width of 2 (in both directions) for the **Sci-Fi story dataset.**

In order to obtain useful embeddings without training too long we recommend an embedding size of 50 and 12-15 epochs on the hotel reviews dataset. On the Sci-fi dataset 2 epochs are enough. And if you want, you can always try larger embedding sizes with more epochs ;)

(**optional**) - Train CBOW5 with a context width of 5 (in both directions). Are predictions made by the model sensitive towards the context size?

(**optional read**) - This paper (here) provides an insight on how to choose a minimum embedding size while still obtaining useful representations.

Note: Training may take multiple hours. If you have been inactive in the Colab environment for a certain period of time Colab may regard your user session as idle and disconnect - which disrupts the training. To prevent that from happening tricks like this help.

Part 2 - Test your embeddings

Word embeddings are not easy to evaluate automatically without suitable test sets. For this exercise, we inspect them manually to get a feel for whether they capture what we think they should capture. Computing the nearest neighbours (see below) for a word allows us to get an intuition on the semantic vector space. Do the following for CBOW2 and, optionally, for CBOW5:

2. For the hotel reviews dataset choose 3 nouns, 3 verbs, and 3 adjectives. Make sure that some of the nouns/verbs/adjectives occur frequently in the corpus and that others are rare. For each of the 9 chosen words, retrieve the 5 closest words according to your trained CBOW2 model. List them in your report and comment on the performance of your model: do the neighbours the model provides make sense? Discuss.
3. Repeat what you did in 2. for the Sci-fi dataset.
4. How does the quality of the hotel review-based embeddings compare with the Sci-fi-based embeddings? Elaborate.
5. Choose 2 words and retrieve their 5 closest neighbours according to hotel review-based embeddings and the Sci-fi-based embeddings. Do they have different neighbours? If yes, can you reason why?
6. (**optional**) What are the differences between CBOW2 and CBOW5 (if trained)? Can you "describe" them?

Function for nearest neighbour computation

```
import torch.nn as nn

def get_closest_word(word, topn=5):
    word_distance = []
    emb = net.embeddings_target
    pdist = nn.PairwiseDistance()
    i = word_to_index[word]
    lookup_tensor_i = torch.tensor([i], dtype=torch.long)
    v_i = emb(lookup_tensor_i)
    for j in range(len(vocabulary)):
        if j != i:
            lookup_tensor_j = torch.tensor([j], dtype=torch.long)
            v_j = emb(lookup_tensor_j)
```

```
            word_distance.append((index_to_word[j], float(pdist(v_i, v_j))))
    word_distance.sort(key=lambda x: x[1])
    return word_distance[:topn]
```

"net" above corresponds to the CBOW class in the Colab notebook. Note that you might have to adapt the code above, so it fits your initialization of the model.

Important

Please make sure you run your code on Google Colab with GPU selected. Make the GPU selection from "Edit" → "Notebook Settings" and then make the GPU hardware accelerator.

Peer Review Instructions

If you are not already registered on Eduflow follow this link https://app.eduflow.com/join/GXHN93 and register with the E-mail address you use for OLAT. Then you should be added to the course page automatically.
As soon as the deadline for handing in the exercise expires you will have time to review the submissions of your peers. You need to do **2 reviews** to get the maximum number of points for this exercise.
Here some more rules:

- If you do not submit 2 reviews, the maximum number of points you can achieve is 0.75 (from a total of 1).
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- **All reviews are anonymous: Do not put your name into the python scripts, the lab report or the file names.**
- You must also give your reviewers feedback. The same criteria as above apply.
- Students that consistently provide very helpful feedback can be awarded with a bonus in case they earned less than 6 points in total. Ways to obtain points are thus the following:
    - 5 exercises = 5 points
    - 1 presentation or research paper dissection = 1 points
    - consistently good reviews = 1 point

Groups:

- You can create groups of two to solve the exercise together.
- Both students should submit the solutions separately.
- If you did not already work together for the previous exercise, write a small post in the "Groups"-thread in the exercise forum on OLAT to notify the instructors about the group.
- As a group member, you still have to review two submissions with your own eduflow account. However, you may work together in the group to write all 4 reviews.