Natural Language Processing VU (K706.230)
SS24
Practicals - Default Project

March 2024

Q&A Session:	18.04.2024
Deadline Stage 1:	09.05.2024 (just for bonus points)
Deadline Stage 2:	30.05.2024 (just for bonus points)
Deadline Stage 3:	27.06.2024
Final Deadline:	27.06.2024 (report & source code for all 3 stages/tasks)
Submission:	via NLP Dropzone (https://cloud.tugraz.at/index.php/s/2NfASLnwerAYWFW)
Group Size:	2-3 per default. Send an email to us if you want to work in a larger/smaller group.

1 Introduction

For the practicals, we have to know about two important concepts: **Text generation** and **text style transfer**.

Text generation is used, if we want to create more text. For example given 100 pages of a certain book, create 10 more pages. This problem can then be tackled by many different approaches like statistics, n-gram models, hidden Markov models, long-short-time models (LSTMs), your own algorithm, ...

Text style transfer is used, if we want to keep the meaning of a sentence, but change the style. For example given 100 pages of a certain book, written by writer W_1, rewrite those 100 pages in the style of writer W_2, such that it seems that the book is written by W_2. As you might imagine, this leads to a problem that one can easily fool people (=fake news, politician p has said "..."). This problem is quite hard to solve and most likely requires you to use an LLM - but it would be exciting if you came up with something on your own to solve this.

We will now explore ways to fuse those two approaches. This should sensitize you to see, how easy it is to create hardly distinguishable text and how fast this

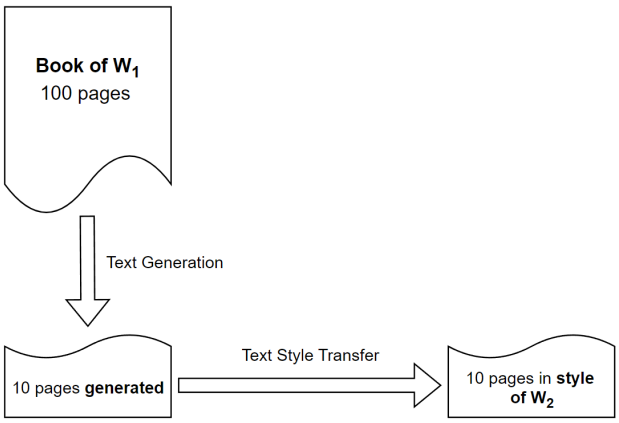

Figure 1: Overview of the fusion between text generation and text style transfer for book data.

technology has evolved. We will do this, by 1. generating new text (here you could try to inject some words you want to be written about if you are interested) 2. applying text style transfer. See Figure 1 for an overview.

1.1 Submission

You have to hand in

- 2 files (generated text, text style transferred text) for stage 1.
- 4 files (2 generated texts, 2 text style transferred texts) for stage 2.
- 1 file (1 text with the back and forth of Trump/Musk) for stage 3.

Thus you have to hand in 7 files in total before the respective deadline for each stage (to get bonus points). Otherwise, you can hand in the files when you want, but before the final deadline on 27.06.2024. You have to hand in the files for ALL 3 stages, that's also the case if you only do one submission at the end. You can find an example submission with dummy data for each stage on TC 1 month before the deadline.

Please follow the **naming convention**, or else your submission might not be valid. Here group is "group" + your group number (eg group69_stageX_Y). You can also add a custom group name to the end if you want to have more uniqueness on the leaderboard (eg group69_stageX_Y_Fun_Name):

- Stage 1: groupXX_stage1_generation and groupXX_stage1_style.
- Stage 2: groupXX_stage2_generation1, groupXX_stage2_generation2, groupXX_stage2_style1, and groupXX_stage2_style2.
- Stage 3: groupXX_stage3

To get bonus points, you simply have to beat the baseline of a certain stage before the given date. But the content/submissions for everything is the same. The baseline of a stage will be updated once per week. Three days before the respective deadline it will be updated daily.

Additionally, the more people you have in your team, the more different approaches we would like to see. For example, if you are a team of three we would like to see that you solve each stage with a different approach (eg: statistical, n-grams, LLMs).

And just for your information, I plugged the assignment sheet into ChatGPT 4 and generated a solution multiple times, so if I do believe that you copy-pasted the solution / the approach we will have an interview such that I can see that you understood everything. Else there will be no assignment interview.

2 Tasks

For all stages think about how to **evaluate** them. The score on TC does NOT count as an evaluation.

2.1 Stage 1 - Nihilistic Spongebob

In this stage, we want to ease into the wonderful landscape of NLP. Is there a better way to do that than to generate nihilistic literature and then transfer it to the style of Spongebob?

Therefore, this task is quite straightforward. Given the dataset "data_stage1" (can be found on TC) which consists of text data with (roughly) 20000 words, **generate** as many new sentences until you hit (roughly) 2000 new words and save that. This leads to the first results which should be submitted in the file group_stage1_generation.
After that execute a **text style transfer** of the generated sentences (aka in this subtask the input is group_stage1_generation) to fit the style of Spongebob Squarepants. The resulting file should be named group_stage1_style and be submitted as well.

The baseline you have to beat before 25.04.2024 to earn bonus points is called: averaging_tutor_baseline and can be found on TC in the Leaderboard section.

2.1.1 Stage 1 - Hints

To maximize your learning gains and see as many different tools/approaches as possible, I would recommend approaching this task with a statistical model. Because the text in a book is often easier to handle because of the uniqueness

of each writer and the large amount of text in the same style.
For the style transfer, you can use any LLM, you don't have to fine-tune/use prompt engineering here. I will use a simple open-source model for this stage and I won't do any fine-tuning / prompt engineering such that the first baseline is easy to beat.
Also, don't forget to **preprocess** the data as we always do - if necessary.

But that's just a recommendation - as mentioned in the lecture you are free to use whatever approach you want.

2.2 Stage 2 - That Doesn't Sound Right! (or Left)

In this stage, we want to see how we could apply this method in a more complex, yet realistic setting. For that, we use the speeches of Austrian politicians, create fake news (aka **text generation**) and then we will let them swap their political orientation (left / right) by doing a **text style transfer** of the generated text again.

This task is a little bit more complex, but in essence, it's quite similar to stage 1. Given the datasets "data_stage2_1_kogler" (from Werner Kogler) and "data_stage2_2_kickl" (from Herbert Kickl) which are speeches with (roughly) 2500 words. You have to **generate** additional text with (roughly) 250 words. This leads to the files group_stage2_generation1 and group_stage2_generation2.

For the second subtask, you have to use group_stage2_generation1 and group_stage2_generation2 as input and do text style transfer such that it fits the opposite politician. So for group_stage2_generation1 you have to transfer it to the style of Herbert Kickl (resulting in group_stage2_style1) and for group_stage2_generation2 you have to transfer it to the style of Werner Kogler (resulting in group_stage2_style2).

The baseline you have to beat before 30.05.2024 to earn bonus points is called: hidden_tutor_baseline and can be found on TC in the Leaderboard section.

2.2.1 Stage 2 - Hints

Here **text preprocessing** might be quite important because the additional information (eg "(Ruf bei der ÖVP: Das ist aber schon alles, was ihr zusammenbringt!") might be distracting for the model to learn the true style.
For this stage, I would further recommend you to use N-Grams, hidden Markov models, LSTMs or to create your own algorithm. That's because you have less text available and because the speeches might not be as consistent in terms of style.
For the style transfer, it might be good to use fine-tuning or prompt engineering

such that the model has an easier time with German.

As mentioned earlier - those are just recommendations, you are free to do what you want (except plagiarism).

2.3 Stage 3 - Make Twitter... eeeh I mean X, Great Again

For the last stage, we want to push our knowledge even further and tackle a quite challenging task. Here we will work with Twitter / X data, thus we have many short text segments on different topics and maybe with different styles (but written by the same person). Since you are now an expert in text generation and text style transfer, the idea would be to create a generation - text style transfer pipeline which creates a "conversation" of tweets.

Thus given "data_stage3_1_musk" (from Elon Musk) and "data_stage3_2_trump" (from Donald Trump) and an initial tweet in the style of Musk, generate a new tweet in the style of Musk, style transfer it to Trump, generate a new tweet in the style of Trump, style transfer it to Musk, generate a new tweet in the style of Musk... until you have 100 new tweets per person (400 in total (100 tweets + 100 style transferred text for 2 persons = 400)). So in pseudocode, it should look something like Algorithm 1. The results of this pipeline should be saved in "group_stage3", which then consists of 400 tweets and should be in an CSV file. Please check out the dummy submission on TC.

Algorithm 1 Text Generation and Style Transfer

$m \leftarrow$ generate Musk text given an initial tweet
for $i \leftarrow 0$ **to** 100 **do**
    $t \leftarrow$ style transfer to Trump given m
    $tt \leftarrow$ generate Trump text given t
    $mt \leftarrow$ style transfer to Musk given tt
    $m \leftarrow$ generate Musk text given mt
end for

Since this is the last task and the deadline is the final deadline, there will be no bonus points for Stage 3. But you can still try to beat the baseline large_tutor_baseline. :)

2.3.1 Stage 3 - Hints

Since this is the pinnacle of the practicals I would recommend you to use the most advanced methods for this task. Thus using an LLM or finetuning a transformer is the way to go, most likely you need that for both, **generation** and **text style transfer**.

I guess you already know that by heart, but again, that's just a recommendation - you are free to use whatever approach you want.