Data Cleaning & Text Processing

Exercise 1

Pranav, Carolin and Karina
pranav.agrawal@uni-hamburg.de, karina.vida@uni-hamburg.de

1 Data Cleaning

One of the most pain-staking tasks in NLP is cleaning the data. Have a look at this table. One of our companies set up a form to collect the details of the donations of their fundraiser and the collected data is really messy. Here are some of the rows:

Email	**Phone Number**	**Country**	**Donation**
abc@gmailcom	04288221533	USA	20$
xyz[at]gmail.com	+497161768885	Germany	50€
hello@uni-hamburg.de	0232 4495146	U.S.A	25.0
hamburg @uni-hamburg.de	02841371135a	Deutschland	7
g123@gmail.com	02773 919199	HK	500.0 HKD
dayta@co.de	06181965021510293	Hong Kong	$ 70

1. Point out where the data is messing up and suggest the ways to clean it up.

   > Emails are irregularly formatted. Some have entered it properly but others have typos like spaces or '[at]'. There is no fixed format for phone numbers. The countries need to be normalized. The donations are in local currencies and have no fixed format.

2. Write a one-shot prompt for a LLM (Large Language Model) to clean and normalize the country column.

   > In the following data, entries of countries entered by the customers are given. This data is not cleaned and standardized. Your task is to normalize this column.
   >
   > Example: U.S.A and USA will be USA

3. Write a one-shot prompt for a LLM (Large Language Model) to clean and normalize the donation column.

   > In the following data, entries of donations entered by the customers are given. This data is not cleaned and standardized. Your task is to normalize this column. Convert everything to EUR. Assume that everything customers have provided is in local currency.
   >
   > Example: 20$ would be 18.50 EUR

4. The best way to avoid irregularities in the phone number column is to provide a regular expression-based validation. Write a regular expression for a German phone number.

```
^+49\d{9}$
```

5. Similarly, the best way to avoid irregularities in the email column is to provide a regular expression-based validation. Write a basic regular expression for an email.

```
^[^@]+@[^@]+\.[^@]+$
```

6. Once the data is cleaned, you need to feed it to an API which accepts the data in JSON form. Write a prompt with a template to convert this dataset into JSON form.

Convert this data into a JSON form for the API input. Example:

```
{
email: {email_address},
phone_number: {phone_number},
. . .
}
```

2 Regular Expressions and Text Analysis

Your TA, Pranav, has recently started learning German. As a text analysis nerd, he plans to use his NLP skills to improve his German vocabulary. Check out this beginner-level paragraph:

> Der Hund bellt laut. Der große Hund spielt im Garten. Die Katze schlaft auf dem Sofa. Der braune Hund jagt die Katze. Das kleine Kind streichelt den Hund. Die Katze trinkt Milch. Der Hund frisst sein Futter. Die Sonne scheint hell. Das Kind geht in die Schule. Die Schule ist groß. Der Lehrer unterrichtet in der Schule. Das Kind lernt viel. Der Hund wartet zu Hause. Die Familie kommt nach Hause. Der Hund freut sich. Die Katze bleibt auf dem Sofa. Am Abend essen alle zusammen. Der Hund liegt unter dem Tisch.

1. One of the biggest challenges for people who are learning German is to learn the grammatical gendered article of the words (like der, das, die). Write a regular expression to extract the German article followed by the word (for example: Der Hund, die Katze, das kind).

```
[D|d][er|as|ie] [a-zA-Z]+
```

2. It is important for Pranav to learn the most frequent collocations to improve his vocabulary. Write a one-shot prompt and expected output for the LLM to find out at least 4 collocations here.

Prompt: In corpus linguistics, a collocation is a series of words or terms that co-occur more often than would be expected by chance. Using the frequency window method, extract at least 4 good collocations in this paragraph.

An example would be: Der Hund

<para>

Output:

"Der Hund" appears 6 times

"Die Katze" appears 4 times

"Das Kind" appears 3 times

"Die Schule" appears 3 times

3. Pranav will build a German-English vocab list. For that, he wants to extract nouns, verbs and adjectives from the German text and translate them into the English. He also wants to include variants of the verbs and plurals of the nouns in his vocab list. See the below picture of how it should look like. Write a few-shot prompt to output a similar list.

der **Ạnzug,** "-e	suit
aus|gehen, er geht aus, ist ausgegangen *(Abends beim Ausgehen haben wir viel Spaß.)*	to go out *(We have a lot of fun when we go out in the evening.)*
das **Kleid,** -er	dress
die **Kleidung** (Sg.)	clothes
die Krawạtte, -n	tie
der **Pullover,** -	jumper
tragen, er trägt, hat getragen	to wear
die **Bluse,** -n	blouse
die **Jeans,** -	jeans
der **Mạntel,** "-	coat
die **Mützе,** -n	hat

From the given paragraph, make a German-English vocabulary. The first column will be German and the second column will be English. To do this, perform the following steps:

1. Extract the nouns (with articles), adjectives and verbs.
2. For nouns, include the plural forms.
3. For verbs, include the variants (present, past and perfect)
4. Arrange them in columns.

Example:

Das Kleid, -er dress
Tragen, er tragt, hat getragen to wear

3 More Data Cleaning

1. Check out these tweets and texts. What is the best way to clean this data? What tricks would you use to remove repeated letters, emojis and confidential information?

   - SLAYYYYY!!!! ST. PAULI just SLAYED THAT GAME YAAASSS.

     Note: Many solutions exist. You can come up with your own token design. This exercise is inspired by FastAI's data transformation.[a]

     `[allcaps] slay xxrep y 4 xxrep ! 4 st . pauli just slayed that game yas xxrep a 3 xxrep s 4 .`

     [a] `https://fastai1.fast.ai/text.transform.html`

   - 

     Note: Many solutions exist. You can come up with your own token design. This exercise is inspired by FastAI's data transformation.[a]

     `[pizza-emoji] xxrep 25 send [pizza-emoji] xxrep 25 pizza [pizza-emoji] xxrep 25 plz`

     [a] `https://fastai1.fast.ai/text.transform.html`

   - No worries, you can use my password which is katy-perry-flop.

     Note: Many solutions exist. You can come up with your own token design. This exercise is inspired by Microsoft's data anonymization.[a]

     `No worries, you can use my password which is [password].`

     [a] `https://github.com/microsoft/presidio`

   - John Doe, 41, from Hamburg, whose salary is 5000 euros per month, recently found out that he has diabetes.

     Note: Many solutions exist. You can come up with your own token design. This exercise is inspired by Microsoft's data anonymization.[a]

     `[person], 41, from [location], whose salary is 5000 euros per month,`

`recently found out that he has diabetes.`

[a] `https://github.com/microsoft/presidio`

- I recently found out Stephanie and Christina in our school are gay but they haven't told anyone but me.

  Note: Many solutions exist. You can come up with your own token design. This exercise is inspired by Microsoft's data anonymization.[a]

  `I recently found out [person] and [person] in our school are gay but they haven't told anyone but me.`

  [a] `https://github.com/microsoft/presidio`

2. Many times, the texts might contain swear words. Would you consider removing those words? When would you do that, explain with examples.

   Note: Many solutions exist. As long as you take a stance and give a nuanced answer, you will get a point. Here I am giving out the solution which has been well-tested and researched. This is inspired by Dodge et al.[a]

   If I am training the model from scratch, I won't remove any swear words. The model needs to understand the context of the words. If every lewd word is being removed, the model will be extremely sanitized and might be prone to attacks. The blanket removal of swear words can be problematic because:

   (a) It may remove legitimate discussions, cultural expressions, and quoted speech

   (b) Some words considered "swears" in one context may be neutral or meaningful in another

   (c) It could disproportionately remove content from certain communities or dialects that use particular language varieties.

   But for fine-tuning for specific downstream purposes, say for like schools or professional settings, I will remove the swear words from fine-tuning data.

   [a] `https://arxiv.org/pdf/2104.08758`

3. In order, to combat swear words, the earlier versions of OpenAI models used a simple blocklist to clean the data.[1] This blocklist contained many swear words but also contained words like *transgender, twink, homosexual* and so on. How this is problematic? Would you still recommend going for blocklists in cleaning the data?

Note: Many solutions exist. As long as you take a stance and give a nuanced answer, you will get a point. Here I am giving out the solution which has been well-tested and researched. This is inspired by Dodge et al.[a] and Anne Lauscher's paper.[b]

[1] `https://arxiv.org/pdf/2104.08758`

I will not use blocklists at all. Words have contexts! Unless I am sure that I have a specific use case for my AI, I won't use blocklist.

Blocklist might cause problems in downstream applications like harming marginalized communities.

Including words like "transgender", "twink", "homosexual" in blocklists is highly problematic because:

1. It erases content about and by LGBTQ+ communities
2. It removes legitimate discussions about identity and human rights
3. It creates representational harm by making these communities less visible in training data
4. It can perpetuate biases in downstream AI systems

Better approaches could include:

1. More nuanced content filtering that considers context
2. Working with affected communities to develop better filtering approaches
3. Focusing on removing actually harmful content rather than blanket word bans
4. Preserving authentic language use while filtering for genuinely problematic content
5. Regular auditing of filtering systems for potential biases

[a] https://arxiv.org/pdf/2104.08758
[b] https://aclanthology.org/2023.findings-acl.502.pdf