1. LT in general (20% for BS students and MS students)
In this part of the assignment, answer the following questions in writing. Make sure you
reason your answers and cite sources where applicable.
a) Compare the four methods we discussed for data gathering, i.e. collecting data
manually, scraping the internet, using crowdsourcing methods and using NLP tools. What are their pros and cons? Which one would you pick if you were making a corpus from scratch and why?

Manual Data Collection
The pros are we can have custom-tailored data that allows for precise and targeted data collection specific to the project. And that we have **control over the quality since h**umans oversee, ensure relevance and the accuracy in the context of the task.
The cons are that it is time-consuming and costly and that it requires significant human effort, leading to increased labor costs. Even though a pro was that we have control over the quality I also think it is an con since error-prone: Manual methods are subject to human error and bias in both recording and interpretation. Also it **delays data since** the process is slower, which may make the data less actionable in real time. And last the **scalability is very limited since it is** difficult to handle large-scale projects effectively without automation.

(Source: MachineMetrics, https://www.machinemetrics.com/blog/manual-data-collection)

Scraping the Internet
The pros for example it gives us access to massive data like diverse, publicly available information from websites, blogs, and forums. With this method we can also gather large datasets quickly using web scraping tools. It is as well very **cost-effective o**nce set up, we can use us it's tools to collect data at minimal costs. It can also **g**ather real time or frequently updated data for up to date analysis.
The cons er the legal and ethical concerns, scraping may violate website terms of service or copyright laws, leading to potential legal issues. Also that websites often implement anti-scraping measures, such as CAPTCHAs or IP blocking.

(Source: Web Scraping Site, https://webscrapingsite.com/guide/pros-and-cons-of-web-scraping/?utm_source=chatgpt.com)

Crowdsourcing

Pros:

High probability of success: Engaging a broad audience can yield valuable input, enhancing market research and product development.

Cons:

Quality control challenges: Ensuring the reliability and quality of contributions can be difficult due to varying levels of participant expertise.

(Source: ISPO, https://www.ispo.com/en/news/markets/what-crowdsourcing-definition-advantages-tips)

NLP

Pros:

Improved Human-Computer Interaction: NLP enables efficient communication between humans and machines using natural language, eliminating the need for users to learn complex programming languages or specific command structures. This accessibility makes technology more inclusive.

Cons

Algorithmic Bias: NLP systems are trained on data that may not represent the full diversity of the population. This can result in biased or unfair outcomes, especially when NLP tools are used in decision-making processes with high stakes.

(Source: LinkedIn, https://www.linkedin.com/pulse/nlp-pro-cons-in-depth-analysis-bonifas-angkadiredja)

I would pick NLP tools to make a corpus from scratch because they efficiently process large datasets, saving time and effort while ensuring scalability.

b) What consequences can it have if the data we use to train NLP models (and other AI models) are biased, i.e. if they contain prejudice? Name some examples
(either imaginary or real) about a system whose function could be impacted or even not possible in this situation.

If the data used to train NLP models is biased, it can lead to unfair and harmful outcomes. For example, a recruitment system might favor male candidates over equally qualified female ones if the training data reflects past hiring biases. In healthcare, a chatbot trained on biased data might provide better advice to certain groups while neglecting others. Biased systems can also make poor decisions in important areas like law or finance, such as denying loans to certain groups unfairly. This can cause people to lose trust in AI systems. Real-world examples include Amazon’s hiring tool, which showed bias against women, and predictive policing systems that unfairly target minority communities. To avoid these problems, it’s important to use diverse and fair training data and regularly check how AI systems work.

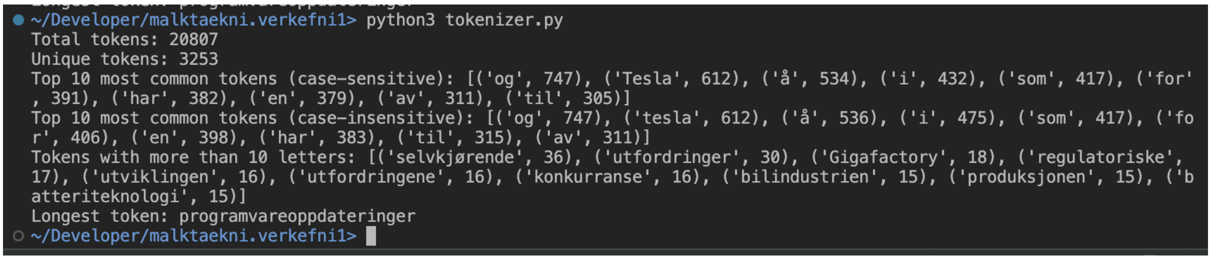
c) Briefly explain the concepts accuracy, precision, recall and f-score in the context of measuring the performance of NLP models.

Accuracy, precision, recall, and f-score are used to measure how well an NLP model performs. **Accuracy** shows how often the model’s predictions are correct overall. **Precision** measures how many of the positive predictions like detecting spam are actually correct, focusing on avoiding false alarms. **Recall** checks how many of the actual positive cases like real spam emails the model successfully identifies, making sure it doesn’t miss too much. **F-score** is a balance between precision and recall, combining them into a single number to show how well the model performs overall when both are important. Together, these metrics help evaluate the model’s strengths and weaknesses.

2. Collect your own data (20% for BS students – 10% for MS students)

For this task, I collected raw textual data in Norwegian about Elon Musk and his company, Tesla. I gathered the text by copying plain text from multiple Norwegian websites that I was familiar with. The resulting dataset is a clean text file containing slightly over 20,000 words. The text is saved in a single .txt file, which is submitted along with this assignment.

3. Design your own tokenizer (30% for BS students – 20% for MS students)



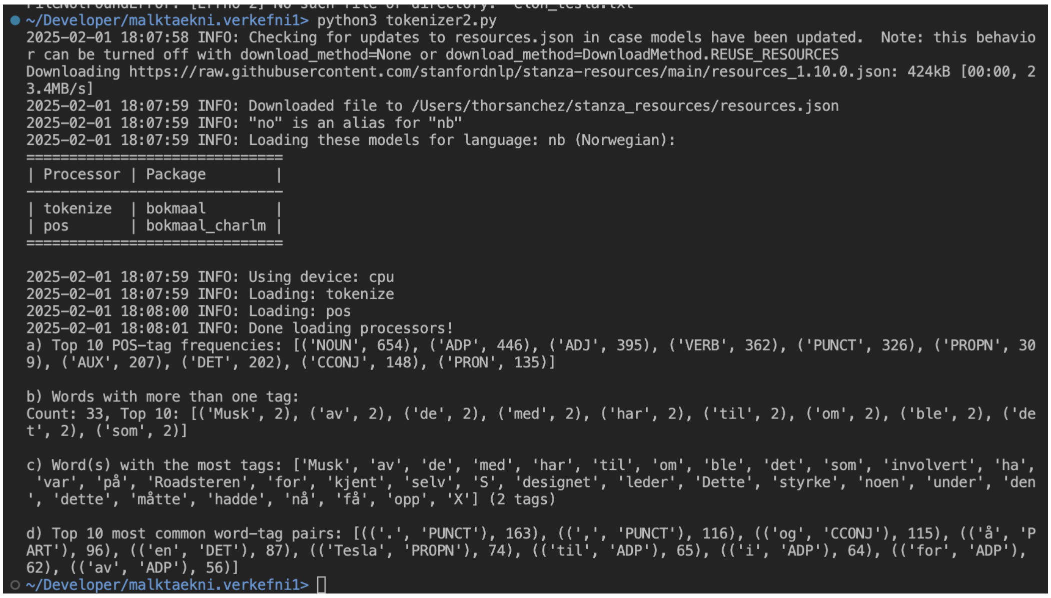
```
~/Developer/malktaekni.verkefni1> python3 tokenizer.py
Total tokens: 20807
Unique tokens: 3253
Top 10 most common tokens (case-sensitive): [('og', 747), ('Tesla', 612), ('å', 534), ('i', 432), ('som', 417), ('for'
, 391), ('har', 382), ('en', 379), ('av', 311), ('til', 305)]
Top 10 most common tokens (case-insensitive): [('og', 747), ('tesla', 612), ('å', 536), ('i', 475), ('som', 417), ('fo
r', 406), ('en', 398), ('har', 383), ('til', 315), ('av', 311)]
Tokens with more than 10 letters: [('selvkjørende', 36), ('utfordringer', 30), ('Gigafactory', 18), ('regulatoriske',
17), ('utviklingen', 16), ('utfordringene', 16), ('konkurranse', 16), ('bilindustrien', 15), ('produksjonen', 15), ('b
atteriteknologi', 15)]
Longest token: programvareoppdateringer
~/Developer/malktaekni.verkefni1> █
```

For this task, I wrote a Python program to create a tokenizer that splits the text into individual tokens using regular expressions. The text was loaded from a file named elon_tesla.txt, which contains the Norwegian corpus I collected earlier.

Results from the output:

- The total number of tokens in the corpus is 20,807, and there are 3,253 unique tokens.
- I identified the top 10 most common tokens in the text, both in a case sensitive and case insensitive way. This shows the frequency of words like “og” (and) and “Tesla,” which are very common in the text.
- Using a function, I found tokens with more than 10 letters and listed the 10 most frequent ones. Words like “selvkjørende” (self-driving) and “Gigafactory” appear prominently.
- Finally, I determined that the longest token in the corpus is “programvareoppdateringer” (software updates).

4. POS-taggers (30% for BS students – 20% for MS students)



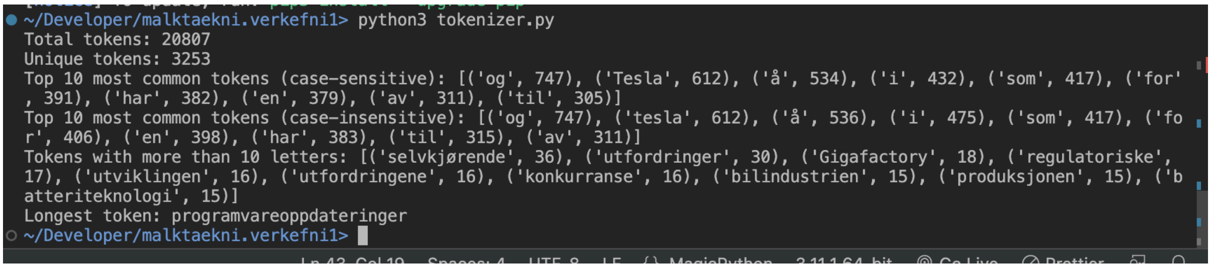
```
~/Developer/malktaekni.verkefni1> python3 tokenizer2.py
2025-02-01 18:07:58 INFO: Checking for updates to resources.json in case models have been updated.  Note: this behavio
r can be turned off with download_method=None or download_method=DownloadMethod.REUSE_RESOURCES
Downloading https://raw.githubusercontent.com/stanfordnlp/stanza-resources/main/resources_1.10.0.json: 424kB [00:00, 2
3.4MB/s]
2025-02-01 18:07:59 INFO: Downloaded file to /Users/thorsanchez/stanza_resources/resources.json
2025-02-01 18:07:59 INFO: "no" is an alias for "nb"
2025-02-01 18:07:59 INFO: Loading these models for language: nb (Norwegian):
==============================
| Processor | Package        |
------------------------------
| tokenize  | bokmaal        |
| pos       | bokmaal_charlm |
==============================

2025-02-01 18:07:59 INFO: Using device: cpu
2025-02-01 18:07:59 INFO: Loading: tokenize
2025-02-01 18:08:00 INFO: Loading: pos
2025-02-01 18:08:01 INFO: Done loading processors!
a) Top 10 POS-tag frequencies: [('NOUN', 654), ('ADP', 446), ('ADJ', 395), ('VERB', 362), ('PUNCT', 326), ('PROPN', 30
9), ('AUX', 207), ('DET', 202), ('CCONJ', 148), ('PRON', 135)]

b) Words with more than one tag:
Count: 33, Top 10: [('Musk', 2), ('av', 2), ('de', 2), ('med', 2), ('har', 2), ('til', 2), ('om', 2), ('ble', 2), ('de
t', 2), ('som', 2)]

c) Word(s) with the most tags: ['Musk', 'av', 'de', 'med', 'har', 'til', 'om', 'ble', 'det', 'som', 'involvert', 'ha',
 'var', 'på', 'Roadsteren', 'for', 'kjent', 'selv', 'S', 'designet', 'leder', 'Dette', 'styrke', 'noen', 'under', 'den
', 'dette', 'måtte', 'hadde', 'nå', 'få', 'opp', 'X'] (2 tags)

d) Top 10 most common word-tag pairs: [(('.', 'PUNCT'), 163), ((',', 'PUNCT'), 116), (('og', 'CCONJ'), 115), (('å', 'P
ART'), 96), (('en', 'DET'), 87), (('Tesla', 'PROPN'), 74), (('til', 'ADP'), 65), (('i', 'ADP'), 64), (('for', 'ADP'),
62), (('av', 'ADP'), 56)]
~/Developer/malktaekni.verkefni1> ▯
```

In this task, I used the Stanza library to perform Part-of-Speech (POS) tagging on my Norwegian text corpus. The pipeline was set up for Norwegian, and I tagged the first

20,000 words of the corpus. The tagged tokens, which pair each word with its corresponding POS tag, were saved for reuse.

- I calculated the frequency of all unique POS tags in the corpus. The top 10 most frequent tags include nouns (654 occurrences), adjective, and verb, showing the distribution of word types in the text.
- I identified words that have more than one tag. There are 33 such words, and the top examples include “Musk” and “av.”
- I found the word with the most tags. “Musk” and several other words each had two different tags.
- Finally, I retrieved the top 10 most common word tag pairs, such as punctuation marks and their respective tags, and the proper noun “Tesla.”

5. Comparing tokenizers (optional for BS students – 15% for MS



```
~/Developer/malktaekni.verkefni1> python3 tokenizer.py
Total tokens: 20807
Unique tokens: 3253
Top 10 most common tokens (case-sensitive): [('og', 747), ('Tesla', 612), ('å', 534), ('i', 432), ('som', 417), ('for'
, 391), ('har', 382), ('en', 379), ('av', 311), ('til', 305)]
Top 10 most common tokens (case-insensitive): [('og', 747), ('tesla', 612), ('å', 536), ('i', 475), ('som', 417), ('fo
r', 406), ('en', 398), ('har', 383), ('til', 315), ('av', 311)]
Tokens with more than 10 letters: [('selvkjørende', 36), ('utfordringer', 30), ('Gigafactory', 18), ('regulatoriske',
17), ('utviklingen', 16), ('utfordringene', 16), ('konkurranse', 16), ('bilindustrien', 15), ('produksjonen', 15), ('b
atteriteknologi', 15)]
Longest token: programvareoppdateringer
~/Developer/malktaekni.verkefni1>
```