1. *Text Representation.* What is the document-term matrix?
   - ☐ Each term of a language is getting documented by specially trained linguists
     { just does not make sense }
   - ☐ The latent representation of attention heads in Transformer blocks
     { maps a query and a set of key-value pairs to an output }
   - ☑ Feature space spawned by terms over documents represented as a matrix
   - ☐ Adjacency matrix of the bipartite graph of documents and the terms contained in each document
     { why bipartite? Adjacency matrix would be square, document-term matrix #documents $\times$ #terms }

2. *Assumptions.* Which are reasonable approaches to capture information in text?
   - ☑ Words are influenced by their surrounding words in context
   - ☐ The sentiment influences the word order in conjunctions
     { we would expect no influence here, needs to be tested }
   - ☑ Average sentence length depends on the author's specific writing style
     { also does depend on other factors, e.g. what type of text is the author writing}
   - ☐ Lexems depends on the anaphoric resolution
     { anaphoric resolution ... the problem of resolving what a pronoun, or a noun phrase refers to
     lexeme ... root form of a word
     pronouns are probably invariable in terms of lexemes? }

3. *Learning.* Which of these statements is true?
   - ☐ Deep learning is always preferred over rule-based approaches
     { rule-based might be simpler, if it also solves the problem why not use it, also rule-based approaches need far less data than deep learning}
   - ☐ Rule-based approaches require large amounts of labelled training data
     { rules are often created manually and then just applied, no need for large amount of data}
   - ☐ The industry often prefers rule-based approaches over machine learning
     { can't really speak for the whole industry, but I guess that both approaches have valid applications
     some companies might also like the buzzwords for machine learning}
   - ☑ Deep learning is considered the state of the art for many NLP tasks
     { deep learning produces the best results for almost all tasks}

4. *Named Entity Recognition.* There are multiple ways on how to encode the tokens used to train sequence classification systems, like named entity recognition systems.

   (a) Why is a simple binary scheme like EO (entity token, outside token) not a good idea?

   (b) What could be the reason that a BERT-based sequence classifier fail to learn `[O B I O ···]` and outputs `[O I I O ···]` instead?

   (a) By only using entity and outside tokens we can't represent two entities next to each other. We have no tag to create a boundary between these two entities.

   (b) BERT does not learn the beginning token since it just learns what is a named entity and what is not a named entity. BERT might also have trouble with neighboring entities, does not really know when one entity ends and another begins. For example two first names are probably two different people or they could be the name of one person.

5. *Word Embeddings.* Traditional word embedding techniques like word2vec have been popular in past years, but today are less used and contextual embedding techniques are preferred.

   (a) What are the advantages of contextual word embeddings, like BERT, over classical word embeddings?

   (b) Are there cases, where a traditional word embedding method is preferred? If yes, please provide an example.

   (a) For classical word embeddings we would use the same embedding for words even if they are used in different contexts. For example "computer mouse" and "mouse rodent", in this case we would get the same word embedding vector for both ('global' word embedding). Contextual word embedding we would in the best case get two different embeddings depending on the context the word is used in.

   (b) Can't really think of any. In some cases the contextual word embedding would not provide any advantages, but I see no case were it would actively hurt to use contextual word embeddings. One disadvantage of the contextual word embedding, like BERT, is that we need to know the model in order to use the embeddings, since the model generates the embedding based on the context. For traditional word embedding techniques we get a context independent vector that we can directly use.

6. *Deep Learning.* Many attention-based approaches combine as inputs a token embedding and a position embedding.

   (a) Why is there a position embedding, what is its purpose?

   (b) Are there tasks, where the position embedding is not useful? If yes, please provide an example.

   (a) The many attention-based approach does not really process the sequence of input tokens one by one (more of a parallel computation, sequence at once). Therefore the order of tokens gets lost. We believe that language has an order therefore we would like to keep this information intact. Otherwise the sentences "cook kneads dough" and "dough kneads cook" would be the same. This is achieved by additionally adding a position embedding, which could be either learned or fixed. Now the transformer is able to know the relative position of the words.

   (b) Text generation
   If we would like to generate text the transformer does not really net a position embedding for the input. Also if we only provide a single word as input the position does not matter.
   Speech to text
   Input is a recording or live feed of spoken text. The postition is embedded in the temportal domain.
   Spam detection
   The model probably tries to identify key words that indicate a spam message and does not really care about the order of the text.

7. *Word Senses.* You are asked to develop a method for disambiguation of named entities, e.g., names of celebrities, based on Tweets. For example this tweet refers to Adam Scott, the golfer and not Adam Scott, the actor.

   AmerExperience.com/Golf @ShopAmerGolf May 23

   FREE GOLF MAGAZINE Golf: Will Zalatoris, Adam Scott, Keegan Bradley among those now exempt into 2022 U.S. Open — Flipboard <url> Read for free #golfnews #golflessons #golftraining

   (a) Are there any Twitter-specific features, which can be used for this task?

   (b) Would there be other data sources that can be useful for the task?

   (c) Briefly describe how you would solve the task.

   (a) hashtags, twitter id / @handle

   (b) We could crawl or find a dataset that has the name of celebrities and some information about their field of work (sport, movie, politician, …) e.g. actor/actress imdb and use that to identify the topics of the tweet and to differentiate between two people that have the same name. Also some kind of graph that connects celebrities with each other would be helpful, for example cluster of sport teams, cluster of actors/actresses that play together in a movie, singers that colaborate, … .

   (c) We could crawl a dataset from twitter by using the twitter handle of celebrities. Now we know that these tweets correspond to the individual celebrities. We could train a model by masking out the text of the twitter handle of the person and asking the model to predict which celebrity the tweet corresponds too. We know the ground truth and after training we could apply the model on tweets that to not contain the twitter handle.

8. *Plagiarism.* Consider the university asks you to develop a system to test thesis (e.g., Bachelor and Master) for cases of plagiarism. Your system should for each thesis check, if there are plagiarised passages and mark the beginning and the end of a suspicious passage, which is then checked by human experts.

   (a) Which data sources would you consider?

   (b) What features would you use? (short list with explanation)

   (c) What method would you choose?

   (d) How well do you expect your method to work? What are the bottlenecks?

   (a) Consider larger knowledge bases in the web (wikipedia, ...), other bachelor thesis, master thesis and publications. If we know the field we could maybe narrow down the data needed. Also the submission of other students from the same course or previous years.
   Previously known cases of plagiarism (probably hand labeled) will also be useful.

   (b) A simple feature to use would be n-grams and then there compare the overlap. Maybe use POS tagging for semantic similarity. Use some vector embedding like GloVe. Compute similarity features and containment. Find subsequences that are the same (long = probably plagiarism). We should also decide if we do this on a word or sentence level. Also stylometry could be considered. If some part of the text has a vastly different style and is not cited id could be an indication for plagiarism.

   (c) Some deep learning model. There are Deep Structured Sematic Models which might be a good fit for this task. Otherwise maybe a Bi-LSTM or Transformer. Ensemble the features described. If we have a large amount of labeled plagiarism file we can apply a supervised method. Otherwise we might need to apply a unsupervised method or create a labeled dataset ourselves. One idea on how to do that would be to use text and insert a random sentence from another text into it.

   (d) Data collection would need large amount of resources. Furthermore training the model on a large amount of data will take quite some time. Would definitely work well on the form of “copy-paste plagiarism”. If somebody translates one language into another and uses this to plagiarize we would not detect that. It would probably be unfeasible to do this for all languages. Since “stealing” ideas is also a form of plagiarism, we could also not detect that. I am not sure how that could even be possible to detect.

9. *Causality.* You are asked to build a system to extract causal statements from text of a manufacturing company with a lot of textual document, including technical reports. For example the sentence
“`Mechanical stress is one of the main causes of yield loss`”
should be automatically annotated as
“$\{\texttt{Mechanical stress}\}_{Cause}$ `is one of the main` $\{\texttt{causes}\}_{Cue}$ `of` $\{\texttt{yield loss}\}_{Effect}$”.

    (a) What type of approach do you choose?

    (b) What properties do you expect for your approach? E.g., better recall/precision, better performance on longer sentence?

    (a) We want to detect the span and roles of causality in the text. A machine learning approach seems to be the best fit. Probably some kind of deep learning like BERT (maybe we find a pretrained BERT-model that is trained on technical reports or similar) since those currently produce the best results.

    (b) Evaluation often struggles as there is often no ground truth. But we would expect similar precision and recall values, maybe a bit higher recall than presicion. The longer the sentence the harder it gets to find causal relations, since there could also be multiple (and convoluted) causal relations in a single long sentence.

10. *Evaluation.* You developed a method for style transfer for German text. Given a sentence written by an arbitrary writing style, you methods outputs the "same" text as it were written by a famous author (e.g., Thomas Mann, Wolf Haas). Now you are requested to assess how well your system is working.

    (a) What evaluation methodology do you follow?

    (b) What evaluation measures do you use?

    (c) Are there known limitations in your evaluation methodology or evaluation measures?

    (a) Somehow create some ground truth that we can compare to, otherwise we need to involve humans that rate the output of the model. Might also make sense to use a explainable AI model, if possible. We also need a baseline to compare against (heuristic, human).

    (b) Compare the stylometry of the output of the model with texts from the author we are trying to mimic. Ensemble some typical stylometric features like average word length, vocabulary richness measure, white space ratio, ... Maybe the individual authors have some consistent style that is very recognizable, look at what feature separates that author from the others. Another idea would be to use an already trained model that identifies authors and let that predict the author of the created texts by writing styles.

    (c) If we use humans to create a ground truth or to rate the output we also need to look at the inter-rater agreement. They might disagree in some aspects.