

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
{ I would expect no influence here }
 - ☒ Average sentence length depends on the author's specific writing style
{ also does depend on other factors, e.g. what text are we writing }
 - ☐ Lexemes 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
{ often created manually, 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 }
 - ☒ Deep learning is considered the state of the art for many NLP 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) can't really tell where one sequence ends and the next one starts
- (b) (BERT sequence should start with a [CLS] token)
inside, outside, beginning
there is a beginning inside of two outsides

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) Rewrite this

For instance, if we have a sentence like “Your mouse likes cheese” and “Buy a new mouse for your computer” if we use ordinary embedding for the “mouse” we will get the same word embedding vector for both. In contextual word embedding, the goal is to have a different word embedding vector for each word in a specific situation.

Traditional word embedding techniques learn a global word embedding. They first build a global vocabulary using unique words in the documents by ignoring the meaning of words in different context. Then, similar representations are learnt for the words appeared more frequently close each other in the documents. The problem is that in such word representations the words’ contextual meaning is ignored.

- (b) Can’t really think of any

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) Rewrite this

Unlike RNNs that recurrently process tokens of a sequence one by one, self-attention ditches sequential operations in favor of parallel computation. To use the sequence order information, we can inject absolute or relative positional information by adding positional encoding to the input representations. Positional encodings can be either learned or fixed.

On the contrary, the transformer's encoder-decoder architecture uses attention mechanisms without recurrence and convolution. That's what the paper title *Attention Is All You Need* means. However, without positional information, an attention-only model might believe the following two sentences have the same semantics: Tom bit a dog.; A dog bit Tom.

Well if you don't have a positional embedding matrix then the Transformer has no way of knowing the relative position of each word. It would be exactly like randomly shuffling the input sentence. So the positional embeddings let the model learn the actual sequential ordering of the input sentence (which something like an LSTM gets for free).

- (b) speech to text, text generation

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 #golfflessons #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) look at the hashtags, twitter id / @handle
 - (b) 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
 - (c) Look at the other people mentioned in the tweet and find out if they have a similar profession. Use the hashtags. Crawl a dataset of tweets that mention the celebrities by handle. Get a list of celebrities to twitter handles. Now we can train a model to identify which celebrity was mentioned in a tweet (mask out the handle of this person).

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 other publications. If we know the field we can narrow down the data needed. Previously known cases of plagiarism will also be useful.
- (b) Use some vector embedding like GloVe. Compute similarity features and containment. Find subsequences that are the same (long = probably plagiarism)
- (c) some deep learning model
- (d) Data collection would need large amount of resources. Further training the model on a large amount we would take quite some time.

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 “{Mechanical stress}_{Cause} is one of the main {causes}_{Cue} of {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) machine learning approach, deep learning BERT (maybe we find a pretrained BERT-model that is trained on technical reports or similar),
 - (b) Evaluation often struggles as there is often no ground truth.

10. *Evaluation.* You developed a method for style transfer for German text. Given a sentence written by an arbitrary writing style, your method 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
- (b) compare the stylography of the output of the model with texts from the author we are trying to mimic
- (c)