Unit 2

Lecture 2.1

1. What component would you *not* typically need in a pipeline for interlingual machine translation?

   - part-of-speech tagger

     Incorrect. A part-of-speech tagger helps identify grammatical categories of words, which can be useful in an interlingual MT pipeline.

   - syntactic dependency parser

     Incorrect. A syntactic dependency parser provides structural relationships between words, which is often essential for constructing an interlingual representation.

   - sentiment classifier

     Correct. A sentiment classifier is not directly necessary for translation, as it is mainly used to analyse emotions in text rather than its syntactic or semantic structure.

2. In the Noisy Channel Model, which of the following quantities do we want to maximize when translating from Arabic (A) to Swedish (S)?

   - $P(A|S)P(S)$

     Correct. This is the decomposition of $P(S|A)$ using Bayes' Rule.

   - $P(S|A)$

     Incorrect. This expresses the probability of a target-language sentence given a source-language sentence, but the Noisy Channel Model rewrites this using Bayes' rule.

   - $P(A|S)$

     Incorrect. This represents the likelihood of the source sentence given the target sentence, but it does not consider how likely the target sentence is in the target language.

3. Which of the following statements is true about word alignments when viewed as a mathematical relation R between the set of positions on the source side and the set of positions on the target side?

   - R is a function.

     Incorrect. A function requires that each input (source position) maps to exactly one output (target position), but in word alignment, a single source word can align to multiple target words.

   - The inverse of R is a function.

     Incorrect. The inverse of R would mean each target position maps to exactly one source position, but this is also not true in word alignment.

   - Neither of these statements is true.

     Correct. Word alignment is a relation, not necessarily a function, because it allows one-to-many and many-to-one mappings.

4. Which of the following is an advantage of neural machine translation (NMT) over statistical machine translation (SMT)?

   - NMT systems do not need complex feature engineering.

     Correct. Unlike SMT, which relies on manually designed features, NMT automatically learns representations from data.

   - NMT systems can be trained without parallel text.

     Incorrect. Like SMT, NMT requires parallel text for supervised training, although some unsupervised approaches exist.

   - NMT systems are more interpretable than SMT systems.

     Incorrect. NMT models are often criticised for their lack of interpretability compared to SMT models, which have explicit probabilistic structures.

5. Why does the BLEU evaluation measure include a brevity penalty?

   - It is easy to achieve high precision with short translations.

     Correct. Short translations can match many n-grams from the reference translation while omitting important content, inflating the BLEU score without truly reflecting translation quality.

   - It is easy to achieve high recall with short translations.

     Incorrect. Recall measures how much of the reference translation is captured, and short translations tend to have low recall.

- Short translations should be penalised because they are typically not very informative.

  Incorrect. While short translations may be less informative, the BLEU brevity penalty is specifically designed to counteract artificially high precision scores.

Lecture 2.2

1. Which of the following tasks do *not* usually lend themselves to the use of autoregressive language models?

   - machine translation

     Incorrect. Machine translation is commonly handled by autoregressive models, where each token is generated based on previously generated tokens.

   - text summarisation

     Incorrect. Autoregressive models are widely used for text summarisation, where they generate a summary token by token.

   - document classification

     Correct. Document classification is not typically performed using autoregressive models, as it does not require sequential token generation.

2. Suppose we translate from Arabic (A) to Swedish (S). Which of the following quantities does a neural sequence-to-sequence model learn?

   - $P(S|A)$

     Correct. Neural sequence-to-sequence models learn the conditional probability $P(S|A)$, which models the probability of a target sequence given a source sequence.

   - $P(A|S)$

     Incorrect. This represents the probability of the source sentence given the target sentence, which is not what sequence-to-sequence models learn.

   - $P(A, S)$

     Incorrect. This represents the joint probability of both sentences occurring together, which is not explicitly modelled in neural translation systems.

3. Which of the following resources do we need in order to train sequence-to-sequence translation models?

    - parallel texts for the source language–target language pair

      Correct. Training a sequence-to-sequence model requires parallel text corpora to learn correspondences between the source and target language.

    - word alignments between the words in the source language and target language

      Incorrect. Word alignments are useful for phrase-based statistical translation models but are not directly required for training modern neural translation models.

    - a language model for the target language

      Incorrect. While a separate target-language model can help in some hybrid approaches, sequence-to-sequence models inherently learn a target language model during training.

4. Which of the following parameters does *not* impact the space complexity of beam search?

    - number of possible translations for the source sentence

      Correct. Beam search only keeps track of a fixed number of hypotheses, independent of the total number of possible translations.

    - width of the beam

      Incorrect. A wider beam requires storing more hypotheses, increasing space complexity.

    - lengths of the generated target sentences

      Incorrect. Longer sentences require more memory storage, impacting space complexity.

5. Why do we use length normalisation together with beam search in decoding?

    - We do not want to penalise long translations.

      Correct. Without length normalisation, longer translations can be disproportionately penalised because the probability of a sequence decreases as more tokens are added. Length normalisation mitigates this bias.

    - We do not want to penalise short translations.

      Incorrect. Standard beam search often *favours* shorter translations because the probability of a sequence decreases as more tokens are added.

- We want to avoid numerical overflow.

  Incorrect. Numerical overflow is not a major concern in beam search; the primary issue is the tendency of shorter sequences to have higher probability scores.

Lecture 2.3

1. Which of the following NLP tasks is most likely to be handled comparatively well using static word vectors rather than contextual embeddings?

   - topic classification

     Correct. Topic classification can often be effectively handled using static word vectors, as the overall topic of a document may not require deep contextual understanding of individual words.

   - coreference resolution

     Incorrect. Coreference resolution often requires understanding the context in which words are used, making contextual embeddings more suitable.

   - word sense disambiguation

     Incorrect. Word sense disambiguation relies heavily on context to determine the correct meaning of a word, making contextual embeddings more appropriate.

2. Consider the sentence "Dogs may bark at strangers" and assume that words are indexed from 1 to 5. Which attention weight do you expect to be highest?

   - α_{31}

     Correct. When refining the representation for the word "bark" (position 3), the word "dogs" (position 1) is highly relevant to distinguish the word sense "bark" (as in the sound a dog makes) from other possible meanings (e.g., the outer layer of a tree). Therefore, we expect a high attention weight α_{31}.

   - α_{33}

     Incorrect. The attention weight α_{33} corresponds to the word "bark" attending to itself, which is typically the highest in self-attention mechanisms.

   - α_{35}

     Incorrect. The word "strangers" (position 5) is less relevant to the meaning of "bark" in this context, so we would not expect a high attention weight α_{35}.

3. Consider following values for the example of “Contextual word embeddings via attention”.

   $$\begin{aligned} h_1 &= [0.5539, 0.7239] \\ h_2 &= [0.4111, 0.3878] \\ h_3 &= [0.2376, 0.1264] \end{aligned}$$

   Assuming that the attention score is computed using the unscaled dot product, what is the refined representation for h_2?

   - $[0.5084, 0.3194, 0.1467]$

     Incorrect. This is the vector of attention scores, not the refined representation.

   - $[0.3962, 0.3279, 0.2759]$

     Incorrect. This is the vector of attention weights, not the refined representation.

   - $[0.4198, 0.4488]$

     Correct. In self-attention, the output vector must have the same dimensionality as each representation h_i.

4. Which of the following statements about the more general characterisation of attention in terms of queries, keys and values is true?

   - The output has the same length as each value.

     Correct. In attention mechanisms, the output is a weighted sum of values, meaning it retains the same dimensionality as the values.

   - The query has the same length as each value.

     Incorrect. The query and value dimensions can differ depending on the model architecture.

   - Each key has the same length as each value.

     Incorrect. Keys and values do not necessarily have the same dimensionality, though they often do in standard self-attention.

5. Consider an instance of multi-head attention with 8 heads where the queries and keys have size 256. What would be the typical key length in each block’s attention mechanism?

   - 256

     Incorrect. The total query/key size is 256, but it is divided across the 8 attention heads.

- 32

  Correct. The key length per head is $\frac{256}{8} = 32$ because the attention mechanism splits the dimensions evenly across multiple heads.

- 8

  Incorrect.

Lecture 2.4

1. Which of the following is the main advantage of the Transformer architecture over recurrent neural networks?

   - direct access to all elements in the input sequence

     Correct. Unlike RNNs, which process sequences sequentially, Transformers use self-attention, allowing each token to access all tokens in the input at once, leading to more efficient parallel computation.

   - significantly reduced need for training data

     Incorrect. While Transformers can leverage large datasets effectively, they typically require *more* data than RNNs to achieve good performance.

   - supports significantly more compact models

     Incorrect. Transformers often have a higher number of parameters than RNNs due to their attention mechanisms and feedforward layers.

2. Consider the example translation used to illustrate the Transformer architecture. Which of the following statements is *false*?

   - The final encoder representation of *drink* depends on the token embedding of *Kaffee*.

     Correct. The Transformer encodes the English sentence and uses the representations computed in this process to generate the German sentence – not the other way around.

   - The final encoder representation of *coffee* depends on the token embedding of *drink*.

     Incorrect. Self-attention ensures that each token's representation is influenced by other tokens in the sequence.

   - The final decoder representation of *Kaffee* depends on the final encoder representation of *coffee*.

Incorrect. In an encoder–decoder architecture, all representations computed in the encoder can influence the representations computed in the encoder through cross-attention.

3. The Transformer architecture uses three different variants of multi-head attention. Which one is used in the encoder?

   - self-attention

     Correct. The Transformer encoder uses self-attention to allow each token to attend to all other tokens in the input sequence.

   - masked self-attention

     Incorrect. Masked self-attention is used in the decoder to prevent attending to future tokens during training.

   - cross-attention

     Incorrect. Cross-attention is used in the decoder to attend to the encoder's output.

4. What is the purpose of layer normalisation?

   - centering and scaling the layer's input values

     Correct. Layer normalisation standardises the inputs to a layer by centering and scaling them to stabilizs.

   - squeezing the layer's input values into the interval $[0, 1]$

     Incorrect. Layer normalisation does not necessarily map values into the $[0, 1]$ range; it normalises them based on mean and variance.

   - down-scaling the layer's output values

     Incorrect. The normalisation occurs on the inputs, not the outputs, and it involves both centering and scaling rather than just down-scaling.

5. True or false: Permuting the input tokens to a Transformer encoder does not change the final token representations.

   - True

     Incorrect. The Transformer architecture relies on positional encodings to capture word order, so permuting tokens changes their representations.

   - False

     Correct. Transformers process input tokens in parallel, but they use positional encodings to maintain word order, meaning changing the order alters final representations.

- Depends on the input tokens

  Incorrect. While some word orders may yield similar representations, in general, changing the order affects the token representations due to positional encodings.

Lecture 2.5

1. What does the term generative pre-training refer to?

   - pre-training on a language modelling task

     Correct. Generative pre-training refers to training a model on a language modelling task where it learns to predict the next token in a sequence before fine-tuning it on specific downstream tasks.

   - pre-training with a generative probabilistic model

     Incorrect. While generative pre-training involves a generative model, it specifically refers to pre-training using autoregressive language modeling, not just any generative probabilistic approach.

   - pre-training on automatically generated text

     Incorrect. Pre-training is typically done on large corpora of natural text rather than automatically generated text.

2. Looking at the original GPT model architecture (Radford et al., 2018), what is the approximate number of trainable parameters in the FNN?

   - 4,718,592

     Correct. The feedforward neural network (FNN) layer in GPT-1 contains approximately 4,718,592 trainable parameters based on its architecture.

   - 589,824

     Incorrect.

   - 9,216

     Incorrect.

3. Suppose you want to fine-tune a GPT model on the Stanford Natural Language Inference dataset. What is the minimal number of parameters you need to update?

   - the number of parameters in the pre-trained GPT model

     Incorrect. Full model fine-tuning updates all parameters, but minimal tuning requires updating only part of the model.

- the number of parameters in the final Linear layer

  Correct. The minimal number of parameters that need to be updated corresponds to the final Linear layer, which maps the model's hidden representations to the output task.

- the sum of these two

  Incorrect. While full fine-tuning updates both, the minimal required update involves only the final Linear layer.

4. What do we mean when we say that GPT-3 exhibits zero-shot behaviour?

   - It can solve tasks without any task-specific fine-tuning.

     Correct. Zero-shot learning means that GPT-3 can perform tasks it was not explicitly trained on, relying only on its general language understanding.

   - It can solve tasks without any training.

     Incorrect. GPT-3 is extensively trained on a large corpus of text before demonstrating zero-shot capabilities.

   - It can solve tasks without receiving any input.

     Incorrect. GPT-3 requires input prompts to generate responses, even in a zero-shot setting.

5. What is a reasonable explanation for the observation that GPT-3 can translate from English to French?

   - The data sets used for pre-training contain example translations.

     Correct. GPT-3 is trained on large-scale internet text, which includes many instances of translations, allowing it to learn translation patterns.

   - The number of model parameters approaches the number of neurons in the human brain.

     Incorrect. While GPT-3 has more parameters than there are neurons in the human brain (175B vs. 86B), the model's translation ability is primarily due to the data it was trained on, not its parameter count.

   - The model has been trained based on feedback from professional translators.

     Incorrect. GPT-3 is trained using unsupervised learning on large text corpora rather than direct feedback from translators.

Lecture 2.6

1. What is the purpose of the segment encoding in BERT?

   - to distinguish between different segments of sentence pairs

     Correct. In BERT, segment embeddings are used to differentiate between two input sentences in tasks like next sentence prediction.

   - to distinguish between different segments of words

     Incorrect. Words in BERT are represented using WordPiece embeddings, not segment encodings.

   - to distinguish between different segments of the word embeddings

     Incorrect. Segment encodings operate at the sentence level, not at the embedding level.

2. BERT is pre-trained on the masked language modelling task. Why is this task not suitable for pre-training GPT models?

   - GPT is based on the Transformer decoder, and as such can only "look back".

     Correct. GPT is an autoregressive model that generates text sequentially, meaning it cannot use bidirectional context like masked language modeling in BERT.

   - Masked language modelling does not scale up to the number of parameters in GPT.

     Incorrect. The scalability of masked language modelling is not the limiting factor; the autoregressive nature of GPT is.

   - GPT is trained on individual sentences, not sentence pairs (like BERT).

     Incorrect. While GPT does not rely on sentence pairs, this is not the reason masked language modeling is unsuitable; rather, GPT's causal attention prevents it from utilising masked tokens.

3. What is the main purpose of the `[CLS]` token in BERT?

   - It is used as a representation of the complete input sentence pair.

     Correct. The `[CLS]` token provides a fixed-length representation of the entire input and is commonly used for classification tasks.

   - It is used as a padding token.

     Incorrect. Padding is handled separately and does not involve the `[CLS]` token.

- It is used for the masked language modelling task.

  Incorrect. Masked language modelling applies to other tokens, but the `[CLS]` token itself is not masked.

4. In masked language modelling, we generate training examples by randomly branching into one of three cases. Which of these would typically make the token representation at the selected position more dissimilar to the representations of the surrounding tokens?

   - replace the selected token with the `[MASK]` token

     Incorrect. The `[MASK]` token still allows the model to infer meaning from surrounding tokens.

   - replace the selected token with a random word

     Correct. Replacing a token with a random word disrupts contextual consistency, making the token representation more dissimilar to its surroundings.

   - not replace the selected token

     Incorrect. Keeping the original token preserves its contextual relationship with surrounding words.

5. Which advantage does replaced token detection have over masked language modelling?

   - It learns from all input tokens.

     Correct. Unlike masked language modelling, which modifies only a subset of tokens, replaced token detection operates on all tokens, improving data efficiency.

   - It only needs two classes.

     Incorrect. The advantage of replaced token detection is not related to the number of classification categories.

   - It does not need the `[MASK]` token.

     Incorrect. While replaced token detection avoids the need for a special masking token, its primary advantage is the ability to learn from all tokens.