i

Front page

IN4080 Natural Language Processing

Fall 2022

Tuesday, December 13
09:00 AM - 13:00 PM (4 hours)

All questions should be answered!
Each question is assigned a weight which is indicated.
The maximum number of points for the whole set is 100 points.

Permitted materials: None
An on-screen calculator is available.

You may answer in English, Norwegian, Danish or Swedish.

1(a)

Naive Bayes

NB

$$\arg\max_{c \in C} P(c \mid \mathrm{f}) = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i = v_i \mid c)$$

The formula shows the model for Naive Bayes classification.

- Give a short description of the formula:
    - What is C and c?
    - What is **f**, f_i, v_i and n?
    - What is meant by *argmax*?

Fill in your answer here

Maximum marks: 5

- c indicates one class, C is the set of all classes. (1 pt)
- Question 2 (2 pts)
  - $f_i$is one feature, $v_i$is the value of this feature
  - *n* is the number of features
  - **f** is the feature vector $\boldsymbol{f} = \langle f_1 = v_1, f_2 = v_2, \dots, f_n = v_n \rangle$ which may also be written $\boldsymbol{f} = \langle v_1, v_2, \dots, v_n \rangle$ if the order of features is determined
- Argmax (2 pts.)
  - *argmax_{c in C}* means consider the expression within the scope of *argmax* for each *c* in *C* and choose the *c* that yields the largest value.

1(b) Assumptions

Which simplifying assumptions are made by the Naive Bayes model? Why can these assumptions result in less accurate classifiers compared to other learning algorithms?

Fill in your answer here

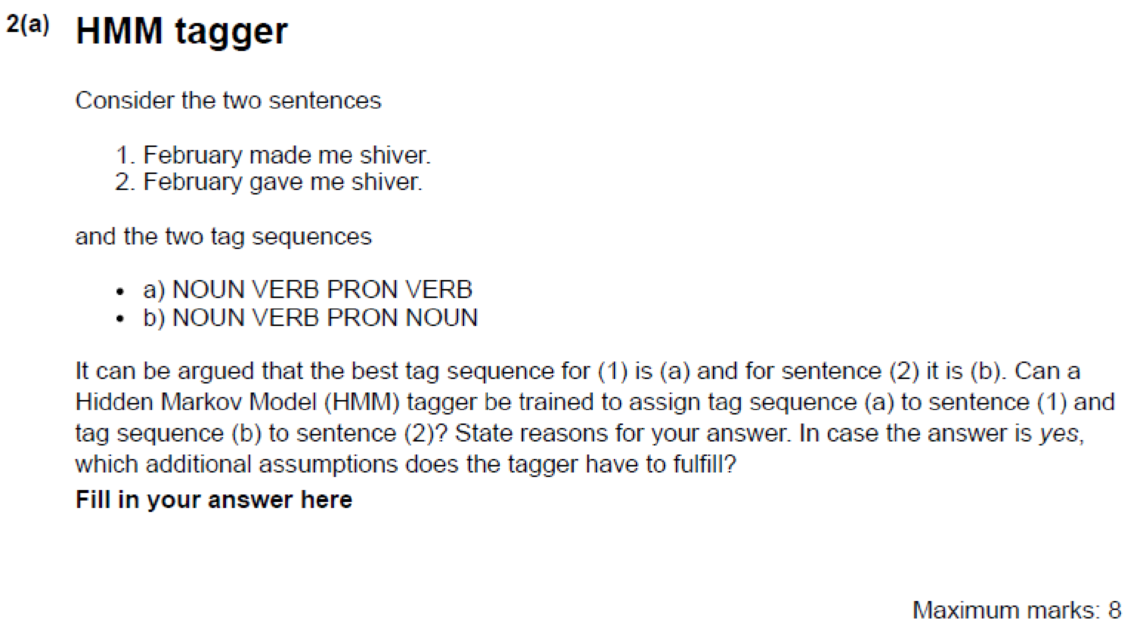
Maximum marks: 5

The simplifying assumption is that the value of each feature given a class is independent of the values of the other features, i.e., that

$$P(f_1 = v_1, f_2 = v_2, \dots f_n = v_n | c) = \prod_{k=1}^{n} P(\, f_k = v_k | c)$$

(3 pts.)

Features are in general not independent of each other. For example, if the task is bag-of-words text classification, the given name and family navn of a person, e.g. “Barack” and “Obama”, tend to co-occur. They are not indendent features. As a result, a NB model might put more weight on Obama relative to an entity which is only represented by one word, say “Senate”.
(2 pts.)



2(a) HMM tagger

Consider the two sentences

1. February made me shiver.
2. February gave me shiver.

and the two tag sequences

- a) NOUN VERB PRON VERB
- b) NOUN VERB PRON NOUN

It can be argued that the best tag sequence for (1) is (a) and for sentence (2) it is (b). Can a Hidden Markov Model (HMM) tagger be trained to assign tag sequence (a) to sentence (1) and tag sequence (b) to sentence (2)? State reasons for your answer. In case the answer is *yes*, which additional assumptions does the tagger have to fulfill?

Fill in your answer here

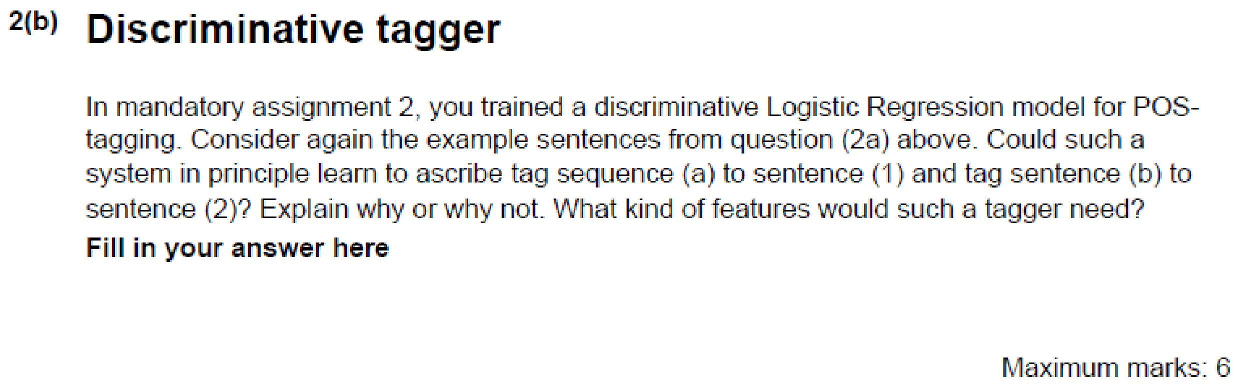
Maximum marks: 8

The HMM tagger chooses the tag sequence which yields the largest value for a product of factors of the forms P(t_n | t_{n-1}) and P(w_n | t_n). Here w_n is word n, and t_n is the corresponding tag. In the choice between (a) and (b) for sentence (1), the only difference is in the last part

a') P(t_4 | t_3)P(w_4 | t_4)P(t_5 | t_4) = P(VERB | PRON)P(VERB | shiver)P(<\s> | VERB)
b') P(t_4 | t_3)P(w_4 | t_4)P(t_5 | t_4) = P(NOUN | PRON)P(NOUN | shiver)P(<\s> | NOUN)

If the tagger chooses sequence (a) over sequence (b), then expression (a') much have a larger value than expression (b'). As the only difference between sequence (a) and (b) for sentence (2) is the difference between (a') and (b'), it will choose (a) for sentence (2) as well.

Comment:
There a several more or less formal ways to formulate the argument. The important thing is to show sufficient understanding of the HMM model to show that the choice is independent of the distinction between w2=made and w2=gave.



2(b) Discriminative tagger

In mandatory assignment 2, you trained a discriminative Logistic Regression model for POS-tagging. Consider again the example sentences from question (2a) above. Could such a system in principle learn to ascribe tag sequence (a) to sentence (1) and tag sentence (b) to sentence (2)? Explain why or why not. What kind of features would such a tagger need?

Fill in your answer here

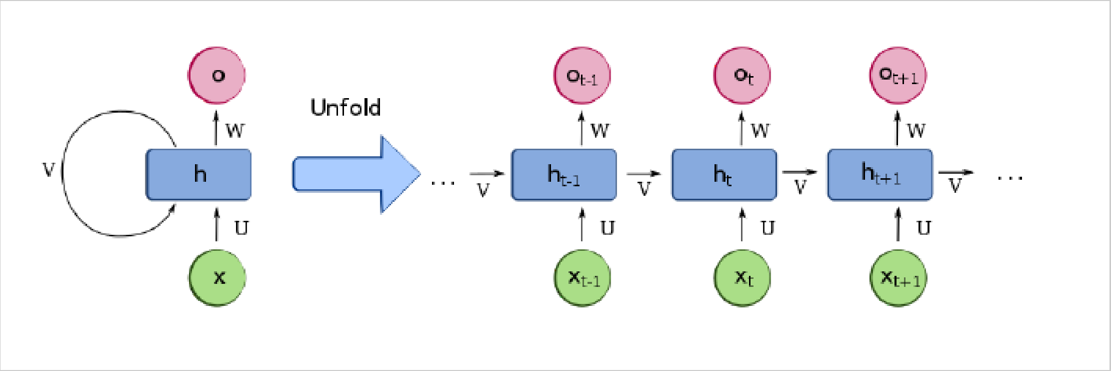
Maximum marks: 6

This tagger makes is decision on the basis of *m* many preceding and following words and *k* many preceding tags for some numbers *m* and *k*, formally

$$\underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P\left(t_i|\, t_{i-k}^{i-1} w_{i-m}^{i+m}\right)$$

By choosing *m*≥2, we could use w_{i-2} as a feature. Hence we could get a different result for *t*_4, when *w*_2= gave from when *w*_2=made.

2(c) RNN



The figure, taken from Wikipedia, is meant to illustrate a simple Recurrent Neural Network (RNN). We will use such a network for POS-tagging. Taking sentence (1) from question (2a) as example, answer the following questions:

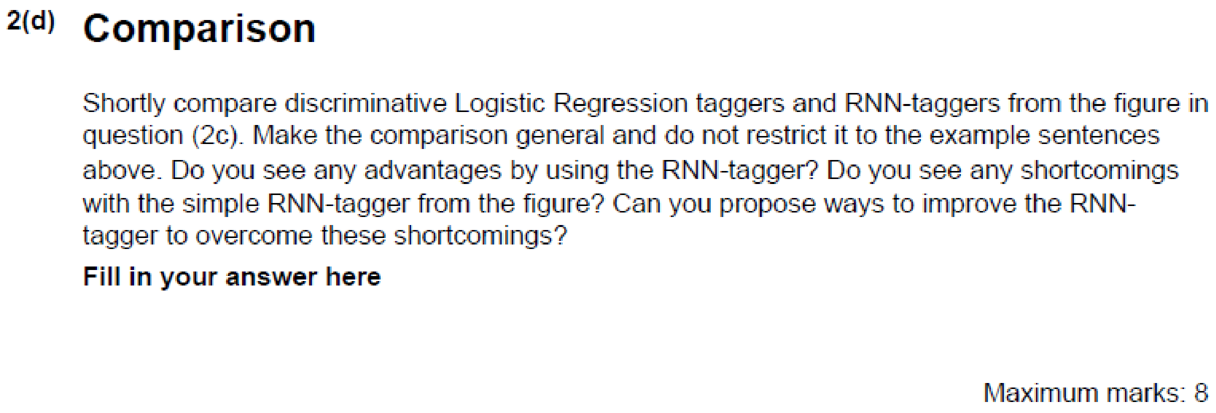
- To what do x_t and o_t correspond in the case of POS-tagging?
- What is h_t meant to represent?
- How is h_t computed?
- How is o_t computed?

Fill in your answer here

Maximum marks: 8

Exercise 2c (2 pts. for each)

- x_t would be a representation of the word w_t, most probably a word embedding and o_t the predicted tag for this word
- $\boldsymbol{h}$ is a layer of hidden states. $\boldsymbol{h}_t$ is the state of this layer after seeing word *t*. It is meant to represent the word sequence so far: w_1, w_2, …, w_t
- $\boldsymbol{h}_t$ is calculated from the preceding value $\boldsymbol{h}_{t-1}$of $\boldsymbol{h}$ together with the current word **x**_t. It can be written $\boldsymbol{h}_t = g(V\boldsymbol{h}_{t-1} + U\boldsymbol{x}_t)$. There are weighted connetions from the nodes in $\boldsymbol{h}$ to the nodes in $\boldsymbol{h}$. *V* is the weight matrix for these connections. There are also weighted connections from the word representation **w** to the hiden layer **h**, with weight matrix U. $\boldsymbol{h}_t$ is computed by taking the sum of the weighted sums $V\boldsymbol{h}_{t-1}$, and $U\boldsymbol{x}_t$, and applying an activation function *g*.
- $\boldsymbol{o}_t = f(W\boldsymbol{h}_t)$ where W are weighted connections from the of $\boldsymbol{h}$ layer to the outpu layer, while *f* is the softmax-function.



2(d) Comparison

Shortly compare discriminative Logistic Regression taggers and RNN-taggers from the figure in question (2c). Make the comparison general and do not restrict it to the example sentences above. Do you see any advantages by using the RNN-tagger? Do you see any shortcomings with the simple RNN-tagger from the figure? Can you propose ways to improve the RNN-tagger to overcome these shortcomings?

Fill in your answer here

Maximum marks: 8

Advantages with the RNN-tagger compared to the LogReg-tagger: (3 pts.)

- The RNN-tagger can – in principle – consider the whole preciding sequence of words. The LR-tagger is restricted to a window of a given size of words and tags
- One has to give the features for the LR-tagger explicitly, while one does not have to specify this for the RNN-tagger. The RNN-tagger can learn what is important or not.
- The RNN-tagger will use embeddings. It will thereby not only learn from words, but also from similar words. The LR-tagger will only have access to the words themtselves, the tags and classes of words we state explicitly (e.g. all words having the same last three 3 lettr suffix.). Many features, including the pattern of neighboring words themselves may be rare phenomena, and thus do not provide good generalizations.

Shortcomings of the RNN-tagger compared to the LR-tagger (3 pts.)

- Most importantly, the RNN-model only considers the left-cotext of the word while the LR-tagger can consider a number of following words. Considers for example the sequence of words (3) with associated tags (c)

  3) They saw her

  c) PRON VERB ?

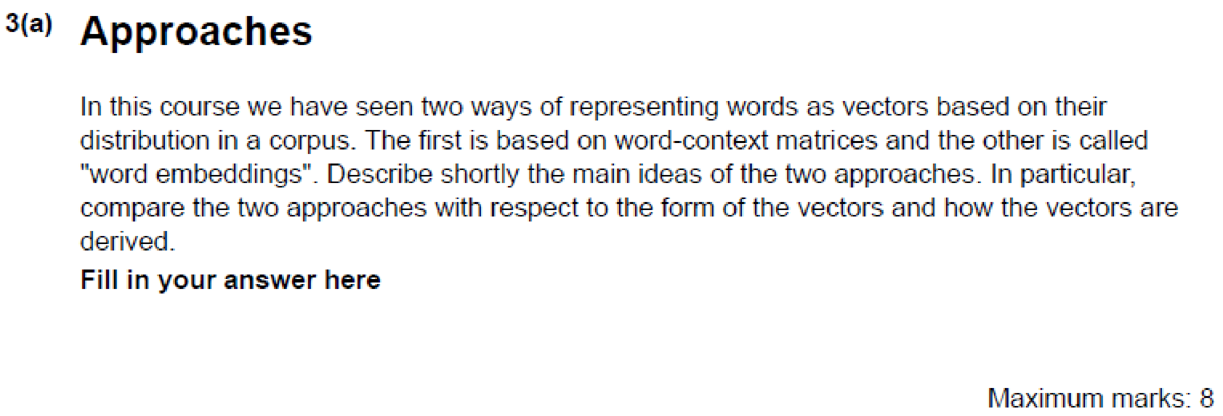
  There is a choice for *her* between PRON and DET. The RNN has to make a choice at this stage, but the best choice is different in (4) compared to (5)

  5) They saw her .

  6) They saw her car .

Improvements (2 pts.)

- One option is to use a Bi-RNN, i.e., two RNNs one running from left to right and one running from right to left, and compute the output from the corresponding hidden states in both networks.



3(a) Approaches

In this course we have seen two ways of representing words as vectors based on their distribution in a corpus. The first is based on word-context matrices and the other is called "word embeddings". Describe shortly the main ideas of the two approaches. In particular, compare the two approaches with respect to the form of the vectors and how the vectors are derived.

Fill in your answer here

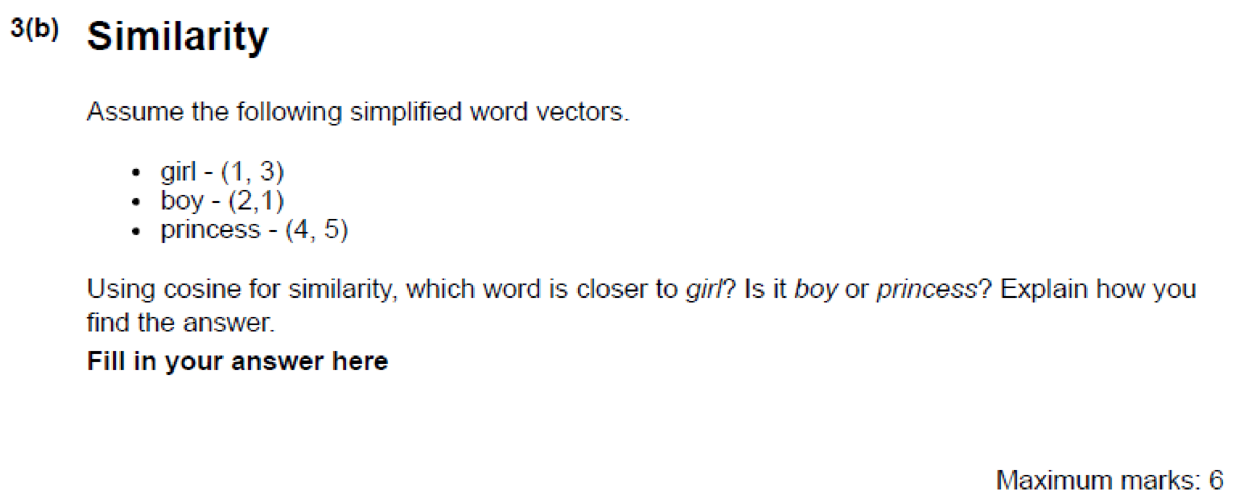
Maximum marks: 8

The word-context or term-term matrix.
The word representations are derived from a corpus.
First we decide on a set of context-words, C=(c_1, c_2, …, c_n). This can be all the word types in the corpus, or a subset where we have excluded words of low or very high frequency. Let n be the size of C. Each word w is represented by a vector of n integers (v_1, v_2, …, v_n) where v_i counts of many times c_i occurs in a context of w. Context can be defined in various ways. It could e.g., be a sentence or a window of a given size *m* around each occurrence w_i of the word w.

In the word-embeddings approach, each word w is represented by a vector of reals of a fixed length m, embedding(w) = (e_1, e_2, …, e_m). The length m is somewhere between 50 and 1000. The embeddings are learned from a prediction task, e.g., predicting the next word following w, or predicting all words in a given context of the occurrence of w.

Comparison:

- The word-context matrix representations are calculated by counting, while the embeddings are learned by a machine-learning task
- The word-context matrix representations are long and sparse, i.e., may contain many zero entries. The embeddings are dense (i.e., no zeros) and shorter
- The word-context matrix representations are based on words. They do not exploit semantic similarlities between the context words. The embeddings exploit similarities between words in the context when they are learned as the training constructs one set of embeddings for words and another set of embeddings for context words.



3(b) Similarity

Assume the following simplified word vectors.

- girl - (1, 3)
- boy - (2,1)
- princess - (4, 5)

Using cosine for similarity, which word is closer to *girl*? Is it *boy* or *princess*? Explain how you find the answer.

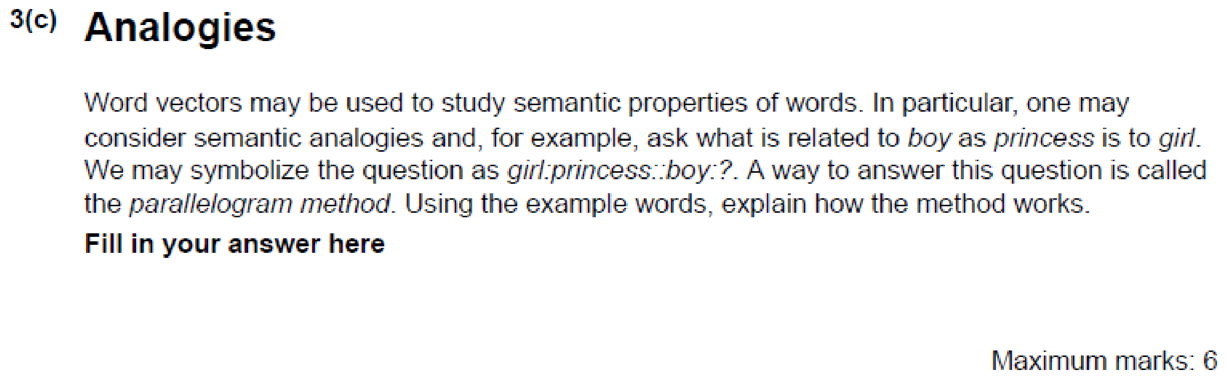
Fill in your answer here

Maximum marks: 6

Let g = (1,3) be the vector for girl
b = (2, 1) the vector for boy
p = (4, 5) the vector for princess.

Then the length of g is ||g||=sqrt(1**2 + 3**2) = sqrt(10)
||b||= sqrt(5)
||p|| = sqrt(4**2 + 5**2) = sqrt(41)

cos(g, b) = (g dot b)/(||g||*||b||) = (1*2 + 3*1)/(sqrt(10)*sqrt(5)) = 5 /sqrt(50) = 1/sqrt(2) = 0.71
cos(g, p) = (g dot p)/(||g||*||p||) = (1*4 + 3*5)/(sqrt(10)*sqrt(41)) = 19 /sqrt(410) = 0.93
princess is closer to *girl* than what *boy* is.



3(c) Analogies

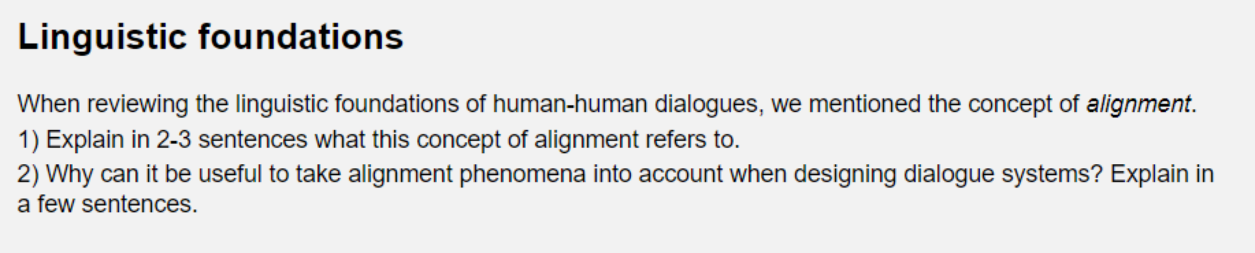
Word vectors may be used to study semantic properties of words. In particular, one may consider semantic analogies and, for example, ask what is related to *boy* as *princess* is to *girl*. We may symbolize the question as *girl:princess::boy:?*. A way to answer this question is called the *parallelogram method*. Using the example words, explain how the method works.

Fill in your answer here

Maximum marks: 6

We calculate a vector intuitively corresponding to *princess* and *boy*, but not *girl*:
b + p – g = (2,1) + (4, 5) – (1, 3) =(5, 3)
Then we compare this vector with all the word vectors and choose the one with the largest cosine value with this vector.

4(a)



Linguistic foundations

When reviewing the linguistic foundations of human-human dialogues, we mentioned the concept of *alignment*.
1) Explain in 2-3 sentences what this concept of alignment refers to.
2) Why can it be useful to take alignment phenomena into account when designing dialogue systems? Explain in a few sentences.

Max marks: 4

1) The concept of “interactive alignment” is grounded in the idea that dialogue is *a collaborative activity*. Consequently, the participants in a conversation seek to continuously align their mental representations, ensuring that they stay “on the same page” through the course of the interaction. This alignment (also called *grounding*) is achieved through various types of feedback signals and ensures that the *common ground* of the interaction is maintained and gradually expanded.

   In addition to this alignment of mental representations, dialogue participants have also been shown to unconsciously *imitate* each other, in terms of lexical choices, pronunciation, speech rate, and even gestures. If I talk about going to the park and chose the word “stroll”, my interlocutor is more likely to also adopt this very word in their reply compared to other possible choice of words, like e.g. “walk”.

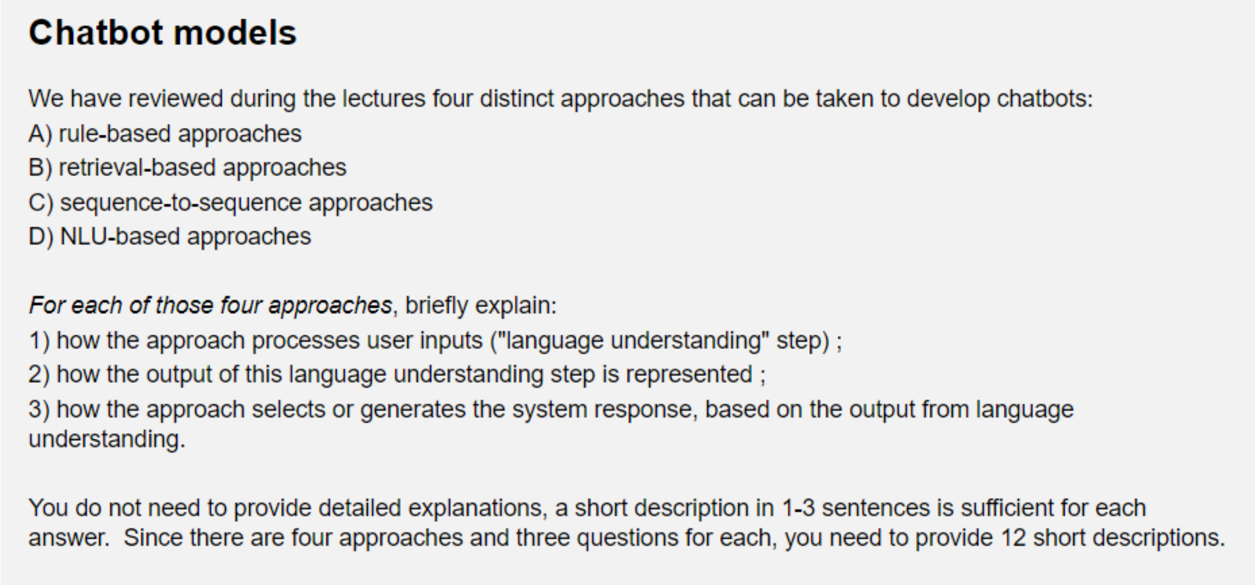
   ***Points**: 1 pt if the student correctly explains this alignment of mental representations and its relation to grounding, and 1pt if the student also mentions those imitation phenomena. Note that the students do not need to provide an answer as long as the one I have given above.*

2) It is useful to remember the occurrence of those alignment phenomena for a few reasons. First, human users are likely to utter various grounding signals (such as backchannels, clarification requests, etc.) whose goal is precisely to facilitate the alignment of mental representations with their conversational partner. They will also expect the dialogue system to produce such grounding signals to ensure it has correctly understood the user. It is therefore important that the dialogue system can both understand and produce such grounding signals as part of its conversational behaviour.

   In addition, the existence of imitation phenomena is also something to keep in mind. For instance, human users will tend to unconsciously reuse the same types of words or constructions as the one uttered by the dialogue system. This can be exploited to “nudge” the users into formulating their intents in a way that is less ambiguous or easier for the system to interpret. This is also true for speech processing: users will tend to adopt a pronunciation and speaking style that is closer to the dialogue system.

   ***Points**: 2 points if they correctly explain at least one reason.*

4(b)



Chatbot models

We have reviewed during the lectures four distinct approaches that can be taken to develop chatbots:
A) rule-based approaches
B) retrieval-based approaches
C) sequence-to-sequence approaches
D) NLU-based approaches

For each of those four approaches, briefly explain:
1) how the approach processes user inputs ("language understanding" step) ;
2) how the output of this language understanding step is represented ;
3) how the approach selects or generates the system response, based on the output from language understanding.

You do not need to provide detailed explanations, a short description in 1-3 sentences is sufficient for each answer. Since there are four approaches and three questions for each, you need to provide 12 short descriptions.

Max marks: 12

A) Rule-based approaches:

1) Rule-based approaches typically search for handcrafted patterns in the user inputs, either at a superficial level (i.e. a match for a regular expression) or at a deeper level (i.e. a particular pattern in the syntactic or semantic structure of the utterance)

2) A pattern match is typically associated with a certain category of user inputs or “intents”. The output of this pattern matching can either be the matched pattern itself or the category it seeks to represent.

3) Each pattern or category is then mapped to a particular response, which is typically handcrafted. Some responses may be templated, with slots to be filled.

B) Retrieval-based approaches:

1) Retrieval-based approaches encodes the user utterance into a (document) vector. This encoding can be done in various ways, from the simple calculation of TF-IDF values to the use of sentence transformers or similar neural models.
2) The output of the language understanding step for retrieval-based approaches is a vector, which can be sparse (for e.g. TF-IDF values) or dense (for e.g. document embeddings).

3) Based on this vector representing the user input, the response selection seeks to select from the corpus the utterance that is most appropriate as response to the user input. One simple approach is to first find the corpus utterance that is most similar to the input utterance (based the cosine similarity of their respective vectors), and then take the utterance that comes immediately after it. Alternatively, one can rely on a dual encoder model that gives a score to (input, response) pairs, and retrieve the response with the highest score.

C) Sequence to sequence approaches:

1) seq2seq approaches rely on a neural encoder model to process the user input. This encoder model is typically a large, neural language model fine-tuned to the task and dialogue domain.

2) The output is expressed as contextualized word vectors associated with the tokens of the user utterance.

3) The generation of the system response is done using a decoder model which predicts the response token by token. The decoding process attends to both the vectors of the user input as well as the tokens that have been produced so far. The decoding stops when a special end-of-utterance token is produced.

D) NLU-based approaches:

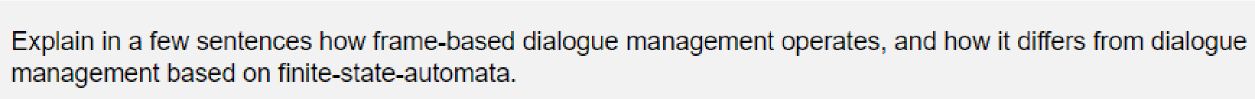
1) NLU-based approaches seek to classify the user input into a set of predefined categories, often called intents. This classification can be done using a wide range of text classification techniques, from simple BOW models to neural language models fine-tuned for text classification. In some applications, the NLU also needs to detect the occurrence of specific slot values. This slot detection is typically done using sequence models.

2) The output of the NLU is an intent, or a probability distribution over intents, possibly associated with a set of detected slots in the user inputs.

3) The response selection is typically done through handcrafted responses (or response templates) associated with each intent.

***Points**: one per answer (12 in total). The answers do not need to be identical to the ones provided here, as long as the key idea is conveyed.*

4(c)



Explain in a few sentences how frame-based dialogue management operates, and how it differs from dialogue management based on finite-state-automata.

Max marks: 4

The key idea of frame-based dialogue management is to formulate the dialogue management task in terms of domain-specific slots to fill. Those slots typically correspond to some variable or attribute that must be determined in order to complete the task. For instance, a flight booking system will need to at least know the departure date and time as well as the airports of departure and destination.

Crucially, frame-based dialogue management does not impose a strict order on the slots to fill, and allow for several slots to be filled at the same time. Instead of following a rigid script of dialogue steps, as done with fine-state-automata, frame-based dialogue management operates by

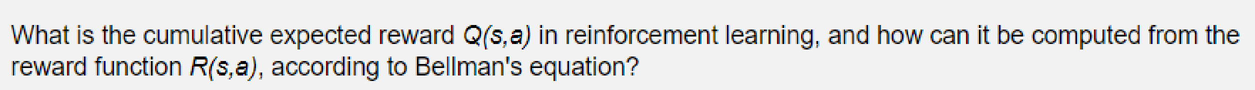
1) Updating the current dialogue state (represented by the slots already filled and the ones yet to be filled) given the user inputs
2) Selecting the next system response to determine the value of the slots that remain unfilled in the most effective manner.

The dialogue proceeds until all slots are filled, in which case the task is complete. The selection of system response can be done either manually or using a data-driven model (optimized through supervised or reinforcement learning).

Points:

- 1 point if the student mentions the concept of slot and their use
- 1 point if the student explains that frame-based dialogue management allows for a more flexible conversational behaviour than FSA, as it does not need to follow a rigid script
- 1 point if they mention that slots are updated based on user inputs
- 1 point if they mention that the selection of system responses seeks to fill the slots that are yet to be filled. Nice if the student also elaborates on how this selection can be performed (using rule-based techniques or data-driven models), but this is not required.

4(d)



What is the cumulative expected reward *Q(s,a)* in reinforcement learning, and how can it be computed from the reward function *R(s,a)*, according to Bellman's equation?

Max marks: 5

As the name says, the cumulative expected reward *Q(s,a)* represents the expected accumulation of rewards over time upon executing action *a* in state *s*. The estimation of *Q(s,a)* rests on the specification of an MDP model characterized by

- a reward function *R(s,a)* that indicates the immediate reward (which may be positive or negative) that can be obtained by executing action *a* in state *s.*
- A transition model *P(s'|s,a)* that expresses the probability of reaching state *s'* after executing action *a* in state *s*.
- A set of states *S* and actions *A* over which the reward function and transition model are defined.

The Bellman equation formulates this Q-value as:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

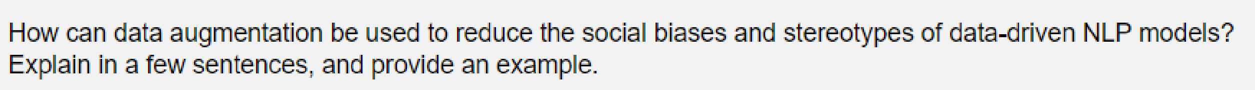
In this formula, the \lambda corresponds to a discount factor that express the relative worth of future rewards at time t+1 relative to the reward that can be obtained at time t.

As we can observe from the Bellman equation, the *Q(s,a)* value is thus the sum of the reward obtained by executing action *a* in state *s* at time *t* and the *Q(s',a')* values that can be obtained at time *t+1*, weighted by the probability that we reach state *s'* at time *t+1*. In this state *s'*, many actions *a'* are again possible, but we only need to consider the action *a'* that yields the maximum Q-value as we expect the agent to behave rationally. This Bellman definition is recursive, but, if we assume that the transition model and reward function are known, those Q-values can be estimated iteratively through dynamic programming. Those Q-values can also be estimated through reinforcement learning.

Points:

- 1 point if they explain what the reward function R(s,a) represents
- 1 point if they explain what the transition model P(s'|s,a) represents
- 1 point if they explain that the Q-values express the expected accumulation of rewards over time
- 1 point if they mention the discount factor \lambda and explain what it represents
- 1 point if they give the correct formula for the Bellman equation

5(a)



How can data augmentation be used to reduce the social biases and stereotypes of data-driven NLP models? Explain in a few sentences, and provide an example.

Max marks: 5

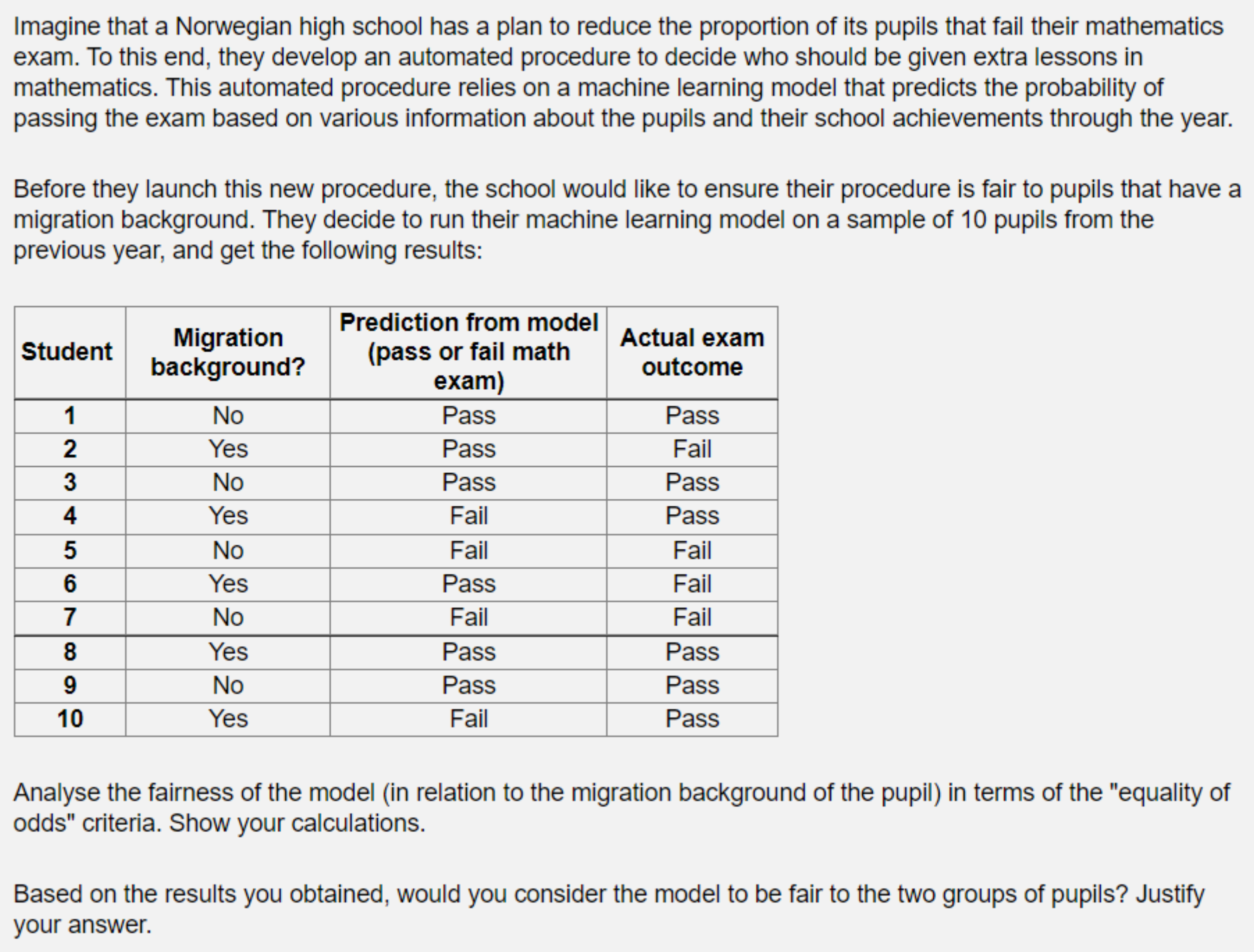
Data-driven NLP models often end up reproducing the stereotypes and biases expressed in the text corpora employed to train them. For instance, a machine translation model may produce translations that are tainted by gender stereotypes, such as when "legen" is translated into "der Arzt" in German. Similarly, a text classification model may be trained on corpora that ignore or underrepresent certain demographic groups. This may lead to a wide range of undesirable outcomes, especially since those NLP models may end up not only reflecting those existing biases and stereotypes but also reinforce them.

Data augmentation can mitigate this problem by creating a more "balanced" dataset out of the original training set. Concretely, data augmentation operates by selecting a training sample from the existing data and applying a small transformation to it. For instance, a sentence pair including the mention of a scientist referred to by a male pronoun can be edited to use a female pronoun instead (in both the source and target sentences). Similarly, a coreference resolution model may change the gender of pronouns to ensure the model does not learn social stereotypes. Through data augmentation, we can therefore obtain a larger and more balanced corpus containing both the original data samples as well as the transformed ones.

Points:

- 2 points if they explain how NLP models can be biased, and why it constitutes an ethical problem
- 1 point if they explain the general idea of data augmentation
- 1 point if they explain how data augmentation can create a more balanced dataset
- 1 point if they provide a correct example

5(b)



Imagine that a Norwegian high school has a plan to reduce the proportion of its pupils that fail their mathematics exam. To this end, they develop an automated procedure to decide who should be given extra lessons in mathematics. This automated procedure relies on a machine learning model that predicts the probability of passing the exam based on various information about the pupils and their school achievements through the year.

Before they launch this new procedure, the school would like to ensure their procedure is fair to pupils that have a migration background. They decide to run their machine learning model on a sample of 10 pupils from the previous year, and get the following results:

Student	Migration background?	Prediction from model (pass or fail math exam)	Actual exam outcome
1	No	Pass	Pass
2	Yes	Pass	Fail
3	No	Pass	Pass
4	Yes	Fail	Pass
5	No	Fail	Fail
6	Yes	Pass	Fail
7	No	Fail	Fail
8	Yes	Pass	Pass
9	No	Pass	Pass
10	Yes	Fail	Pass

Analyse the fairness of the model (in relation to the migration background of the pupil) in terms of the "equality of odds" criteria. Show your calculations.

Based on the results you obtained, would you consider the model to be fair to the two groups of pupils? Justify your answer.

Max marks: 10

The equality of odds criteria states that that the prediction $\hat{Y}$ of the model should be conditionally independent to the protected attribute A (in this case the migration background), given the true outcome Y:

$P_{Migrant}(\hat{Y} \mid Y) = P_{Not\text{-}Migrant}(\hat{Y} \mid Y)$

As we have access to both the actual outcomes and the model predictions, we can easily compute those measures:

$P_{Migrant}(\hat{Y}=pass \mid Y=pass) = 1 / 3$ while $P_{Not\text{-}Migrant}(\hat{Y}=pass \mid Y=pass) = 3 /3$

And

$P_{Migrant}(\hat{Y}=fail \mid Y=fail) = 0 /2$ while $P_{Not\text{-}Migrant}(\hat{Y}=fail \mid Y=fail) = 2/2$

The equality of odds criteria is thus doubly not satisfied:

- Students with a migration background have a higher chance of being wrongly "flagged" as needing extra lessons
- And the students with a migration background that do need help are also more likely to be ignored.

This violation of the equalized odds criteria occurs even though the proportion of students predicted to fail the exam (according to the model) is the same in both groups (40%). Intuitively, we can see that the model provides more accurate predictions for the non-migrant student group than for the migrant group. This is ethically unfair, as the student group with a migration background is less likely to benefit from those extra lessons, as the pupils that will be offered them will not be the ones needing them the most.

Points:

- 3 points for the explanation of the equality of odds criteria
- 4 points for the calculations of the conditional probabilities in both groups
- 3 points for the ethical discussion about whether the model is fair to both groups.