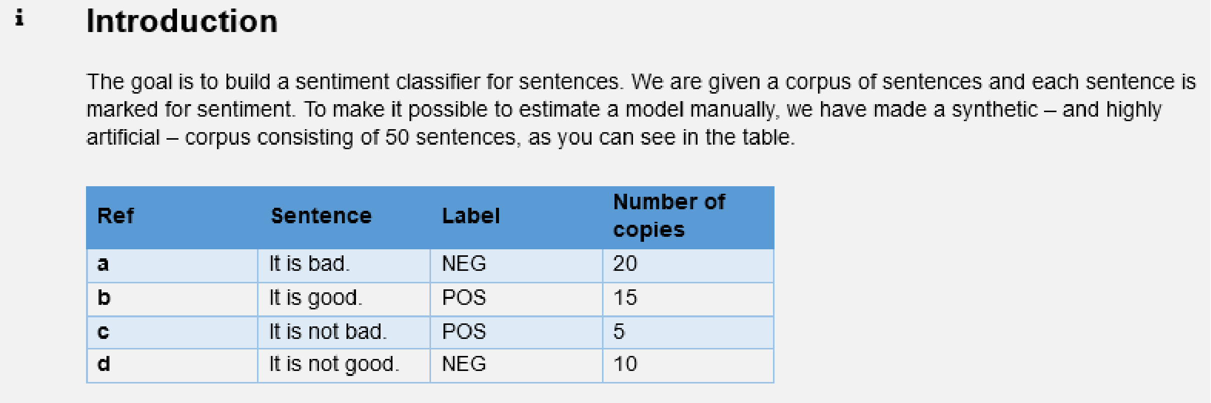
Grading guidelines IN4080 Natural Language Processing 2020

General guidelines

This is an “open-book” exam, and the grading should take that into consideration. In questions asking for concepts, models, etc., e.g. (1a), (2a), (3a), (3b), it is important that the student shows a good **understanding**. The students should give explanations that are intelligible, consistent and comprehensive. It is not sufficient to reproduce parts of the teaching material that partly answer the questions.

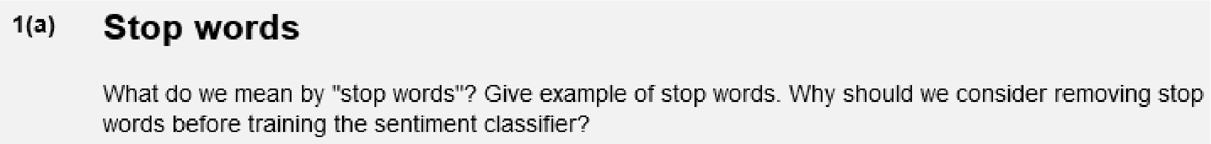
Exercise 1: Text classification



i **Introduction**

The goal is to build a sentiment classifier for sentences. We are given a corpus of sentences and each sentence is marked for sentiment. To make it possible to estimate a model manually, we have made a synthetic – and highly artificial – corpus consisting of 50 sentences, as you can see in the table.

Ref	Sentence	Label	Number of copies
a	It is bad.	NEG	20
b	It is good.	POS	15
c	It is not bad.	POS	5
d	It is not good.	NEG	10



1(a) **Stop words**

What do we mean by "stop words"? Give example of stop words. Why should we consider removing stop words before training the sentiment classifier?

Maximum marks: 3

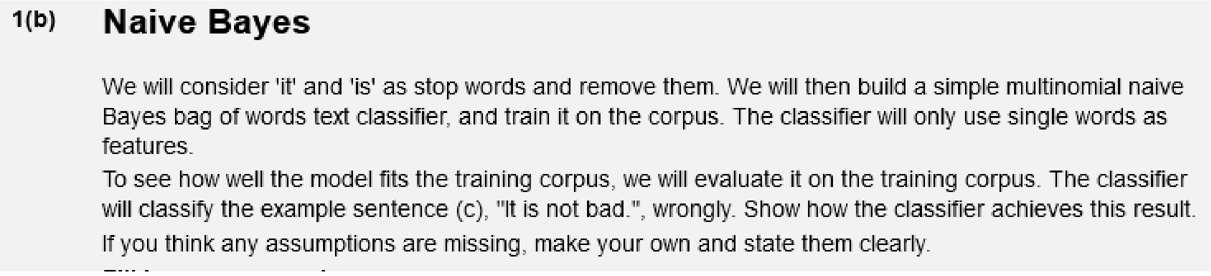
The following should be taken into consideration:

Stop words

- Removed before further processing
- Words that lack content, functional words, pronouns
- Frequent words, but not all frequent words, e.g. *not*
- Stop word lists, no fixed set

Why

- They don’t contribute positively
- They might confuse by giving attention to irrelevant features
  - (But for many models, they are harmless)



1(b) **Naive Bayes**

We will consider 'it' and 'is' as stop words and remove them. We will then build a simple multinomial naive Bayes bag of words text classifier, and train it on the corpus. The classifier will only use single words as features.

To see how well the model fits the training corpus, we will evaluate it on the training corpus. The classifier will classify the example sentence (c), "It is not bad.", wrongly. Show how the classifier achieves this result.

If you think any assumptions are missing, make your own and state them clearly.

Maximum marks: 10

A correct solution:

The class probabilities:
P(NEG) = 30/50 = 0.6
P(POS) = 20/50 = 0.4

The vocabulary = {good, bad, not}

		Class		
	P(t|C)	POS	NEG	
Terms	bad	5/25=0.2	20/40=0.5	
	good	15/25=0.6	10/40=0.25	
	not	5/25=0.2	10/40=0.25	

Classifying sent_c
p) P(POS | sent_c) = P(POS)*P(bad | POS) *P(not | POS)/P(sent_c) = 0.4*0.2*0.2/P(sent_c) = 0.016/P(sent_c)
n) P(NEG|sent_c) = P(NEG)* P(bad | NEG) *P(not | NEG)/P(sent_c) = 0.6*0.5*0.25/P(sent_c) = 0.075/P(sent_c)
Hence it chooses the class NEG
+++++++++++++

A solution should include

1. The goal is to compare P(POS | sent_c) to P(NEG|sent_c)
2. The formulas for the expressions, e.g. P(POS | sent_c) = P(POS)*P(bad | POS) *P(not | POS)/P(sent_c)
3. How the multinomial model determines P(bad | POS), etc.
4. Getting the various probabilities correct e.g. P(NEG)=0.6, P(bad | POS)=0.2, etc.
5. The final calculation

Subtract 1 or 2 points for mistakes at each of these points.
For example, an otherwise correct solution using the Bernoulli model below should get 8 points.

+++++++++++++
With Bernoulli

		Class			
		POS		NEG	
		P(t=1|POS)	P(t=0|POS)	P(t=1|NEG)	P(t=0|NEG)
Terms	bad	5/20=0.25	15/20=0.75	20/30=2/3	10/30=1/3
	good	15/20=0.75	5/20=0.25	10/30=1/3	20/30=2/3
	not	5/20=0.25	15/20=0.75	10/30=1/3	20/30=2/3

Classifying sent_c
P(POS | sent_c) = P(POS)*P(bad=1 | POS) *P(not=1 | POS)*P(good=0 | POS)/P(sent_c) = 0.4*0.25*0.25*0.25/P(sent_c) = 0.00625/P(sent_c)
P(NEG|sent_c) = P(NEG)* P(bad=1 | NEG) *P(not = 1 | NEG)*P(good=0 | NEG/P(sent_c) = 0.6*(2/3)(1/3)(2/3)/P(sent_c) = 0.6*4/(3*9)/P(sent_c)=0.8/9P(sent_c) = 0.089/P(sent_c)
Hence it chooses the class NEG

Sent_d:
POS: 0.4*0.75*0.75*0.25/P(sent_d) = 0.057/P(sent_d)
NEG: 0.6*(1/3)*(1/3)*(1/3)/P(sent_d) = 0.2/9P(sent_d)= 0.022/P(sent_d)
Chooses POS

1(c) Accuracy

What is the accuracy of the classifier on the training corpus?

Maximum marks: 4

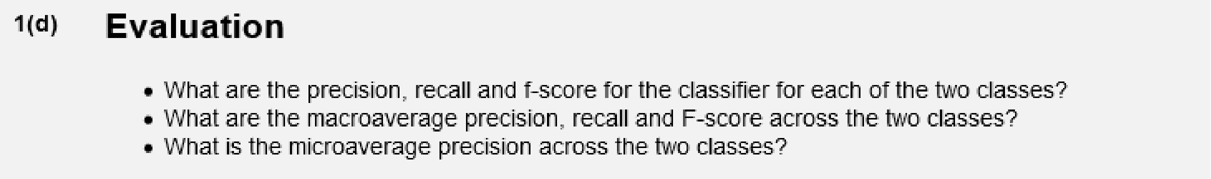
Similarly to above we see that the classifier will classify sentence (d) wrongly as POS.

Classifying sent_d
p) P(POS | sent_d) = P(POS)*P(good | POS) *P(not | POS)/P(sent_c) = 0.4*0.6*0.2/P(sent_c) = 0.048/P(sent_d)
n) P(NEG|sent_d) = P(NEG)* P(good | NEG) *P(not | NEG)/P(sent_c) = 0.6*0.25*0.25/P(sent_c) = 0.0375/P(sent_d)
Hence it chooses the class NEG

This yields the following confusion table

		Gold		
		POS	NEG	sum
Predicted	POS	15 (sent_b)	10 (sent_d)	25
	NEG	5 (sent_c)	20 (sent_a)	25
	sum	20	30	

With an accuracy of (20+15)/50=0.7



1(d) Evaluation

- What are the precision, recall and f-score for the classifier for each of the two classes?
- What are the macroaverage precision, recall and F-score across the two classes?
- What is the microaverage precision across the two classes?

Maximum marks: 6

- POS:
  - P =15/25=0.6
  - R=15/20=0.75
  - F= 2PR/(P+R)=2*0.6*0.75/(0.6+0.75)=0.9/1.35=2/3
- NEG:
  - P=20/25=0.8
  - R=20/30=2/3
  - F=2*0.8*(2/3)/(0.8+2/3)=2*0.8*2/(2.4+2)=3.2/4.4=8/11=0.73
- MACRO_P=(0.6+0.8)/2=0.7
- MACRO_R=(3/4+2/3)/2=(9+8)/2*12=17/24
- Macro_F=(F_POS + F_NEG)/2 = (2/3 + 8/11)/2 = (2*11 + 8*3)/(3*8*11)=23/33 = 0.7

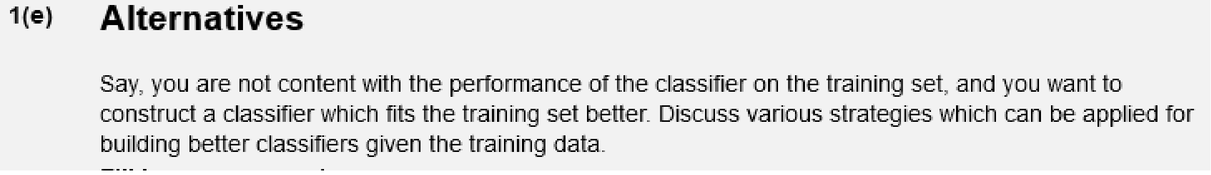
For MICRO_P=Acuracy=0.7 for all binary classifiers (and so is MICRO_R)

Alternatively, we could use the pooled table

		Gold		
		POS	NEG	sum
Predicted	POS	15+20	10+5	50
	NEG	5+10	20+15	50
	sum	50	50	

MICRO-P = 35/50=0.7

2 points for each of the three questions.



1(e) **Alternatives**

Say, you are not content with the performance of the classifier on the training set, and you want to construct a classifier which fits the training set better. Discuss various strategies which can be applied for building better classifiers given the training data.

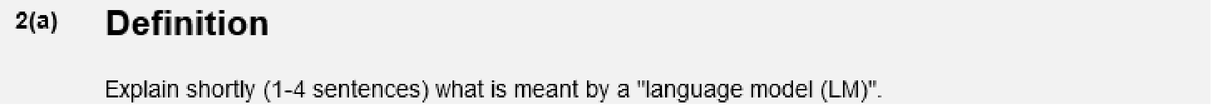
Maximum marks: 7

This is a relatively open question where the students should show understanding. The discussion should consider two aspects

- Other learners than NB, e.g. logistic regression or neural networks
- Features, e.g. combining features,like *not_bad*

A stellar solution would discuss interactions bewteen the modifications, e.g. the extra features can give a perfect classifier already with NB in this particular case. LR can imprve the results even with the given features.

Exercise 2 Language models



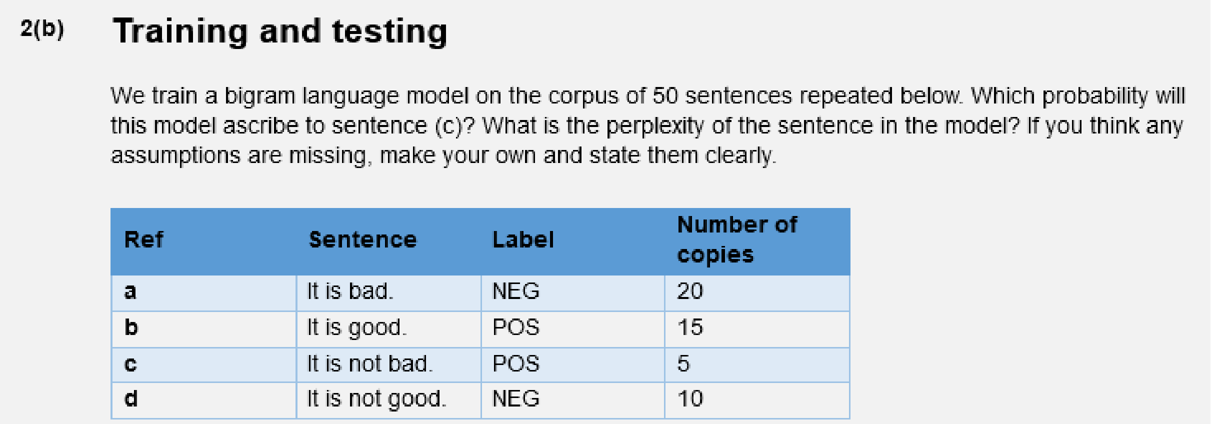
2(a) **Definition**

Explain shortly (1-4 sentences) what is meant by a "language model (LM)".

Maximum marks: 3

A language model ascribes probabilities to sequences of words P(w1 w2 … wn).
Alternatively, it can be described as ascribing a probability to a word in a sequence giving the preceding words, P(wn | w1, w2,…, w-(n-1))
The two definitions are interchangeable because
P(w1 w2 …wn) = P(w1)*(P(w2 | w1)*…* P(wn | w1, w2,…, w-(n-1)), and
P(wn | w1, w2,…, w-(n-1)) = P(w1 w2 …wn)/ P(w1 w2 …w-(n-1))

3 points for having one of the two definitions correct. One additional point for having both definitions and pointing out the connection between them.



2(b) **Training and testing**

We train a bigram language model on the corpus of 50 sentences repeated below. Which probability will this model ascribe to sentence (c)? What is the perplexity of the sentence in the model? If you think any assumptions are missing, make your own and state them clearly.

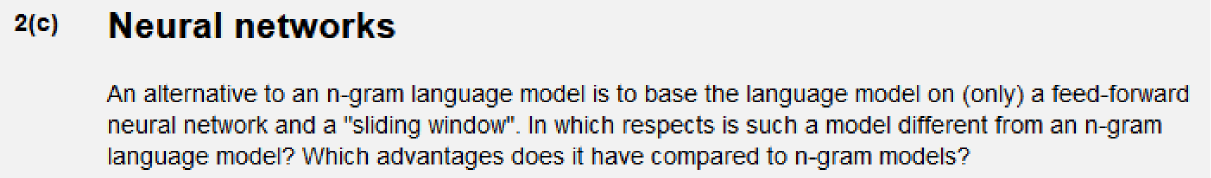
Ref	Sentence	Label	Number of copies
a	It is bad.	NEG	20
b	It is good.	POS	15
c	It is not bad.	POS	5
d	It is not good.	NEG	10

Maximum marks: 7

P(<s> it is not bad <\s>) = P(it | <s>)P(is | it)P(not | is)P(bad | not)P(. | bad)P(<\s> | .) = 1*1*(15/50)*(5/15)*(25/25)*1=0.1

Perplexity = 1/(0.1)**(1/6)=1.47

4 points for the probability, 3 points for the perplexitiy



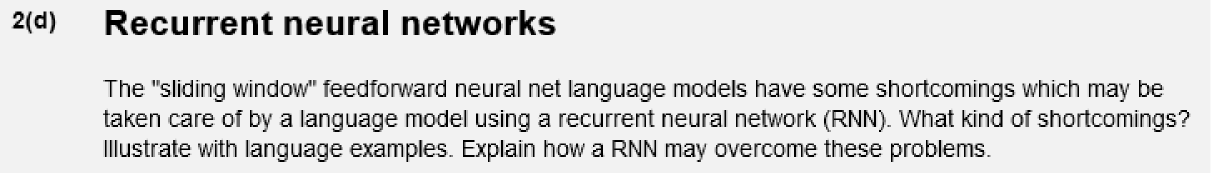
2(c) **Neural networks**

An alternative to an n-gram language model is to base the language model on (only) a feed-forward neural network and a "sliding window". In which respects is such a model different from an n-gram language model? Which advantages does it have compared to n-gram models?

Maximum marks: 5

The n-gram model is based on counting frequencies of n-grams in a large text corpus.
The ffnn language model will instead be based on a machine learning task where the goal is to predict a word, w, from k preceding words for a fixed k, or more precisely learn a probability distribution P(w_n | w_(1+n-k)…w_(n-1)), where w_n varies over the vocabulary. The ffnn language model uses embeddings to represent each word. (3p)

It ill be able to generalize to a word from similar words. In particular, this makes it possible for the model to make prediction from an earlier unseen context w_(1+n-k)…w_(n-1) relying on the similarities to other embeddings for each w_i. (2p)



2(d) **Recurrent neural networks**

The "sliding window" feedforward neural net language models have some shortcomings which may be taken care of by a language model using a recurrent neural network (RNN). What kind of shortcomings? Illustrate with language examples. Explain how a RNN may overcome these problems.

Maximum marks: 5

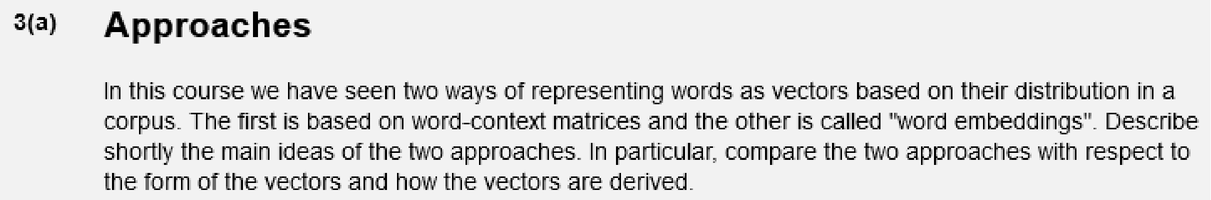
The problem is that the ffnn LM is similar to an n-gram model in the respect that it predicts a word on the basis of a fixed number of preceding words, e.g. 3 preceding words. In natural language, however, there might be dependencise between words that are arbitraily far apart in the sentence, e.g.
*The cows that were seen behind the barn in the valley, are (*is)*

An RNN will read in the words in a sequence one-by-one. After each word it will produce a history state, *h*. The history after word n will be determined by the word wn and the history after the preceding word, h_(n-1). Since h_(n-1) again wil be determined by h_(n-2) and w_(n-1) and so forth, the history h_n will be determined by all the words in the sequence up to and including w_n.

The RNN LM will predict word w_n from h_(n-1), i.e. P(wn | w1, w2,…, w-(n-1)) = P(wn | h_(n-1)). And since all the preceding words w1, w2,…, w-(n-1) influence h_(n-1), they may also participate in prediciting w_n.

1 p for shortcomings
2 p for examples
2 p for explaining the RNN

Exercise 3 Embeddings (20 points)



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.

Maximum marks: 10

Word-contexts

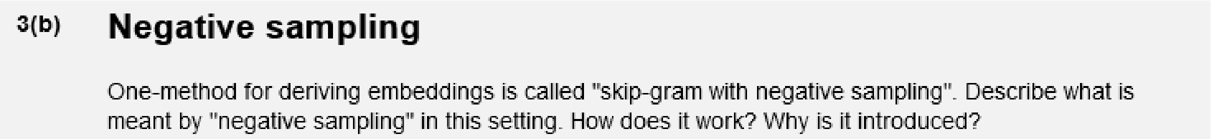
- Description of the word-context matrix (3p)
  - What are the entries?
  - What is a context
- How are the numbers calculated (1 p)

Embeddings

- Fixed number of reals (1p)
- Prediciton task (3p)

Contrasts

- Counting vs. predicition (1p)
- Sparse vs. dense vectors (1 p)



3(b) **Negative sampling**

One-method for deriving embeddings is called "skip-gram with negative sampling". Describe what is meant by "negative sampling" in this setting. How does it work? Why is it introduced?

Maximum marks: 6

Negative sampling (3 points)

- Where do the positive examples come from?
- How are the negative examples derived?
- The goal of the learning task: Spearate between the positive and negative examples

Why (3 points)

- Overreaching goal in the skip-gram model: For a given wi, predict P(w | wi) of w occurring within a context window of wi, for all words w in the vocabulary
- This could be trained from the positive examples alone using soft-max. For example one occurrence of *jam* in the context of *apricot* would raise the probability P(jam | apricot).
- But this means the example would also decrease P(w | apricot) for all other words w. This is quite expensive as one would have to update the weights for all the predicted words based on each training instance.

Exam questions for IN4080, Autumn 2020: part on dialogue systems and ethics

Pierre Lison

December 8, 2020

Question 4: dialogue systems

Question 4(a) (5 points)

Why is it important to thing about (conversational) *grounding* when developing dialogue systems? Support your answer with a few examples.

Human conversations are collaborative processes in which dialogue participants continuously make sure that they remain “on the same page” – that is, they actively listen to one another and strive to understand each other’s contributions to the dialogue. This is done through the production of various *grounding signals*, such as backchannels, (explicit and implicit) communicative feedbacks, clarification requests, and repairs. By producing and interpreting grounding signals, conversational partners are able to gradually expand the interaction’s *common ground*, which is defined as the knowledge shared by all dialogue participants.

Conversational grounding is also a key aspect of human-machine interactions, and more specifically of dialogue systems. Human users must be aware of what the system has understood and what is hasn’t understood (or is uncertain about). For instance, if the system has not understood a particular user input, it needs to be able to utter a clarification request (”sorry I did not get that. Could you repeat?”) and understand the human response (”I meant that ..”). The production of communicative feedbacks (such as ”uh-uh” or ”ok got it”) is also useful to convey to the user that their utterances have been processed and understood. The production and interpretation of grounding signals make conversations with dialogue systems both more intuitive (as it more closely reflects the properties of human-human conversations) and more efficient (as it allows misunderstandings to be detected and repaired early).

Points: Students should, at a minimum, describe what grounding is (in their own words) and what kind of communicative signals/strategies (like backchannels or clarification requests) can be used to perform this grounding (2.5 points). They should also explain why grounding is important when designing dialogue systems, namely inform the user about what has been processed/understood and was hasn’t, and in the latter case seek to repair the misunderstanding.

Question 4(b) (6 points)

Calculate the Word Error Rate (WER) between this utterance:

could you go to my office and pick up my NLP book

And the recognition hypothesis generated by a speech recogniser:

could you got you my office and pickup my NLB book

Show your calculations using an edit distance matrix. You can assume that insertions, deletions and substitutions all have a cost of 1.

Let us construct the edit distance matrix:

	could	you	go	to	my	office	and	pick	up	my	NLP	book
could	0	1	2	3	4	5	6	7	8	9	10	11
you	1	0	1	2	3	4	5	6	7	8	9	10
got	2	1	1	2	3	4	5	6	7	8	9	10
you	3	2	2	2	3	4	5	6	7	8	9	10
my	4	3	3	3	2	3	4	5	6	7	8	9
office	5	4	4	4	3	2	3	4	5	6	7	8
and	6	5	5	5	4	3	2	3	4	5	6	7
pickup	7	6	6	6	5	4	3	3	4	5	6	7
my	8	7	7	7	6	5	4	4	4	4	5	6
NLB	9	8	8	8	7	6	5	5	5	5	5	6
book	10	9	9	9	8	7	6	6	6	6	6	5

Since we have 12 number of words in the “gold standard” transcription, the word error rate WER is therefore $100 \times \frac{5}{12} = 41.7$ %.

Intuitively, we see that we can go from the recognition hypothesis to the gold transcription with 5 operations:

- replacing “got” by “go”
- replacing “you” with “to”
- replacing “pickup“with “pick”
- inserting “up”
- replacing “NLB” with “NLP”.

You wish to develop a (phone-based) spoken dialogue system that will call random U.S. citizens in order to collect opinion poll data for the next US election, with two candidates on the ballot: Kamala Harris and Ivanka Trump.
This system is framed as a *Markov Decision Process* (MDP) formalised as such:

- We have five possible states:

  s_1 is the starting state

s_2 if the callee indicated their intention to vote for Kamala Harris

s_3 if the callee indicated their intention to vote for Ivanka Trump

s_4 if the callee expressed something else (that was not understood)

- The set of actions that can be taken by the dialogue system are as follows:

  a_1 : Say "Hi, I'm a automated bot developed to collect polling data. May I ask you for whom you plan to vote in the next election?"

  a_2 : Say "Sorry I did not understand. Who do you wish to vote for?"

  a_3 : Say "Ok, thank you for your help, and have a nice day!"

- The transition model is as follows:
  - In state s_1, only action a_1 is possible, with three possible transitions:

$$P(s'=s_2|s=s_1,a=a_1) = 0.48, P(s'=s_3|s=s_1,a=a_1) = 0.40$$
$$P(s'=s_4|s=s_1,a=a_1) = 0.12$$

  - In states s_2 and s_3, only a_3 is possible and terminates the dialogue.
  - In state s_4, only a_2 is possible, with the following transitions:

$$P(s'=s_2|s=s_4,a=a_2) = 0.36$$
$$P(s'=s_3|s=s_4,a=a_2) = 0.32$$
$$P(s'=s_4|s=s_4,a=a_2) = 0.32$$

- Finally, the reward model is defined as such:
  - $R(s = s_2, a = a_3) = R(s = s_3, a = a_3) = 10$
    (if the system manages to register the callee's political preference)
  - $R(s = s_4, a = a_2) = -1$
    (to capture the annoyance of asking the callee to repeat).
  - Other actions have a reward of zero.

Question 4(c) (8 points)

Based on this model, calculate the expected cumulative reward of asking the callee to repeat when their answer was not properly understood, that is: $Q(s = s_4, a=a_2)$. You can assume a discount factor of 0.9.

We will be using the famous Bellman equation to calculate our answer:

$$Q(s,a) = R(s,a) + \lambda \sum_{s' \in S} P(s'|s,a) \max_{a'} Q(s',a') \tag{1}$$

In our case, we already know $R(s_4, a_2)$, which is -1. We also know that from s_4, three transitions are possible (to s_2, s_3 and s_4). We can thus write:

$$Q(s_4, a_2) = -1 + 0.9 \left(0.36 \max_{a'} Q(s_2, a') + 0.32 \max_{a''} Q(s_3, a'') + 0.32 \max_{a'''} Q(s_4, a''')\right) \tag{2}$$

Now, in states s_2 and s_3, only action a_3 is possible, and terminates the dialogue. And in state s_4, only action a_2 is possible. That means that $\max_{a'} Q(s_2, a')$ and $\max_{a''} Q(s_3, a'')$ can be reduced to $R(s_2, a_3)$ and $R(s_3, a_3)$, which gives us:

$$Q(s_4, a_2) = -1 + 0.9 \left(0.36 \times 10 + 0.32 \times 10 + 0.32 Q(s_4, a_2)\right) \tag{3}$$

The equation above is a standard linear equation with one unknown:

$$\begin{align} Q(s_4, a_2) &= -1 + 0.9 \times 3.6 + 0.9 \times 3.2 + +0.9 \times 0.32 \times Q(s_4, a_2) \tag{4} \\ Q(s_4, a_2) - 0.288 Q(s_4, a_2) &= 5.12 \tag{5} \\ Q(s_4, a_2) &= \frac{5.12}{0.712} \approx 7.19 \tag{6} \end{align}$$

Points: 3 points if they use Bellman's equation, 3 more points if they correctly replace the probabilities and rewards as shown above, and 2 final points if they get the right numerical answer at the end.

Question 4(d) (3 points)

One limitation of this MDP model is that is assumes that the dialogue system will always be 100 % certain it has correctly understood the political preference expressed by the callee. In practice, this will not always be the case, because of e.g. speech recognition or NLU errors, or because the callee may intentionally provide unclear or misleading information. How could this model be adapted to capture those uncertainties?

One way to adapt this MDP model would be to extend it to a *partially observable* MDP, or POMDP for short. In a POMDP, the actual dialogue state is not known with certainty, but is a probability distribution over possible state values. Due to this probabilistic account of the dialogue state, a POMDP could capture the difference between what is observed (for instance speech recognition hypotheses) and what the "true" state can be (in this case, the voting intentions of the callee).

Points: 1.5 if they mention the term POMDP, 1.5 more points if they explain how a POMDP can capture uncertainties.

You have developed an NLP model for automated essay scoring in Norwegian, and you wish to ensure your model is *fair*, in particular when it comes to whether the student is ethnically Norwegian or not.
To this end, you compare the essay scores with scores assigned by experienced teachers. To simplify our problem we will rely on binary pass/fail scores. In addition, we will assume that the human teachers themselves are free from social biases regarding the ethnicity of the students.

Here are the scores produced by your model and by the human teachers for a group of 21 students:

ID	Ethnical	Score from model:	Score from teachers:
	Norwegian?	Pass (✓) or Fail (**F**)	Pass (✓) or Fail (**F**)
1	✓	✓	✓
2	No	✓	✓
3	✓	✓	✓
4	✓	**F**	**F**
5	✓	✓	✓
6	✓	**F**	**F**
7	✓	✓	✓
8	No	**F**	**F**
9	No	✓	✓
10	✓	✓	✓
11	✓	✓	✓
12	No	✓	✓
13	✓	✓	✓
14	No	**F**	**F**
15	✓	✓	✓
16	✓	✓	**F**
17	No	✓	✓
18	✓	**F**	✓
19	No	✓	**F**
20	No	**F**	✓
21	✓	✓	**F**

Question 5(a) (10 points)

Based on this data, determine which fairness criteria[a] covered during the course (demographic parity, predictive parity and equalised odds) are satisfied or not satisfied by your essay scoring model.

[a] We assume the essay scoring model does not have direct access to the ethnicity of the student, and the "unawareness" criteria is thus irrelevant here.

First, some notations:

- The two demographic groups will be written eno (ethnically Norwegian) and $\neg eno$ (non-ethnically Norwegian).
- $\hat{Y}$ corresponds to the predictions of the model
- Y corresponds to the scores from the human teachers (which we assume in this exercise to be bias-free, and thus corresponds to some "true" value)

We can then look at various fairness criteria:

Demographic fairness We need to look whether the probabilities of getting a pass or fail are the same across the two groups:

$$P_{eno}(\hat{Y}) \overset{?}{=} P_{\neg eno}(\hat{Y}) \tag{7}$$

For the eno group, 3 out of 13 students get a fail from the scoring model, while this proportion rises to 3 out of 8 students for the $\neg eno$ group. The demographic fairness criteria is therefore *not* satisfied.

Predictive parity We need to look at the *precision* of our model predictions (compared to the scores provided by the human teachers):

$$P_{eno}(Y = y|\hat{Y} = y) \overset{?}{=} P_{\neg eno}(Y = y|\hat{Y} = y) \tag{8}$$

We can start with the value $y = \checkmark$. For the eno group, we have 10 students that get a pass from the model. 8 of those students also get a pass from the human teachers, which means that the precision $P_{eno}(Y = \checkmark|\hat{Y} = \checkmark) = 0.8$.

For the $\neg eno$ group, we have 5 students that get a pass from the model, and 4 of them also get a pass from the human teachers. The precision $P_{\neg eno}(Y = \checkmark|\hat{Y} = \checkmark)$ is thus also equal to 0.8.

Now, for the value $y = \mathbf{F}$, we can do the same calculations: for the eno group, 3 students failed, and 2 of them were also marked as failed by the human teachers, giving a precision of 2/3. For the $\neg eno$ group, 3 students failed as well, as 2 were marked as failed by human teachers, which also gives a precision of 2/3.

In other words, the predictive parity criteria is satisfied.

Equalised odds We need to look at the *recall* of our model predictions compared to the scores provided by the human teachers:

$$P_{eno}(\hat{Y} = y|Y = y) \overset{?}{=} P_{\neg eno}\hat{Y} = y|Y = y) \tag{9}$$

We can start with the value $y = \checkmark$. For the eno group, we have 9 students that get a pass from the human teachers. 8 of those students also get a pass from the model, which means that the recall is 8/9.

For the $\neg eno$ group, we have 5 students that get a pass from the human teachers, and 4 of them also get a pass from the human model, giving a recall of 8/10.

In other words, the non-ethnical Norwegians will have a higher risk of being a false positive (receiving a fail mark when one should have gotten a pass). An ethnical Norwegian that should receive a pass will have a 11 % change of being mistakenly scored as failed, while this risk increases to 20 % for students that are non ethnical Norwegians.

We do the same calculations for the value $y = \mathbf{F}$: for the eno group, 4 students failed according to the teachers, and 2 of them were also marked as failed by the model, giving a recall of 1/2. For the $\neg eno$ group, 3 students were failed by the human teachers, and 2 of them were marked as failed by the model, which also gives a recall of 2/3.

The criteria of equalised odds is thus *not* satisfied.

Points 2 points if they understand the basic principles of those fairness criteria. 2 points for the demographic fairness, 3 points for the predictive parity and 3 points for the equalised odds.

Question 5(b) (2 points)

Would you consider your model as being fair to the students that are not ethnical Norwegian? Explain your answer.

Several answers are possible here, the key idea is to see whether the student can show an understanding of the type of ethical reasoning necessary to assess the fairness criteria described above.

Personally, I would say that it is fine if the demographic fairness criteria is not satisfied: whether a student is ethnically Norwegian or not is presumably correlated with their fluency in Norwegian. And the fluency in Norwegian should be allowed to influence the likelihood of getting a pass/fail score to evaluate the quality of an essay.

However, the fact that the equalised odds criteria is not satisfied is much more problematic. As mentioned above, it means that a "good" student (that should receive a pass) will have a higher chance of being mistakenly attributed a failing score if they are not ethnical Norwegian. And the difference is fairly large, since the non-ethnical Norwegians will have a 20 % risk, compared to an 11 % risk for the ethnical Norwegians. In this light, I would not consider the scoring model to be fair.