Homework 3

Ana Sofia Carmo, 83810

Deep Structured Learning, 2021/22

Question 1

1.(a)

Preprocessing the dataset prior to transliteration requires the identification of all the characters that comprise the words of both languages (from the training set only). Given the Arabic-English transliteration data released by Google, this corresponds to vocabulary sizes of 50 and 42 for Arabic and English, respectively. Note that these vocabulary sizes also include three additional tokens corresponding to the special symbols that mark the start (`<sos>`) and end (`<eos>`) of a sequence, as well as unknown characters (`<unk>`)[1].

1.(b)

The vanilla encoder-decoder architecture, as the name implies, is comprised of two units: an encoder and a decoder, as illustrated in Figure 1. The encoder unit contains multiple Long-Short Term Memory (LSTM) cells, which take the individual tokens from the input (Arabic) sequence. Although (in the vanilla architecture) the actual output of each cell is not used, the hidden tensors that are returned by each cell are fed into the following cell. By propagating the hidden representations throughout the cells, the encoder unit infers the context of the token sequence and creates a dense representation accordingly.

On the other hand, the decoder unit is responsible for interpreting the context provided by the encoder, as well as the immediate antecedent, in order to predict the next token. As such, it is uses the previous hidden state as contextual information, as well as the previous token (or token from the actual target sequence, if the we are using the teacher forcing method).

In the vanilla approach, the encoder is required to compress the contextual information from the whole sequence into a fixed-length tensor (hidden size), which may present a performance issue when working with particularly long sequences. Nevertheless, for the

[1] For simplicity, the chosen batch size was 1 in order to simplify the implementation (i.e. no need for padding).

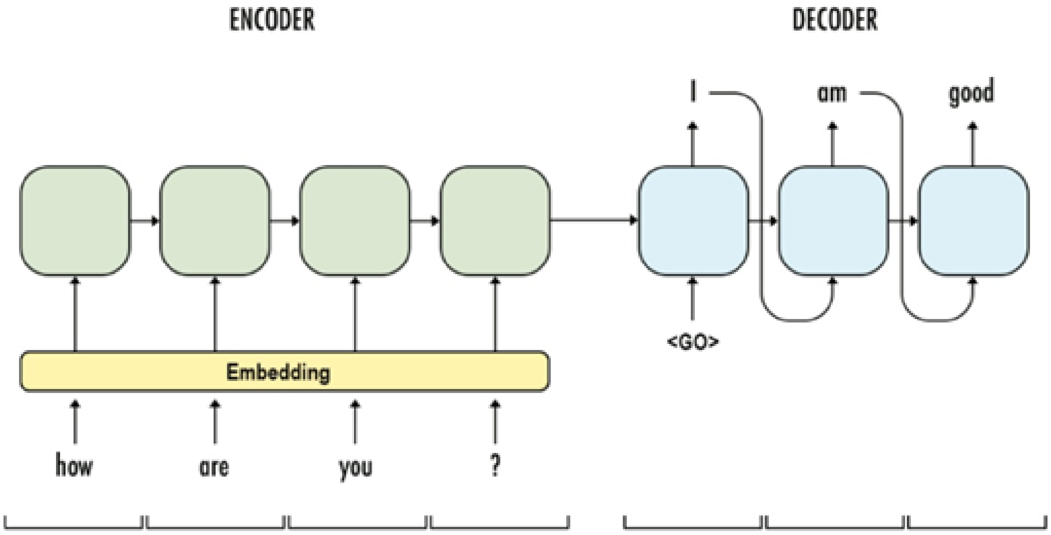


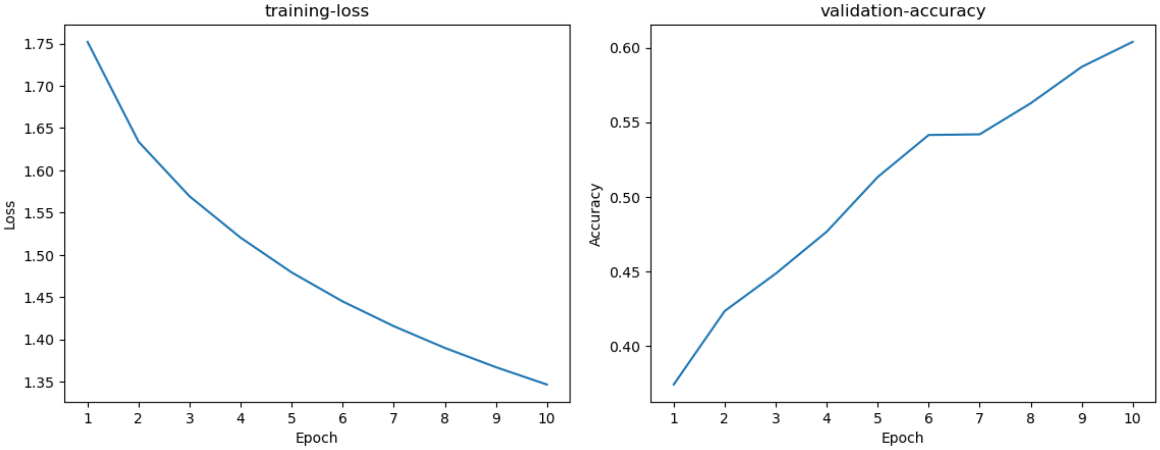
Figure 1: Simplified illustration of the vanilla encode-decoder architecture. Retrieved from "A Formalization of a Simple Sequential Encoder-Decoder", Medium (2019).

particular case of performing the transliteration of small words, this approach should have an adequate performance.
Figures 2(A) and 2(B) provide the loss over training epochs and the corresponding accuracy on the validation set, respectively. Both the training loss and validation accuracy display an suitable trend over the training epochs (decreasing and decreasing, respectively). Moreover, although the test accuracy was not satisfactory (Acc = 0.6054), I believe the architecture would perform significantly better if a slightly more complex network was chosen[2].

1.(c)

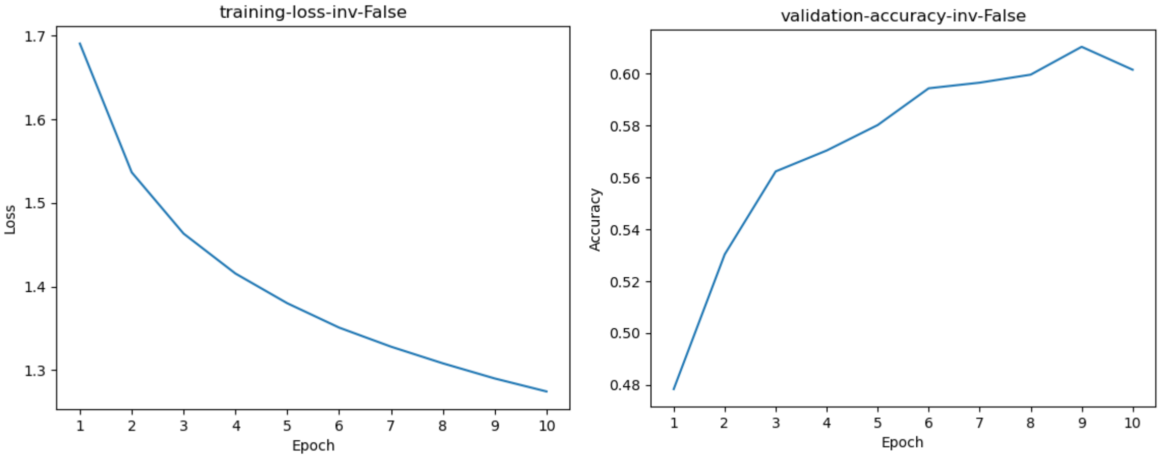
Training and testing the vanilla encoder-decoder model with the source (Arabic) strings inverted held a similar performance to the one obtained in the previous exercise, displaying a slight improvement (Acc = 0.6157) - see Figures 3(A) and 3(B) for the corresponding training evolution. One could hypothesize that this is related to the fact that, in Arabic, characters are written from right to left - the opposite of English. As such, this could impair the model in the identification of corresponding morphological structures (e.g. prefixes and suffixes) between the two sequences - which would be solved when inverting one of the sequences.

[2] The number of epochs and model parameters (namely hidden and embedding dimensions, as well as number of layers) were decreased to ill-advised values in order to decrease computational complexity of the training process.



(A) (B)

Figure 2: Results from the implementation of the vanilla encoder-decoder architecture with two unidirectional LSTMs. (A) Loss over training epochs and (B) the corresponding accuracy on the validation set.



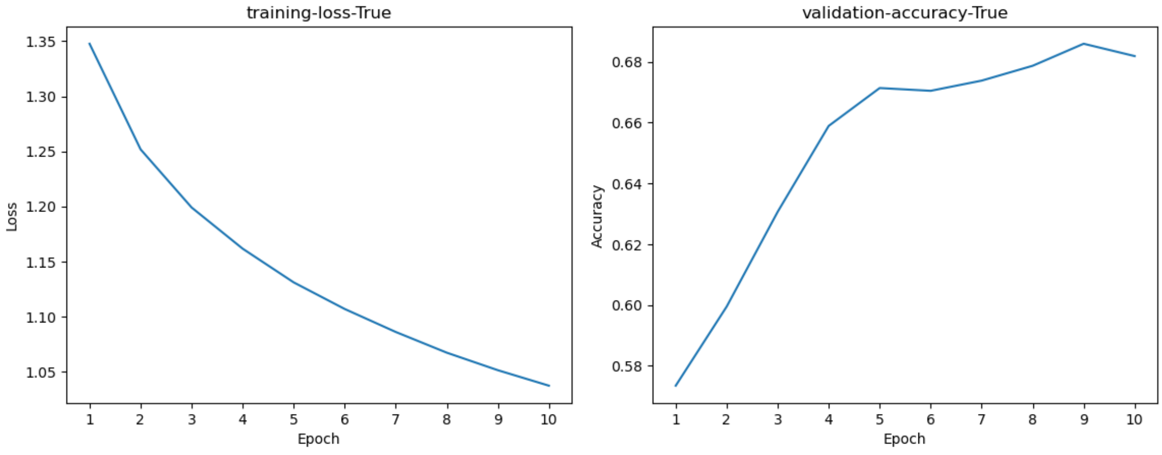
(A) (B)

Figure 3: Results from the implementation of the vanilla encoder-decoder architecture with two unidirectional LSTMs for inverted input sequences. (A) Loss over training epochs and (B) the corresponding accuracy on the validation set.

1.(d)

The attention mechanism comes as a solution to the limitation highlighted in question 1.(b). Instead of passing the successive hidden states from the encoder down through the decoder cells, the learning mechanism of this approach, at each time step, attempts to interpret which encoder cell outputs can be of more relevance.

The implementation of this approach held significant better results (Acc = 0.6864) - see Figures 4(A) and 4(B) - than its vanilla counterparts. We can hypothesize that this is due to both the attention mechanism incorporated, as well as the bidirectionality of the encoder, which allows the context tensor to include information from both directions of the sequence (which kind of serves the same purpose as the inversion of the input sequence).



(A)

(B)

Figure 4: Results from the implementation of the bidirectional encoder-unidirectional decoder architecture with attention mechanism. (A) Loss over training epochs and (B) the corresponding accuracy on the validation set.

Question 2

1.(a)

Figuring out the conditional independence relations entailed by D-separation in a graph, requires the prior identification of all (undirected) paths between the concerned variables. Figure 5 illustrates the four paths between variables `T` and `Y`. Then, for any of these paths, at least one of the following conditions must hold:

- The path includes a fork with observed parents;
- The path includes a chain with observed middle;
- The path includes a collider with unobserved descendants.

Below, is the answer to the question of conditional independence (*Yes - conditionally independent; No - not conditionally independent*) for each of the expressions, along with the D-separation logic.

1. No - at least path (a) checks none of the conditions.
2. No - at least path (d) checks none of the conditions.
3. No - at least path (a) checks none of the conditions.

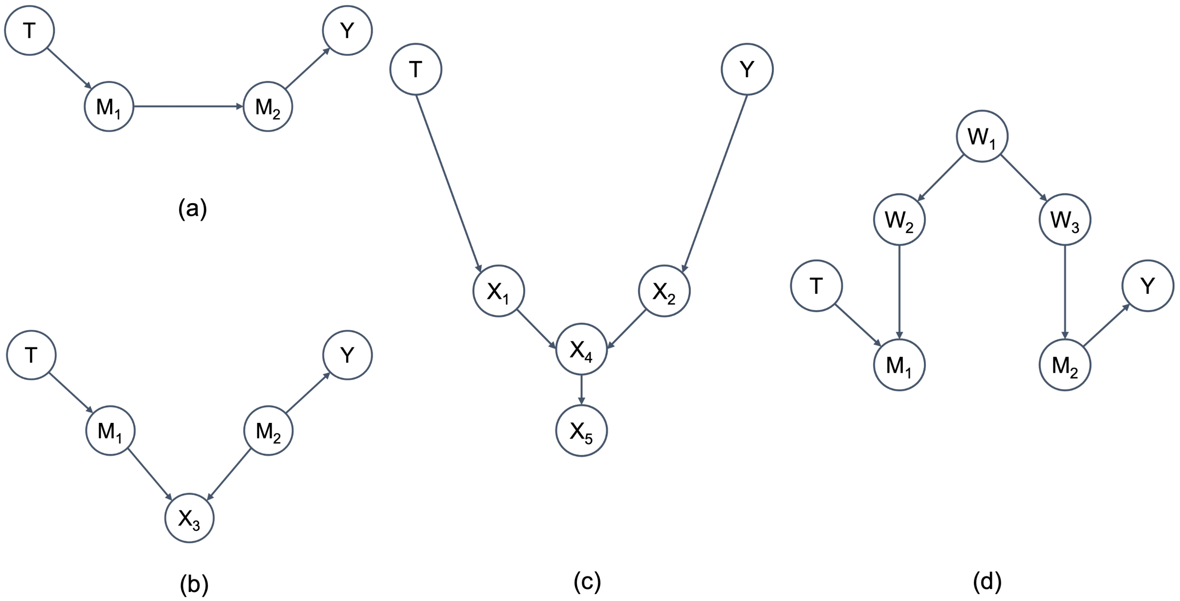


Figure 5: Paths between variables T and Y under consideration for D-separation.

4. Yes - paths (a) and (b) have at least a chain with observed middle, path (c) has at least one collider with unobserved descendants, and path (d) has at least a fork with observed parents.
5. Yes - paths (a), (b) and (c) have at least a chain with observed middle and path (d) has at least a fork with observed parents.
6. Yes - paths (a) and (b) have at least a chain with observed middle, path (c) has at least one collider with unobserved descendants, and path (d) has at least a fork with observed parents.
7. No - at least path (c) checks none of the conditions.
8. Yes - paths (a), (b) and (c) have at least a chain with observed middle and path (d) has at least a fork with observed parents.
9. No - at least path (c) checks none of the conditions.
10. Yes - paths (a), (b) and (c) have at least a chain with observed middle and path (d) has at least a fork with observed parents.

1.(b)

Figure 6 shows the Markov network originated from the conversion of the Bayes network under consideration, where the orange edges correspond to connections between nodes that were not verified in the Bayes network. ($\mathtt{T} \perp\!\!\!\perp \mathtt{W}_2$) is an example of an independence relation that is lost in the conversion process.

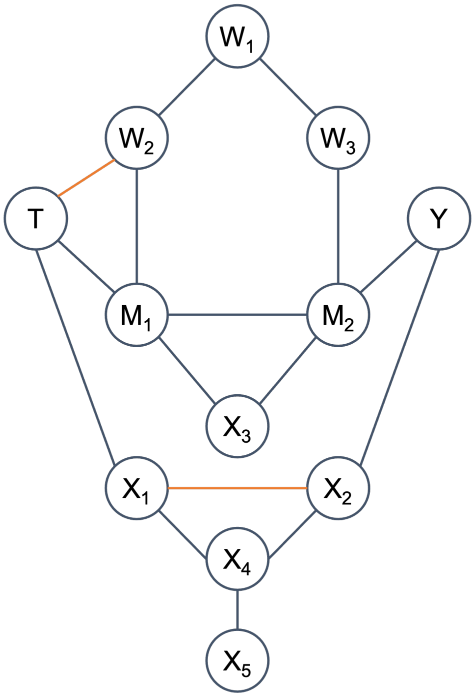


Figure 6: Markov network originated from the conversion of the Bayes network under consideration, where the orange edges correspond to connections between nodes that were not verified in the Bayes network.

Table 1: Observations of the COVID-19 treatment experiment and outcomes, where the number in parenthesis correspond to the total number of people under that category. $T = 0/1$: *no treatment/treatment*; $A = 0/1$: *age below/over 60*; $Y = 0/1$: *death/survival.*

	$\boldsymbol{T = 0}$ **(1500)**		$\boldsymbol{T = 1}$ **(550)**	
	$\boldsymbol{A = 0}$ **(100)**	$\boldsymbol{A = 1}$ **(1400)**	$\boldsymbol{A = 0}$ **(500)**	$\boldsymbol{A = 1}$ **(50)**
$\boldsymbol{Y = 0}$	30	210	100	5
$\boldsymbol{Y = 1}$	70	1190	400	45

2.(a)

Given observation in Table 1 and considering the Bayes theorem, the empirical estimates under consideration are given by:

$$
\begin{aligned}
Pr\{Y=1|T=0,A=1\} &= \frac{1190}{1400} = 0.85\\
Pr\{Y=1|T=1,A=1\} &= \frac{45}{50} = 0.90\\
Pr\{Y=1|T=0,A=0\} &= \frac{70}{100} = 0.70\\
Pr\{Y=1|T=1,A=0\} &= \frac{400}{500} = 0.80\\
Pr\{Y=1|T=0\} &= \frac{1190+70}{1500} = 0.84\\
Pr\{Y=1|T=1\} &= \frac{400+45}{550} = 0.81
\end{aligned}
$$

We can observe that, overall, there appears to be an indication that survival rates are lower when patients are subject to treatment ($Pr\{Y=1|T=0\} = 0.84$ vs $Pr\{Y=1|T=1\} = 0.81$). However, if we look at the individual age subgroups withing each treatment/non-treatment groups, we observe that there is actually a bigger rate of survival within the patients that were subject to treatment, for all age subgroups (e.g. $Pr\{Y=1|T=0,A=1\} = 0.85$ vs $Pr\{Y=1|T=1,A=1\} = 0.90$).
This example perfectly illustrates the Simpson's paradox, in which the age demographics within each experiment group distorts our perception of (supposedly) unbiased data.

2.(b)

Figure 7 shows the causal graph when we intervene on the treatment (`T`) variable.

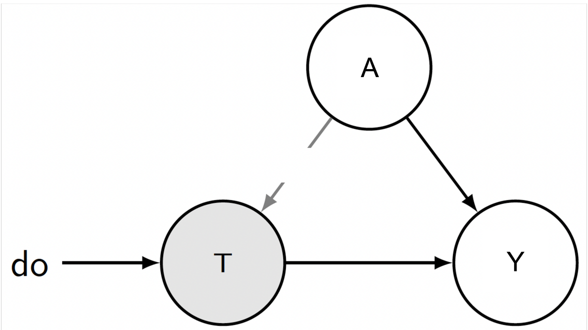


Figure 7: Causal graph when we intervene on the treatment (T) variable, regarding the original causal graph under consideration.

Given that age is a measurable confounder, we can simplify our intervention on treatment as such:

$$
\begin{aligned}
P(Y=1|do(T=t)) &= P(Y=1|do(T=t),A=0)P(A=0|do(T=t))+\\
&\quad + P(Y=1|do(T=t),A=1)P(A=1|do(T=t)) =\\
&= P(Y=1|do(T=t),A=0)P(A=0)+\\
&\quad + P(Y=1|do(T=t),A=1)P(A=1) = \qquad (R3)\\
&= P(Y=1|T=t,A=0)P(A=0)+\\
&\quad + P(Y=1|T=t,A=1)P(A=1) \qquad (R2)
\end{aligned}
$$

where $R2$ corresponds to the second Pearl rule (action/observation exchange: the back-door criterion) and $R3$ corresponds to the second Pearl rule (ignoring actions). Given this, $Pr\{Y=1|T=t\}$ can be computed as:

$$
\begin{aligned}
Pr\{Y=1|do(T=0)\} &= \frac{70}{100}\times\frac{600}{2050}+\frac{1190}{1400}\times\frac{1450}{2050} = 0.81\\
Pr\{Y=1|do(T=1)\} &= \frac{400}{500}\times\frac{600}{2050}+\frac{45}{50}\times\frac{1450}{2050} = 0.87
\end{aligned}
$$

The results of this causal inference with intervention on `T`, allow us to quantify the causal impact of treatment on the outcome, accounting for age as a confounding factor. As such, I would recommend prescribing the treatment.

3.

I was not able to understand the exercise, however, according to the given factorization ($Pr\{X_1 = x_1; X_2 = x_2; X_3 = x_3\} \propto f(x_1;x_2;x_3)g_1(x_1)g_2(x_2)g_3(x_3)$), Figure 8 illustrates the corresponding factor graph.

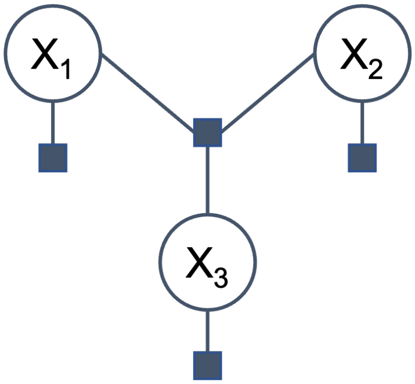


Figure 8: Factor graph that provides the factorization under consideration.