Student Name: João Coelho
Student ID: 86448

Deep Structured Learning
Homework 2

TÉCNICO LISBOA

Question 1

1.1

First, let's show that the Hessian matrix of $E_1(q)$ is positive semi-definite. Starting by computing the gradient:

$$\begin{aligned}\nabla E_1(q) &= \nabla \frac{1}{2} q^T q + \nabla \beta^{-1} \log N + \frac{1}{2} M^2 \\ &= \nabla \frac{1}{2} q^T q \\ &= q \ .\end{aligned} \tag{1}$$

If $\nabla E_1(q) = q$, then $\mathbb{H}(E_1(q)) = \mathrm{I}_D$ (i.e., the DxD identity matrix). Let $v \in \mathbb{R}^D$:

$$v^T \mathrm{I} v = v^T v = ||v||^2 \geq 0 \ . \tag{2}$$

Hence, by definition, the Hessian of $E_1(q)$ is positive semi-definite.

Now, let's show that the Hessian matrix of $-E_2(q)$ is positive semi-definite. Let's start by computing the gradient of the log-sum-exp function with temperature β (lse_β) w.r.t an input vector $v \in \mathbb{R}^D$:

$$\nabla \mathrm{lse}_\beta(v) = \left[\frac{\partial \mathrm{lse}_\beta(v)}{\partial v_1}, ..., \frac{\partial \mathrm{lse}_\beta(v)}{\partial v_D} \right] \ . \tag{3}$$

For a single v_i, the partial derivatives are given by:

$$\begin{aligned}\frac{\partial \mathrm{lse}_\beta(v)}{\partial v_i} &= \beta^{-1} \frac{\partial}{\partial v_i} \log \sum_{j=1}^{D} \exp \beta v_j \\ &= \beta^{-1} \frac{\frac{\partial}{\partial v_i} \sum_{j=1}^{D} \exp \beta v_j}{\sum_{j=1}^{D} \exp \beta v_j} \\ &= \beta^{-1} \frac{\beta \exp(\beta v_i)}{\sum_{j=1}^{D} \exp \beta v_j} \\ &= \frac{\exp(\beta v_i)}{\sum_{j=1}^{D} \exp \beta v_j} \ .\end{aligned} \tag{4}$$

This shows that $\nabla \mathrm{lse}_\beta(v) = \mathrm{softmax}(\beta v)$. Now we can find the gradient of $-E_2(q) = \mathrm{lse}_\beta(Xq)$:

$$\begin{aligned}\nabla - E_2(q) &= \nabla \mathrm{lse}_\beta(Xq) \\ &= \mathrm{softmax}(\beta Xq) \nabla Xq \\ &= X^\top \mathrm{softmax}(\beta Xq) \ .\end{aligned} \tag{5}$$

To compute the Hessian of lse_β, we need to compute the Jacobian of the softmax function w.r.t an input vector $v \in \mathbb{R}^D$:

$$\begin{aligned}
\mathbb{J}_{i,j} &= \frac{\partial \text{softmax}(v)_i}{\partial v_j} \\
&= \frac{\partial}{\partial v_j} \frac{\exp(v_i)}{\sum_{k=1}^D \exp(v_k)} \\
&= \frac{\left(\frac{\partial}{\partial v_j} \exp(v_i)\right) \sum_{k=1}^D \exp(v_k) - \exp(v_i) \left(\frac{\partial}{\partial v_j} \sum_{k=1}^D \exp(v_k)\right)}{\left(\sum_{k=1}^D \exp(v_k)\right)^2}.
\end{aligned}$$

For $i = j$:

$$\begin{aligned}
\mathbb{J}_{i,j} &= \frac{\exp(v_i) \sum_{k=1}^D \exp(v_k) - \exp(v_i) \exp(v_j)}{\left(\sum_{k=1}^D \exp(v_k)\right)^2} \\
&= \frac{\exp(v_i) \left(\sum_{k=1}^D \exp(v_k) - \exp(v_j)\right)}{\left(\sum_{k=1}^D \exp(v_k)\right) \left(\sum_{k=1}^D \exp(v_k)\right)} \\
&= \frac{\exp(v_i)}{\sum_{k=1}^D \exp(v_k)} \frac{\sum_{k=1}^D \exp(v_k) - \exp(v_j)}{\sum_{k=1}^D \exp(v_k)} \\
&= \text{softmax}(v_i) \left(\frac{\sum_{k=1}^D \exp(v_k)}{\sum_{k=1}^D \exp(v_k)} - \frac{\exp(v_j)}{\sum_{k=1}^D \exp(v_k)} \right) \\
&= \text{softmax}(v_i) - \text{softmax}(v_i)\text{softmax}(v_i).
\end{aligned} \tag{6}$$

For $i \neq j$:

$$\begin{aligned}
\mathbb{J}_{i,j} &= \frac{0 - \exp(v_i) \exp(v_j)}{\left(\sum_{k=1}^D \exp(v_k)\right)^2} \\
&= \frac{-\exp(v_i)}{\sum_{k=1}^D \exp(v_k)} \frac{\exp(v_j)}{\sum_{k=1}^D \exp(v_k)} \\
&= -\text{softmax}(v_i)\text{softmax}(v_j).
\end{aligned}$$

We can write both cases as :

$$\mathbb{J} = \text{diag}(\text{softmax}(v)) - \text{softmax}(v)^\top \text{softmax}(v).$$

Let's also show that the above Jacobian matrix is positive semi-definite. Let $x \in \mathbb{R}^D$:

$$\begin{aligned} x^\top \mathbb{J} x &= x^\top [\text{diag}(\text{softmax})(v))]x - x^\top [\text{softmax}(v)^\top \text{softmax}(v)]x \\ &= \sum_{i=1}^{D} x_i^2 \text{softmax}(v)_i - \left(\sum_{i=1}^{D} x_i \text{softmax}(v)_i \right)^2 . \end{aligned} \tag{7}$$

We need to show that:

$$\begin{aligned} & x^\top \mathbb{J} x \geq 0 \\ & \Leftrightarrow \sum_{i=1}^{D} x_i^2 \text{softmax}(v)_i \geq \left(\sum_{i=1}^{D} x_i \text{softmax}(v)_i \right)^2 \\ & \Leftrightarrow \sum_{i=1}^{D} \frac{x_i^2 \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \geq \left(\sum_{i=1}^{D} \frac{x_i \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \left(\sum_{i=1}^{D} \frac{x_i \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \\ & \Leftrightarrow \left(\sum_{i=1}^{D} \frac{x_i^2 \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \left(\sum_{i=1}^{D} \frac{\exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \geq \left(\sum_{i=1}^{D} \frac{x_i \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \left(\sum_{i=1}^{D} \frac{x_i \exp(v_i)}{\sum_{k=1}^{D} \exp(v_k)} \right) \\ & \Leftrightarrow \left(\sum_{i=1}^{D} x_i^2 \exp\left(v_i\right) \right) \left(\sum_{i=1}^{D} \exp\left(v_i\right) \right) \geq \left(\sum_{i=1}^{D} x_i \exp(v_i) \right)^2 \\ & \Leftrightarrow \left(\sum_{i=1}^{D} (x_i \sqrt{\exp(v_i)})^2 \right) \left(\sum_{i=1}^{D} \sqrt{\exp(v_i)}^2 \right) \geq \left(\sum_{i=1}^{D} x_i \sqrt{\exp(v_i)} \sqrt{\exp(v_i)} \right)^2 , \end{aligned} \tag{8}$$

which is true by the Cauchy-Schwarz inequality with $a_i = \sqrt{\exp(v_i)}$ and $b_i = x_i \sqrt{\exp(v_i)}$.

We can now compute the Hessian of $-E_2$:

$$\begin{aligned} \mathbb{H}(-E_2(q)) &= \mathbb{J}(X^\top \text{softmax}(\beta X q)) \\ &= \mathbb{J}(\text{softmax}(\beta X q))X \\ &= \beta X^T (\text{diag}(\text{softmax}(\beta X q)) - \text{softmax}(\beta X q)^\top \text{softmax}(\beta X q))X \end{aligned} \tag{9}$$

Finally, we need to show that the previous Hessian matrix is positive semi-definite. As previously shown, $\text{diag}(\text{softmax}(v)) - \text{softmax}(v)^\top \text{softmax}(v)$ is positive semi-definite. So we need to show that if M is a positive semi-definite matrix, $X^\top M X$ is also positive semi-definite. Then, it also comes directly that $\beta X^\top M X$ is positive semi-definite if $\beta > 0$. Let $v \in \mathbb{R}^D$:

$$\begin{aligned} v^T X^T M X v &= (Xv)^T M (Xv) \\ &= (Xv)^T \sqrt{M}^T \sqrt{M} (Xv) \\ &= \|\sqrt{M}(Xv)\|^2 \geq 0 . \end{aligned}$$

The above proof holds because M is positive semi-definite, hence its square root is well-defined, which allows M to be written as $\sqrt{M}^T \sqrt{M}$.

With this, we proved that both E_1 and $-E_2$ are convex, since both Hessian matrices are positive semi-definite. Hence, E_1 is convex and E_2 is concave. Therefore, E can be written as the sum between a convex function and a concave function.

1.2

From the statement, we have the update rule $q_{t+1} = \text{argmin}_q \; E_1(q) + \tilde{E}_2(q)$, which by definition holds iff $\nabla(E_1(q_{t+1}) + \tilde{E}_2(q_{t+1})) = 0$. Again, per the statement, we can resort to the first-order Taylor approximation to linearize the concave function E_2 around q_t:

$$\begin{aligned} \tilde{E}_2(q) &= E_2(q_t) + (\nabla_q E_2(q_t))(q - q_t) \\ \Leftrightarrow \nabla_q \tilde{E}_2(q) &= \nabla_q(E_2(q_t)) + \nabla_q((\nabla_q E_2(q_t))(q - q_t)) \\ \Leftrightarrow \nabla_q \tilde{E}_2(q) &= \nabla_q((\nabla_q E_2(q_t))q) \\ \Leftrightarrow \nabla_q \tilde{E}_2(q) &= \nabla_q E_2(q_t) \; . \end{aligned} \tag{10}$$

As such, we can rewrite the update rule:

$$\begin{aligned} \nabla(E_1(q_{t+1}) + \tilde{E}_2(q_{t+1})) &= 0 \\ \Leftrightarrow \nabla E_1(q_{t+1}) &= -\nabla \tilde{E}_2(q_{t+1}) \\ \Leftrightarrow \nabla E_1(q_{t+1}) &= -\nabla E_2(q_t) \; . \end{aligned} \tag{11}$$

In the previous exercise, the gradients of both E_1 and E_2 were computed: $\nabla E_1(q) = q$, and $\nabla E_2(q) = X^\top \text{softmax}(\beta X q)$. Hence, for the update $q_t \mapsto q_{t+1}$:

$$\begin{aligned} \nabla E_1(q_{t+1}) &= -\nabla E_2(q_t) \\ q_{t+1} &= X^\top \text{softmax}(\beta X q_t) \; . \end{aligned} \tag{12}$$

1.3

For the first Hopfield update for a single q_0, considering $\beta = \frac{1}{\sqrt{D}}$, we have:

$$q_1 = X^\top \text{softmax}\left(\frac{X q_0}{\sqrt{D}}\right) \; . \tag{13}$$

For the Transformer, the self-attention computation can be written as follows:

$$\text{Attention}(Q, K, V) = V^\top \text{softmax}\left(\frac{Q K^\top}{\sqrt{D}}\right) \; . \tag{14}$$

The Q, K, and V matrices are obtained by multiplying an input matrix X by W_Q, W_K, and W_V, respectively. In this case, assuming $W_Q = W_K = W_V = I$, we have that $V{=}K{=}X$. So, for a single state pattern q (i.e., a row from Q), we can write:

$$\text{Attention}(Q, K, V) = X^\top \text{softmax}\left(\frac{X q}{\sqrt{D}}\right) \; . \tag{15}$$

Hence, in this specific scenario, the computations are equivalent.

Question 2

2.1a

For the source vocabulary, there are 51 unique tokens, including four special tokens ([unk], [pad], [bos], [eos]). For the target vocabulary, there are 43 unique tokens, including four special tokens ([unk], [pad], [bos], [eos]).

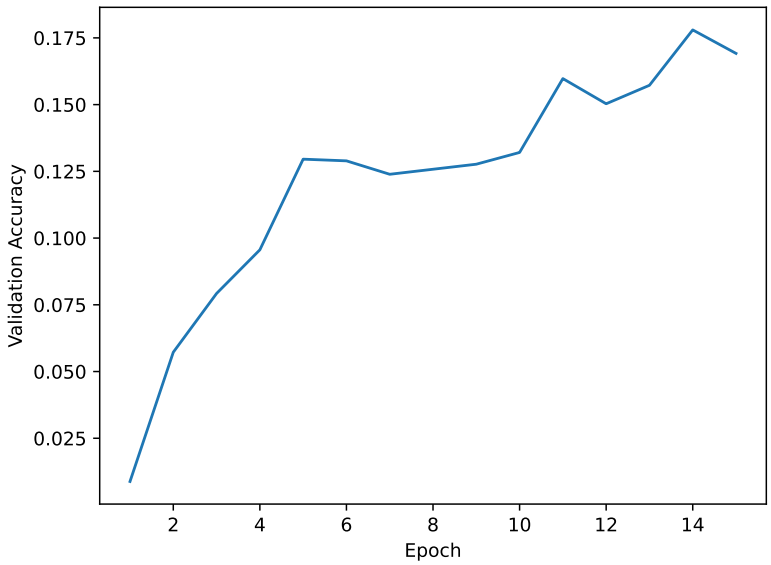
2.1b

Implementation details: Gradient clipping was used to avoid gradient explosion. L2 and dropout regularization were added to avoid overfitting. Teacher forcing was implemented. The cross-entropy loss function ignores padding tokens. The best model in validation is loaded for test.

A model was trained with the following hyperparameters:

- Epochs: 15
- Batch size: 32
- Encoder layers: 1
- Decoder layers: 1
- Hidden size: 128
- Learning rate: 3e-3
- L2 penalty: 1.5e-4
- Dropout probability: 0.15
- Teacher-forcing probability: 0.50

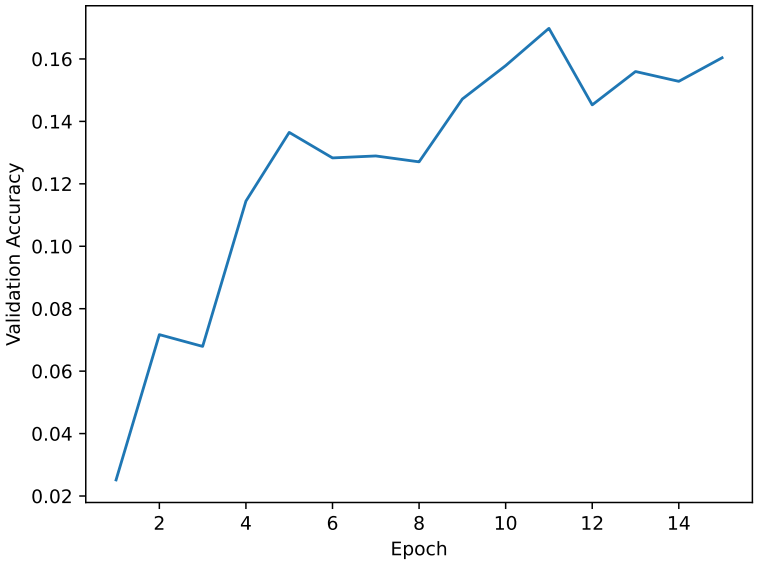
The following plot shows validation word-accuracy per epoch:



The model achieved a word-accuracy on the test set of 0,1670.

2.1c

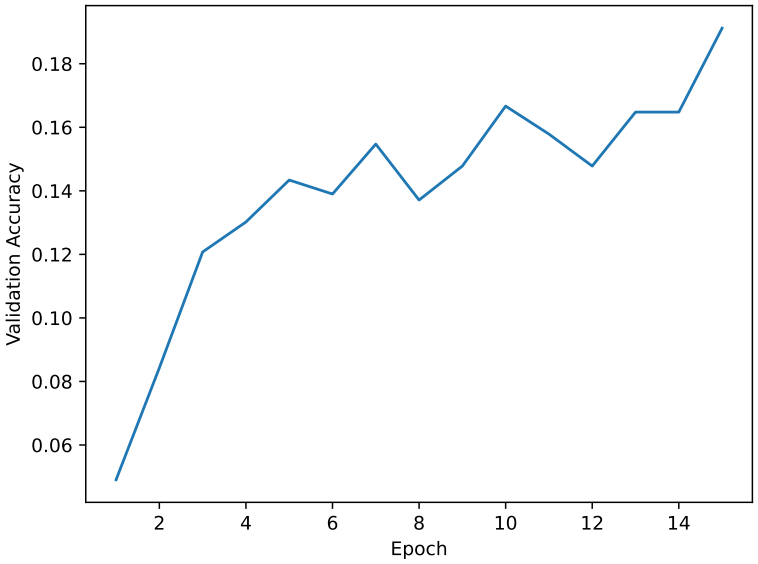
A model was trained with the exact same setup of the previous exercise, the only difference being the reverted input string. The following plot shows validation word-accuracy per epoch:



The model achieved a word-accuracy on the test set of 0,1754.

2.1d

The required changes were added. An attention mask was implemented so that padding tokens are unattended. The model was trained with the same hyperparameters as the previous exercises (the input string was not reverted). The following plot shows validation word-accuracy per epoch:



The model achieved a word-accuracy on the test set of 0,1935.

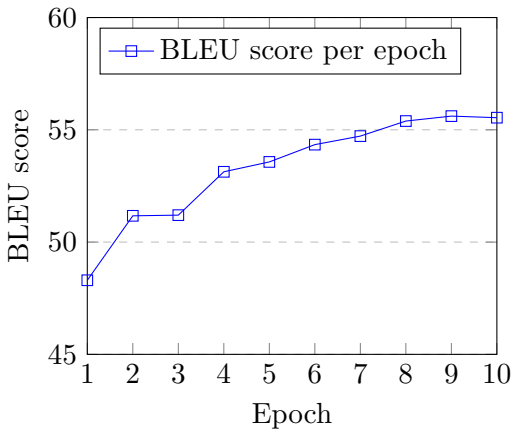
Question 3

3.1a

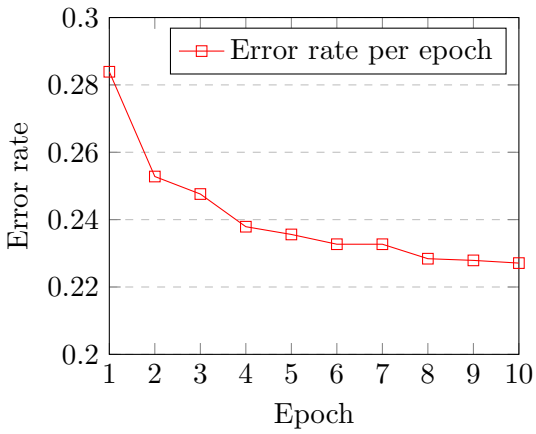
A model was implemented as described. The configuration file with the required parameters as per the statement is presented here:

```
{
  "training_args.output_dir": "models/bert-base-uncased-transileration-en-ar-v2",
  "training_args.learning_rate": 2e-5,
  "training_args.per_device_train_batch_size": 4,
  "training_args.num_train_epochs": 10,
  "training_args.metric_for_best_model": "error_rate",
  "training_args.greater_is_better": False,
  "early_stopping.early_stopping_patience": 3,
}
```

The remaining parameters were changed in the code if necessary, or left as default. Even with the early stopping on, and considering error rate as the main metric, the model trained for the whole 10 epochs. The graphs below show the evolution of the metrics, computed on the validation split, during the 10 epochs:



(a) BLEU score per epoch over 10 epochs.



(b) Error rate per epoch over 10 epochs.

The best model in terms of error rate was loaded, and achieved a BLEU score of 56,23 and an error rate of 0,2298 on the test split.

3.1b

Yes, through the positional encoding step, which consists in summing a vector that captures positional information to the input embeddings. The computation of this vector was originally proposed such that each position corresponds to a sinusoid, alternating between sine and co-sine functions. The authors defined the function in such a way that (i) it outputs a unique encoding for each word position in a sentence (which can vary in length between examples), (ii) the distance between two positions is consistent across sentences with different lengths, (iii) is deterministic, (iv) and can generalize to sequences larger than those seen during training (although this ability is then limited by the self-attention's quadratic complexity). This way, even without RNN's inherent ability of capturing sequential information, the Transformer is able to notice the order of tokens and can capture dependencies from relative positions.

3.2a

A generation script was implemented, using `t5-small` and `t5-base` as the generation models. Since the task of translating from English to French is one of the tasks T5 was pre-trained on, the models' `config.json` contains generation parameters and a specific prefix, which were used:

```
"translation_en_to_fr": {
      "early_stopping": true,
      "max_length": 300,
      "num_beams": 4,
      "prefix": "translate English to French: "
}
```

Regarding the evaluation on the test split of the dataset, `t5-small` achieved a BLEU score of 36,18 and `t5-base` achieved 38,48. The first 10 generations from each model and the respective label are printed in Table 1, in the last page of this document.
The increased number of parameters in `t5-base`, when compared to `t5-small`, seem to have a positive impact in terms of BLEU score. Regarding the translations, even though I am not fluent in French, both models seem to produce strong results. In some cases, both models show preference for the same type of language that, while correct (i think), differs from the label, e.g., for the 6^{th} example, where the label is *Et ça a été un grand succès*, both models used the different verbal form *c'était*; for the 5^{th} example, the label reads *J'ai trouvé que c'était une idée intéressante* and both models opted for *J'ai donc pensé qu'il s'agissait d'une ideé intéressante*. There are also cases where both models have a different choice of words for the same concept, e.g., again in the 6^{th} example, `t5-small` generated *grand succès* while `t5-base` generated *énorme succès*. Summarily, while there may be errors in the models' translations that I can't identify due to not knowing the language, the above-mentioned divergences show a problem for Machine Translation that token-overlapping metrics such as BLEU can't account for (and its worse in cases like this when there is a single label) - there are many correct ways of saying the same thing.

Table 1: First 10 generations for t5-small and t5-base, along with the label.

Label	`t5-small`	`t5-base`
Il y a plusieurs années, ici à Ted, Peter Skillman a présenté une épreuve de conception appelée l'épreuve du marshmallow.	Il y a plusieurs années, à TED, Peter Skillman a présenté un défi de conception appelé le défi des marshmallows.	Il y a plusieurs années, ici à TED, Peter Skillman a présenté un défi de design appelé le défi des marshmallows.
Et l'idée est plutôt simple. Des équipes de quatre personnes doivent bâtir la plus haute structure tenant debout avec 20 spaghettis, un mètre de ruban collant, un mètre de ficelle, et un marshmallow.	Et l'idée est assez simple : les équipes de quatre doivent construire la structure la plus haute à pied à partir de 20 bâtonnets de spaghetti, d'une couronne de ruban, d'une couronne de corde et d'un marshmallow.	Et l'idée est assez simple : les équipes de quatre doivent construire la plus haute structure autonome à partir de 20 bâtons de spaghetti, d'un verge de ruban, d'un verge de corde et d'un marshmallow.
Le marshmallow doit être placé au sommet.	Le marshmallow doit être en haut.	Le marshmallow doit être sur le dessus.
Bien que cela semble vraiment simple, c'est en fait plutôt difficile, parce que ça oblige les gens à collaborer rapidement.	Et, même si c'est vraiment simple, c'est en fait assez difficile parce qu'elle force les gens à collaborer très rapidement.	Et, bien qu'il semble vraiment simple, il est en fait assez difficile parce qu'il oblige les gens à collaborer très rapidement.
J'ai trouvé que c'était une idée intéressante, alors je l'ai insérée dans un atelier de conception.	J'ai donc pensé qu'il s'agissait d'une idée intéressante et j'ai intégré cette idée dans un atelier de conception.	J'ai donc pensé qu'il s'agissait d'une idée intéressante et j'ai incorporé cette idée dans un atelier de conception.
Et ça a été un grand succès.	Et c'était un grand succès.	Et c'était un énorme succès.
Depuis, j'ai dirigé environ 70 ateliers de conception dans le monde entier, avec des étudiants, des designers et des architectes, et même les Directeurs Techniques d'entreprises du Fortune 50, et il y a quelque chose dans cet exercice qui révèle de profonds enseignements sur la nature de la collaboration, et j'aimerais en partager certains avec vous.	Depuis lors, j'ai organisé environ 70 ateliers de conception dans le monde avec des étudiants, des concepteurs et des architectes, même les CTO de la Fortune 50, et il y a quelque chose à propos de cet exercice qui révèle des leçons très profondes sur la nature de la collaboration, et je voudrais partager quelques-uns avec vous.	Depuis lors, j'ai organisé environ 70 ateliers de conception dans le monde entier avec des étudiants, des designers et des architectes, et même des CTO des Fortune 50, et il y a quelque chose à cet exercice qui révèle des leçons très profondes sur la nature de la collaboration, et j'aimerais partager certaines d'entre elles avec vous.
Bon, normalement la plupart des gens commencent par prendre leurs marques par rapport à la tâche.	Ainsi, normalement, la plupart des gens commencent par s'orienter vers la tâche.	Donc, normalement, la plupart des gens commencent par s'orienter vers la tâche.
Ils en parlent, ils cherchent à quoi ça va ressembler, ils manoeuvrent pour le pouvoir,	Ils parlent de cela, ils établissent ce qu'il va ressembler, ils jocquent pour le pouvoir.	Ils en parlent, ils en déterminent l'aspect, ils se battent pour le pouvoir.
et puis ils passent un peu de temps à planifier, à organiser. Ils font des croquis, et disposent les spaghettis.	Ils passent ensuite un peu de temps à planifier, à organiser, à dessiner et à mettre en place des spaghettis.	Ils passent ensuite du temps à planifier, à organiser, à dessiner et à préparer des spaghettis.