Mastafa FOUFA **Advanced NLP**

Lab 2

Assignment 2. NLP across languages, a multilingual setting

Note. Here are some things to keep in mind as you plan your time for this assignment.

• There are some cool math questions.

• In this assignment, you will be given less guidance or scaffolding in how to write the code.

• I am here to help!

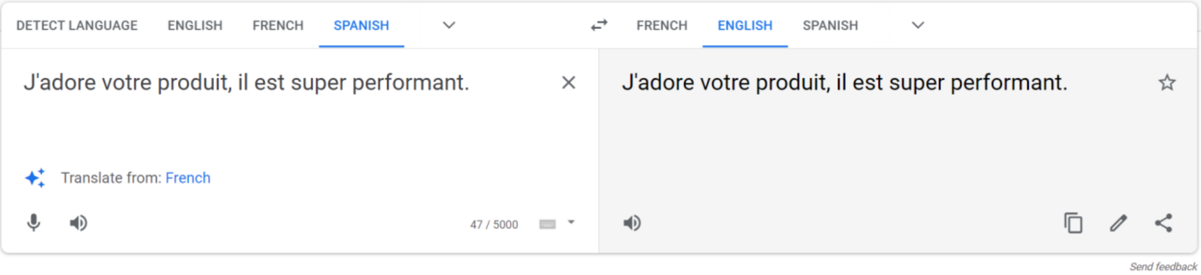
This assignment is focusing on a few key areas:

- Language detection – how to detect a language given a sentence.
- Multilingual embeddings produced in an offline setting – how to align two semantic spaces representing two different languages.
- Transformer self-attention – what are the mathematical properties of transformer models? The intuition is essential to understand the mechanics of complex models.
- Neural Machine Translation – understanding how a real-world model works.

I. Language Detection (24 points) – Guided coding

Understanding the main language of a sentence is essential when you are developing a product across the world and gathering satisfaction feedback. The following section helps you understand some of the technicalities in this space. Keep in mind there is a lot to do in the industry in this area.

0. Try out a translation of a French sentence in Google Translate (or Bing Translate) to English. What happens if you select the wrong source language as follows? Explain in a few sentences what is happening in the backend. (1 point)



J'adore votre produit, il est super performant.

J'adore votre produit, il est super performant.

Figure 1. Wrong source language selected on Google translate. The translation to English does not work because of that.

As you can see from Figure 1 above, Google detects that the language of the **source** sentence is French. In the following, we will understand how we can do that ourselves.

❖ **Download the Colab for Assignment 2 and follow along. Please report your answers in a word document.**

1. Describe the language distribution of the dataset. What is the distribution of languages? (1 point)
2. Do the appropriate pre-processing to maximise the accuracy of language detection. What is your strategy? (1 point)
3. What would be the problem if your dataset was unbalanced? (1 point)
4. What techniques could you use to solve that? (1 point)
5. Train a model of your choice and describe the accuracy across languages. Use an 80%, 20% train-test split. Performance is not key but explain thoroughly the process and the metric(s) you are tracking. (4 points)
6. Train a fasttext model on Tatoeba parallel corpus and check that performance is good. (3 points)
7. Test your fasttext model on the same dataset as in question 1-5. Compare with your custom model (make sure you use the exact same data for testing). How can you explain the difference in performance between the two models? (3 points)
8. Compute your performance metrics yourself and compare with sklearn. (1 point)
9. How could you improve the fasttext model performance from the previous question? Explain in a few sentences. (2 points)
10. Which method would you use for language detection and why? (1 point)
11. Given a sentence with N_1 tokens in English and N_2 token in French, what would be your strategy to assign a language to such sentence? (2 points)
12. Would a multilingual architecture be robust to multiple languages in a single sentence? Elaborate your answer accordingly. (4 points)

II. Rotate two semantic spaces (23 points) – Not guided coding

1. Explain in a few sentences how MUSE[1] is doing the alignment of the semantic spaces in the supervised way. (1 point)
2. What is the limit of doing that alignment based on the approach taken in the supervised way? (2 points)
3. How can we align two semantic spaces in a domain specific field, e.g., in a tech company? (2 points)
4. Align the French space and the English space together, with the method of your choice. (5 points)
5. Visualize the output on a few words of your choice. Comment on the performance of the alignment based on the output. (2 points)
6. How can you find the translation of a word with this approach? Explain your method and the distance metric you choose in a few sentences. (2 points)
7. Apply your approach and comment on the performance of the translation. (3 points)
8. What is the limit of aligning two spaces at a sentence level? What do you suggest to improve the alignment, at a sentence level? (4 points)
9. Someone, in your company, asked you to do sentiment analysis on their dataset. Given a set of sentences $\{s_1, \dots, s_N\}$, where s_i can be written in any language, what architecture would you use to have a vector representation of s_i? Motivate in 2-3 bullet points. (2 points)
    1. …
    2. …
    3. …
10. How would you do sentiment analysis across multiple languages in a domain specific context? Justify your approach step by step. (5 points)

III. Attention Exploration (22 points)

Multi-headed self-attention is the core modelling component of Transformers. In this question, we'll get some practice working with the self-attention equations and motivate why multi-headed self-attention can be preferable to single-headed self-attention.

[1] https://github.com/facebookresearch/MUSE

a. (2 points) **Copying in attention**: Recall that attention can be viewed as an operation on a query $q \in R^d$, a set of value vectors $\{v_1, …., v_n\}, v_i \in R^d$, and a set of key vectors $\{k_1, … , k_n\},\ such\ as\ k_i \in R^d$ specified as follows:

$$c = \sum_{i=1}^{n} \alpha_i v_i$$

Where each α_i is described as follows:

$$\alpha_i = \frac{\exp(k_i^T q)}{\sum_{j=1}^{n} \exp\,(k_j^T q)}$$

The α_i are frequently called the "attention weights", and the outputs $c \in R^d$ is a correspondingly weighted average over the value vectors.

We'll first show that it's particularly simple for attention to "copy" a value vector to the output c.

Describe (in one sentence) what properties of the inputs to the attention operation would result in the output c being approximately equal to v_j for some $j \in \{1, … , n\}$. Specifically, what must be true about the query q, the values $\{v_1, …., v_n\}$ and/or the keys $\{k_1, … , k_n\}$?

b. (4 points) **An average of two**: Consider a set of key vectors $\{k_1, . . . , k_n\}$, where all key vectors are perpendicular, that is $k_i \perp k_j$ for all $i \neq j$. Let ‖ k_i ‖ = 1 for all i. Let $\{v_1, …., v_n\}$ be a set of arbitrary value vectors. Let $v_a, v_b \in \{v_1, …., v_n\}$ be two of the value vectors. Give an expression for a query vector q such that the output c is approximately equal to the average of v_a and v_b, that is, $\frac{1}{2}\ (v_a\ +\ v_b)$.

Note that you can reference the corresponding key vector of v_a and v_b as k_a and k_b .

Hint: while the softmax function will never exactly average the two vectors, you can get close by using a large scalar multiple in the expression.

c. (5 points) **Drawbacks of single-headed attention**: In the previous part, we saw how it was possible for a single-headed attention to focus equally on two values. The same concept could easily be extended to any subset of values. In this question we'll see why it's not a practical solution. Consider a set of key vectors $\{k_1, . . . , k_n\}$ that are now randomly sampled, $k_i \sim N\ (\mu_i\ , \Sigma_i)$, where the means μ_i are known to you, but the covariances Σ_i are unknown. Further, assume that the means μ_i are all perpendicular; $\mu_i^T \mu_j = 0$ if $i \neq j$, and unit norm, ‖ μ_i ‖ $=$ 1.

i. (2 points) Assume that the covariance matrices are $\Sigma_i = \alpha I$, for vanishingly small α. Design a query q in terms of the μi such that as before, $c \approx \frac{1}{2}(v_a + v_b)$, and provide a brief argument as to why it works.

ii. (3 points) Though single-headed attention is resistant to small perturbations in the keys, some types of larger perturbations may pose a bigger issue. Specifically, in some cases, one key vector ka may be larger or smaller in norm than the others, while still pointing in the same direction as πa. As an example, let us consider a covariance for item a as $\Sigma_a = \alpha * I + \frac{1}{2}(\mu_a \mu_a^T)$ for vanishingly small α (as shown in figure 2). Further, let $\Sigma_i = \alpha I \; for \; all \; i \neq a$.

When you sample $\{k_1, ..., k_n\}$ multiple times, and use the q vector that you defined in part i., what qualitatively do you expect the vector c will look like for different samples?

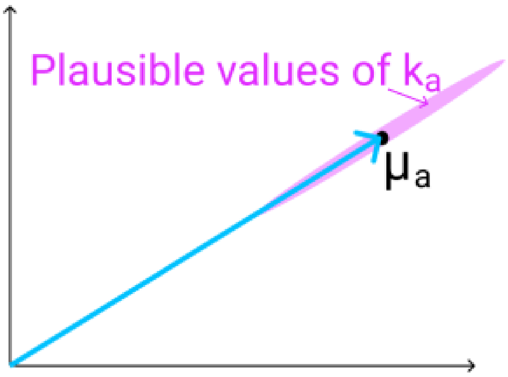


Figure 2. The vector μa (shown here in 2D as an example), with the range of possible values of ka shown in red. As mentioned previously, ka points in roughly the same direction as μa, but may have larger or smaller magnitude.

d. (3 points) **Benefits of multi-headed attention**: Now we'll see some of the power of multi-headed attention. We'll consider a simple version of multi-headed attention which is identical to single-headed self-attention as we've presented it in this homework, except two query vectors ($q_1 \; and \; q_2$) are defined, which leads to a pair of vectors ($c_1 \; and \; c_2$), each the output of single-headed attention given its respective query vector. The final output of the multi-headed attention is their average, $\frac{1}{2}\,(c_1 + c_2)$. As in question 1 (c), consider a set of key vectors $\{k_1, ..., k_n\}$ that are randomly sampled, $k_i \sim N\,(\mu_i\,, \Sigma_i)$, where the means μ_i are known to you, but the covariances Σ_i are unknown. Also as before, assume that the means μ_i are mutually orthogonal; $\mu_i^T \mu_j = 0$ if $i \neq j$, and unit norm, $\parallel \mu_i \parallel = 1$.

i. (1 point) Assume that the covariance matrices are $\Sigma_i = \alpha I$, for vanishingly small α. Design q_1 and q_2 such that c is approximately equal to $\frac{1}{2}(v_a + v_b)$.

ii. (2 points) Assume that the covariance matrices are $\Sigma_a = \alpha * I + \frac{1}{2}(\mu_a \mu_a^T)$ for vanishingly small α, and $\Sigma_i = \alpha I$ for all $i \neq a$. Take the query vectors q_1 and q_2 that you designed in part i. What, qualitatively, do you expect the output c to look like across different samples of the key vectors? Please briefly explain why. You can ignore cases in which $q_i^T k_a < 0$.

IV. Neural Machine Translation (32 points)

Cherokee is a highly endangered Native American language spoken by the Cherokee people. The Cherokee culture is deeply embedded in its language. However, there are approximately only 2,000 fluent first language Cherokee speakers remaining in the world, and the number is declining every year.

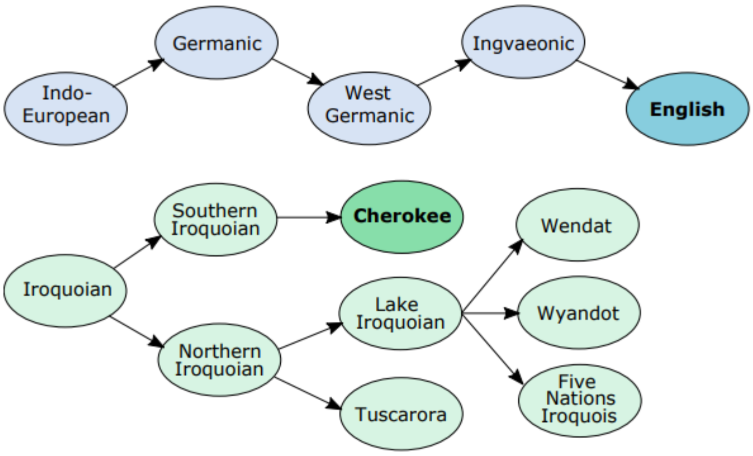


Figure 1: Language family trees.

Figure 3. Language family trees. The translation between Cherokee and English is not easy because the two languages are genealogically disparate. As shown in Figure 1, Cherokee is the sole member of the southern branch of the Iroquoian language family and is unintelligible to other Iroquoian languages, while English is from the West Germanic branch of the Indo-European language family. (TODO: link to 2010.04791.pdf (arxiv.org))

In this section, we assume we built a NMT model at a subword level. That means, given a sentence in the source language, we looked up subword components from an embedding matrix. There are alternative methods such as building a NMT model at a word-level by looking up whole words from the embeddings matrix.

a. (2 points) Why might it be important to model our Cherokee-to-English NMT problem at the subword-level vs. the whole word-level? (Hint: Cherokee is a polysynthetic language.)

b. (2 points) Character-level and subword embeddings are often smaller than whole word embeddings. In 1-2 sentences, explain one reason why this might be the case.

c. (2 points) One challenge of training successful NMT models is lack of language data, particularly for resource-scarce languages like Cherokee. One way of addressing this challenge is with multilingual training, where we train our NMT on multiple languages (including Cherokee). You can read more about multilingual training here. How does multilingual training help in improving NMT performance with low-resource languages?

d. (6 points) Here we present three examples of errors we found in the outputs of our NMT model (which is the same as the one you just trained). The errors are underlined in the NMT translation sentence. For each example of a source sentence, reference (i.e., 'gold') English translation, and NMT (i.e., 'model') English translation, please:

1. Provide possible reason(s) why the model may have made the error (either due to a specific linguistic construct or a specific model limitation).

2. Describe one possible way we might alter the NMT system to fix the observed error. There are more than one possible fixes for an error. For example, it could be tweaking the size of the hidden layers or changing the attention mechanism.

Below are the translations that you should analyse as described above. Only analyse the underlined error in each sentence. Rest assured that you don't need to know Cherokee to answer these questions. You just need to know English! If, however, you would like additional colour on the source sentences, feel free to use resources like https://www.cherokeedictionary.net/ to look up words.

i. (2 points)

Source Translation: *Yona utsesdo ustiyegv anitsilvsgi digvtanv uwoduisdei.*

Reference Translation: *Fern had a crown of daisies in her hair.*

NMT Translation: *Fern had her hair with her hair.*

ii. (2 points)

Source Sentence: *Ulihelisdi nigalisda.*

Reference Translation: *She is very excited.*

NMT Translation: *It's joy.*

iii. (2 points)

Source Sentence: *Tsesdi hana yitsadawoesdi usdi atsadi!*

Reference Translation: *Don't swim there, Littlefish!*

NMT Translation: *Don't know how a small fish!*

(f) (14 points) BLEU score is the most commonly used automatic evaluation metric for NMT systems. It is usually calculated across the entire test set, but here we will consider BLEU defined for a single example.2 Suppose we have a source sentence s, a set of k reference translations r1, . . . , rk, and a candidate translation c. To compute the BLEU score of c, we first compute the modified n-gram precision pn of c, for each of n = 1, 2, 3, 4, where n is the n in n-gram:

$$p_n = \frac{\sum_{\text{ngram} \in \mathbf{c}} \min\left(\max_{i=1,\ldots,k} \text{Count}_{\mathbf{r}_i}(\text{ngram}),\ \text{Count}_{\mathbf{c}}(\text{ngram}) \right)}{\sum_{\text{ngram} \in \mathbf{c}} \text{Count}_{\mathbf{c}}(\text{ngram})}$$

Here, for each of the n-grams that appear in the candidate translation c, we count the maximum number of times it appears in any one reference translation, capped by the number of times it appears in c (this is the numerator). We divide this by the number of n-grams in c (denominator).

Next, we compute the *brevity penalty* BP. Let len(c) be the length of c and let len(r) be the length of the reference translation that is closest to len(c) (in the case of two equally-close reference translation lengths, choose len(r) as the shorter one).

$$BP = \begin{cases} 1 & \text{if } len(c) \geq len(r) \\ \exp\left(1 - \frac{len(r)}{len(c)}\right) & \text{otherwise} \end{cases}$$

Lastly, the BLEU score for candidate c with respect to r1, . . . , rk is:

$$BLEU = BP \times \exp\left(\sum_{n=1}^{4} \lambda_n \log p_n \right)$$

where λ1, λ2, λ3, λ4 are weights that sum to 1. The log here is natural log.

i. (5 points) Please consider this example from Spanish:

Source Sentence s: el amor todo lo puede

Reference Translation r1: love can always find a way

Reference Translation r2: love makes anything possible

NMT Translation c1: the love can always do

NMT Translation c2: love can make anything possible

Please compute the BLEU scores for c1 and c2. Let $\lambda_i = 0.5$ for $i \in \{1, 2\}$ and $\lambda_i = 0$ for $i \in \{3, 4\}$ (this means we ignore 3-grams and 4-grams, i.e., don't compute p3 or p4). When computing BLEU scores, show your working (i.e., show your computed values for p1, p2, len(c), len(r) and BP). Note that the BLEU scores can be expressed between 0 and 1 or between 0 and 100. The code is using the 0 to 100 scale while in this question we are using the 0 to 1 scale.

Which of the two NMT translations is considered the better translation according to the BLEU Score? Do you agree that it is the better translation?

ii. (5 points) Our hard drive was corrupted, and we lost Reference Translation r2. Please recompute BLEU scores for c1 and c2, this time with respect to r1 only. Which of the two NMT translations now receives the higher BLEU score? Do you agree that it is the better translation?

iii. (2 points) Due to data availability, NMT systems are often evaluated with respect to only a single reference translation. Please explain (in a few sentences) why this may be problematic.

iv. (2 points) List two advantages and two disadvantages of BLEU, compared to human evaluation, as an evaluation metric for Machine Translation.