

Project 1

Follow up

LINFO2263 – Computational Linguistics

General comments

If not already started, do it now!

Project takes time

Time needed for each task increases as you advance into the project!

Only around $\frac{1}{4}$ of students started already!

General comments

- Refer to the slides for theoretical notions
- No plagiarism
 - You have to submit the code you designed yourself to get your answers
- Do not forget to cite your public sources, if any
- You may use generative tools, but indicate it in the code as any other public source

General comments

100% on Inginius
before final deadline



100%
in the end

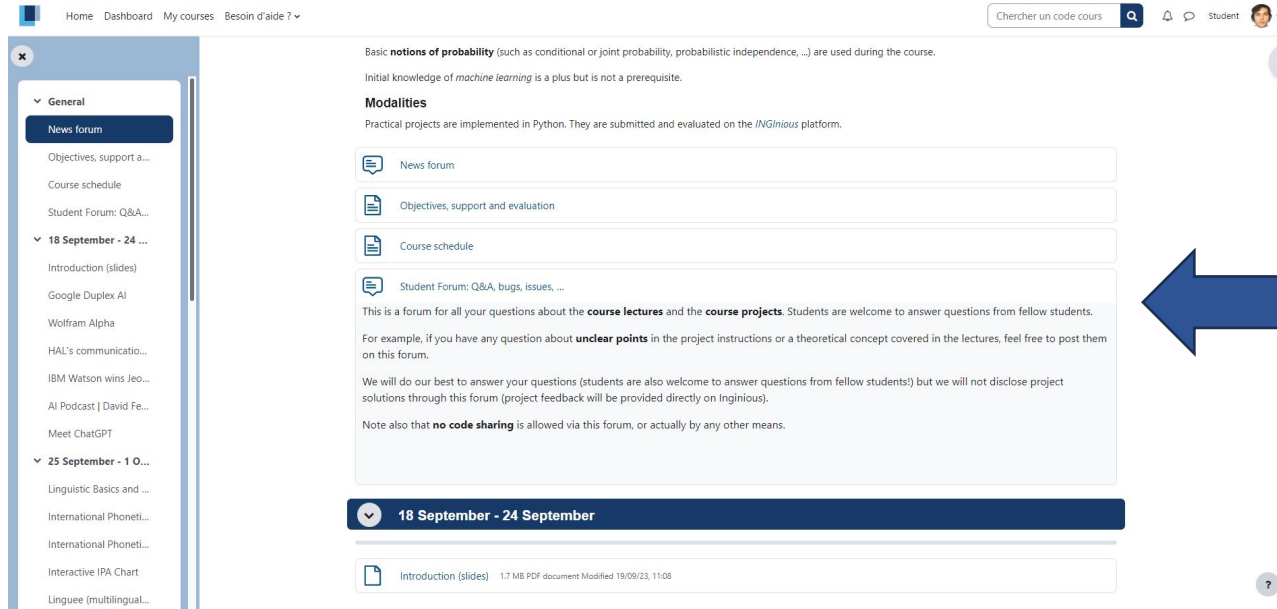
100% before deadline = your answers have the correct format

Grading is run after the deadline

Only the last submission is evaluated

General comments

Get help, help the others: use the Student Forum!



The screenshot shows the 'Student Forum' interface. On the left is a sidebar with a menu. The main content area on the right contains a header with course information, a 'Modalities' section, and a list of forum topics. A large blue arrow points from the right towards the 'Student Forum: Q&A, bugs, issues, ...' topic.

Home Dashboard My courses Besoin d'aide ?

Chercher un code cours

News forum

Objectives, support a...

Course schedule

Student Forum: Q&A...

18 September - 24 ...

Introduction (slides)

Google Duplex AI

Wolfram Alpha

HAL's communicatio...

IBM Watson wins Jee...

AI Podcast | David Fe...

Meet ChatGPT

25 September - 1 O...

Linguistic Basics and ...

International Phonet...

International Phonet...

Interactive IPA Chart

Linguee (multilingual...

Basic **notions of probability** (such as conditional or joint probability, probabilistic independence, ...) are used during the course.

Initial knowledge of *machine learning* is a plus but is not a prerequisite.

Modalities

Practical projects are implemented in Python. They are submitted and evaluated on the *INGenius* platform.

News forum

Objectives, support and evaluation

Course schedule

Student Forum: Q&A, bugs, issues, ...

This is a forum for all your questions about the **course lectures** and the **course projects**. Students are welcome to answer questions from fellow students.

For example, if you have any question about **unclear points** in the project instructions or a theoretical concept covered in the lectures, feel free to post them on this forum.

We will do our best to answer your questions (students are also welcome to answer questions from fellow students!) but we will not disclose project solutions through this forum (project feedback will be provided directly on Inginious).

Note also that **no code sharing** is allowed via this forum, or actually by any other means.

18 September - 24 September

Introduction (slides) 1.7 MB PDF document Modified 19/09/23, 11:08

No code sharing !!

General comments

TASK



1.1



1.2

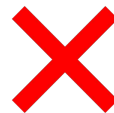


1.3



Use only for preprocessing (vocabulary, ...)

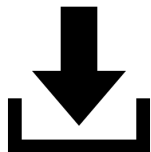
1.4



Use only for preprocessing (vocabulary, ...)

Question 1.0

Download the corpus



!! Preprocess the corpus !!

```
to_skip = ['(', ')', '[', ']', '{', '}', ':', ';', '=', '-', '/', '\\', "'", '"']
```

Question 1.0

Examples

```
to_skip = ['(', ')', '[', ']', '{', '}', ':', ';', '=', '-', '/', '\\', "'", '"']
```

```
tokens = ["hello:", "world!", ";", "hel-lo", "\\everyone"]
```


Question 1.0

Examples

```
to_skip = ['(', ')', '[', ']', '{', '}', ':', ';', '=', '-', '/', '\\', "'", '"']
```

```
tokens = ["hello:", "world!", ";", "hel-lo", "\\everyone"]
```

```
tokens = ["hello", ]
```

Question 1.0

Examples

```
to_skip = ['(', ')', '[', ']', '{', '}', ':', ';', '=', '-', '/', '\\', "'", '"']
```

```
tokens = ["hello:", "world!", ";", "hel-lo", "\\everyone"]
```

```
tokens = ["hello", "world!", ]
```

Question 1.0

Examples

```
to_skip = ['(', ')', '[', ']', '{', '}', ':', ';', '=', '-', '/', '\\', "'", '"']
```

```
tokens = ["hello:", "world!", ";", "hel-lo", "\\everyone"]
```

```
tokens = ["hello", "world!", "hello", "everyone"]
```

Question 1.1

- Check the documentation
- Remark: not all tokens are words
- OOV = frequency of the <UNK> token in the whole corpus (as a %)

Question 1.2

- What would be the padded version of the sentence below?

May the Force be with you

Question 1.2

- What would be the padded version of the sentence below?

May the Force be with you

Answer:

<s> May the Force be with you </s>

- What would be the padded bigram of this sentence?

Question 1.2

- What would be the padded version of the sentence below?

May the Force be with you

Answer:

<s> May the Force be with you </s>

- What would be the padded bigram of this sentence?

Answer:

[("<s>", "May"), ("May", "the"), ("the", "Force"), ("Force", "be"), ("be", "with"),
("with", "you"), ("you", "</s>")]



Question 1.3

- Avoid numerical underflows, **use logarithms**
- 2 classes: HQ (High Quality) and LQ (Low Quality) and 2 models (one for each class)
- Consider the full question directly, do not split by sentence (no need for padding in this case!)

Question 1.4



- 3 sub-tasks, **independent** from each other
 - Binary Naive Bayes
 - Negative Tokens Preprocessing
 - Smoothed Bigrams Model
- Report your accuracy and your code for each of them
- Refer to the slides for theoretical notions



Question 1.4 – Binary Naive Bayes

Word occurrences clipped to 1

Example:

With great power comes great responsibility.

What is the total count of “power”? And “great”?



Question 1.4 – Binary Naive Bayes

Word occurrences clipped to 1

Example:

With great power comes great responsibility.

What is the total count of “power”? And “great”?

↳ 1 (clipped)

And if we **add** the following document in our corpus?

We will watch your career with great interest.



Question 1.4 – Binary Naive Bayes

Word occurrences clipped to 1

Example:

With great power comes great responsibility.

What is the total count of “power”? And “great”?

↳ 1 (clipped)

And if we **add** the following document in our corpus?

We will watch your career with great interest.

↳ power: 1, great: 2



Question 1.4 – Negative tokens

Add « _NOT » after each token between a negative token and a punctuation sign

Consider ONLY ***not***, ***no*** and ***never*** as negative tokens

Consider ONLY these punctuation signs . , ? !



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.

↳ ..., we do not need_NOT roads_NOT.



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.

↳ ..., we do not need_NOT roads_NOT.

Where we are going, we don't need roads.



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.

↳ ..., we do not need_NOT roads_NOT.

Where we are going, we don't need roads.

↳ NO CHANGE! (NB: this is an approximation)



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.

↳ ..., we do not need_NOT roads_NOT.

Where we are going, we don't need roads.

↳ NO CHANGE! (NB: this is an approximation)

Where we are going, we do not need roads



Question 1.4 – Negative tokens

Some examples to be clear...

Where we are going, we do not need roads.

↳ ..., we do not need_NOT roads_NOT.

Where we are going, we don't need roads.

↳ NO CHANGE! (NB: this is an approximation)

Where we are going, we do not need roads

↳ ..., we do not need_NOT roads_NOT

Notice that the .
is missing here

Question 1.4 – Negative tokens



Some examples to be clear...

Do or do not there is no try.



Question 1.4 – Negative tokens

Some examples to be clear...

Do or do not there is no try.

↳ Do or do not there_NOT is_NOT no_NOT try_NOT.



Question 1.4 – Negative tokens



This modifies the original corpus!

What about the vocabulary?

Is it the same as before?

Question 1.4 – Smoothed Bigrams Model



During preprocessing...

How to treat the words not in the vocabulary?

Is it different from the previous tasks?

Question 1.4 – Smoothed Bigrams Model



During preprocessing...

How to treat the words not in the vocabulary?

Is it different from the previous tasks?

↳ Replace them by <UNK> token

Needed for bigrams, <UNK> can be used as a context or predicted word (sometimes both)

Questions?