

NLP course 2021

Homework 2

Aspect-Based Sentiment Analysis

Prof. Roberto Navigli

Teaching assistants:

- Cesare Campagnano
- Pere-Lluís Huguet Cabot



SAPIENZA
UNIVERSITÀ DI ROMA

**SAPIENZA
NLP**



Aspect-Based Sentiment Analysis

A gentle introduction



Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

It usually consists of the following subtasks:

1. aspect term identification

Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

It usually consists of the following subtasks:

1. aspect term identification
2. aspect term polarity classification

Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

It usually consists of the following subtasks:

1. aspect term identification
2. aspect term polarity classification
3. aspect category identification

Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

It is usually consists of the following subtasks:

1. aspect term identification
2. aspect term polarity classification
3. aspect category identification
4. aspect category polarity classification

Aspect-Based Sentiment Analysis (ABSA)

ABSA aims to **identify the aspect terms** of given target entities (in our case **restaurants** and **laptops**) and the **sentiment expressed towards each aspect**.

It is usually consists of the following subtasks:

1. aspect term identification → **a**
2. aspect term polarity classification → **b**
3. aspect category identification → **c**
4. aspect category polarity classification → **d**

For the sake of simplicity, let's call them **a**, **b**, **c** and **d**

ABSA: aspect term identif. + classif. (a + b)

Input: I love their pasta but I hate their Ananas Pizza.

ABSA: aspect term identif. + classif. (a + b)

Input: I love their **pasta** but I hate their **Ananas Pizza**.

ABSA: aspect term identif. + classif. (a + b)

Input: I love their **pasta** but I hate their **Ananas Pizza**.

Output: {(pasta, positive); (Ananas Pizza, negative)}

ABSA: aspect term identif. + classif. (a + b)

Input: I love their **pasta** but I hate their **Ananas Pizza**.

Output: {(pasta, positive); (Ananas Pizza, negative)}

Multi-word aspect terms should
be treated as single terms!

Examples

Input: Ananas Pizza was great to see, but not to taste!

Output: {(Ananas Pizza, conflict)}

Used when there is more than
one contrasting polarity

Examples

Input: I will come back for sure!

Output: {}

There can be samples with no aspect terms at all!

ABSA: aspect category identif. + classif. (c + d)

Input: The people there is so kind and the taste is
exceptional

Output: {(staff, positive); (food, positive)}

The homework

Aspect-Based Sentiment Analysis



Homework 2 - Aspect-Based Sentiment Analysis



The homework

- a) aspect term identification
- b) aspect term polarity classification
- c) aspect category identification
- d) aspect category polarity classification

Mandatory

Extra!

Multi-step model

- Mandatory (**a** + **b**)
 - Subtasks **a** (aspect term identification) and **b** (aspect term polarity classification) are mandatory and **will be evaluated separately**, i.e. both will be scored and **must surpass** a (very low) **baseline** to pass.
 - Instead, the **overall score (a + b)** **will be computed on both** and it will count for the **quantitative evaluation** (see evaluation slide, page X)
- Extra (**c** + **d**)
 - Subtasks **c** (aspect category identification) and **d** (aspect category polarity classification) have no constraints, will not count on the quantitative evaluation but towards the extra points and must surpass a (medium) baseline to be considered as extra.

Multi-step model

Example (subtask **a**): given a sentence, identify the aspect terms



Multi-step model

Example (subtask **a**): given a sentence, identify the aspect terms

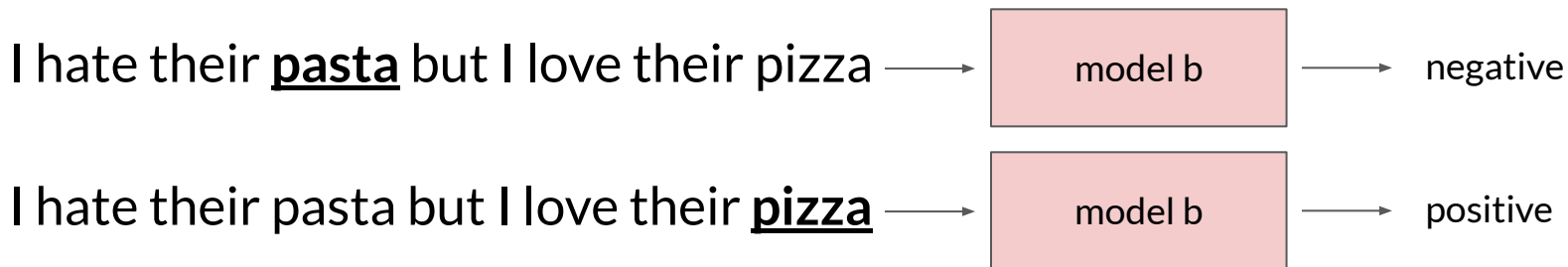


Hint: check out the notebook #6 (in particular, the NER part!)

Multi-step model

Example (subtask **b**):

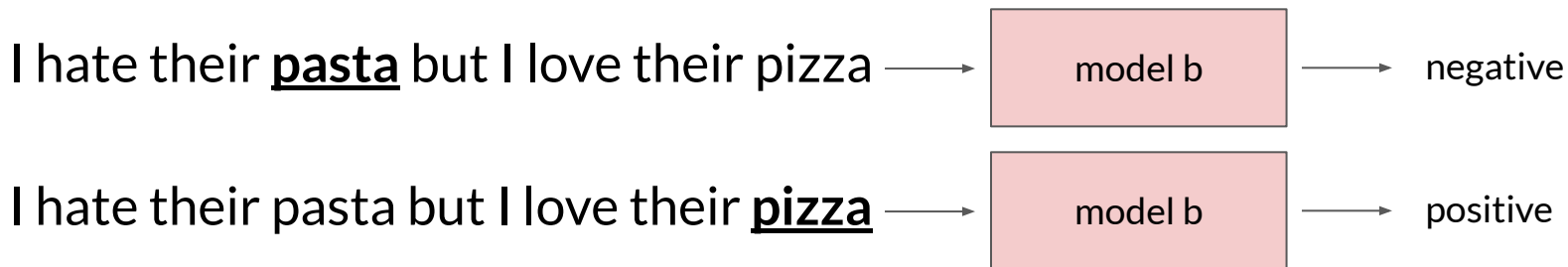
given a sentence and an aspect term, identify the polarity towards the aspect term



Multi-step model

Example (subtask **b**):

given a sentence and an aspect term, identify the polarity towards the aspect term

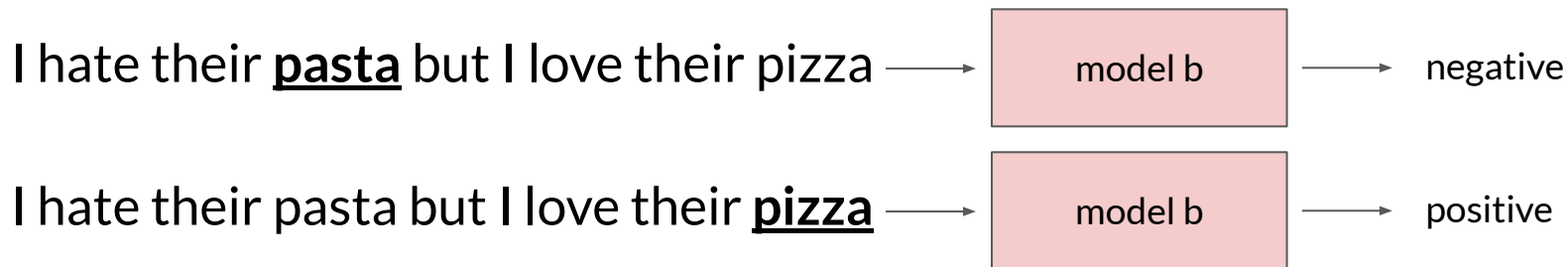


Hint 1: you should indicate where the target word is in the sentence. There are many ways to do this, either a simple 0/1 flag, a trainable flag embedding, special tokens, etc. Be creative!

Multi-step model

Example (subtask **b**):

given a sentence and an aspect term, identify the polarity towards the aspect term



Hint 1: you should indicate where the target word is in the sentence. There are many ways to do this, either a simple 0/1 flag, a trainable flag embedding, special tokens, etc. Be creative!

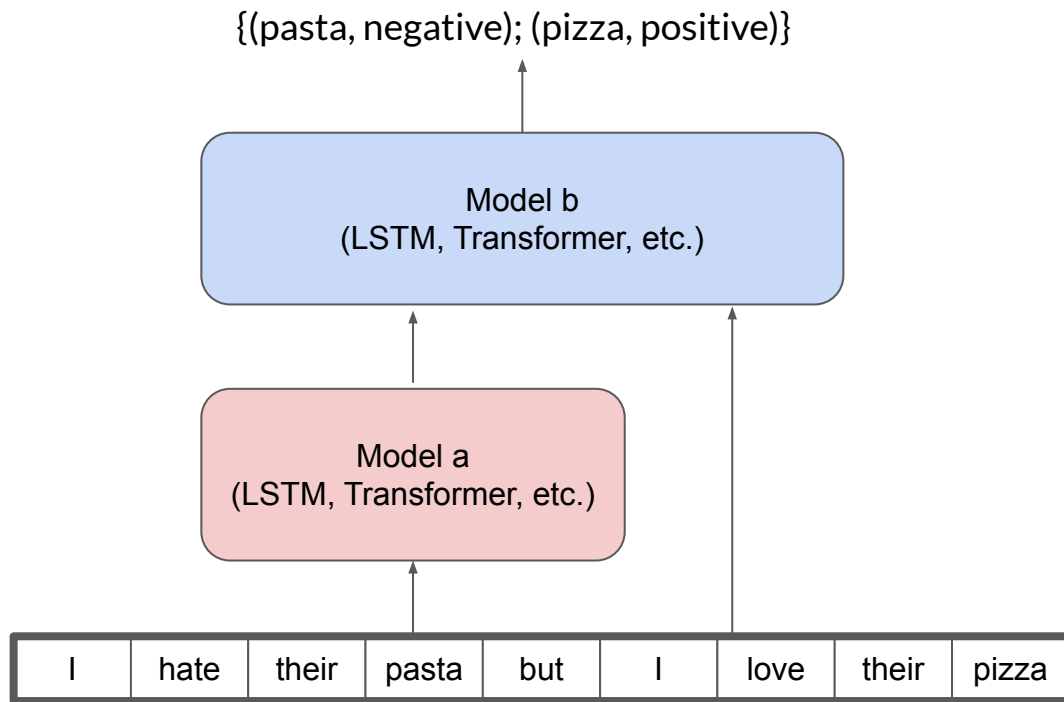
Hint 2: the formulation is similar to WiC, you could start from your WiC architecture!

Modeling ABSA

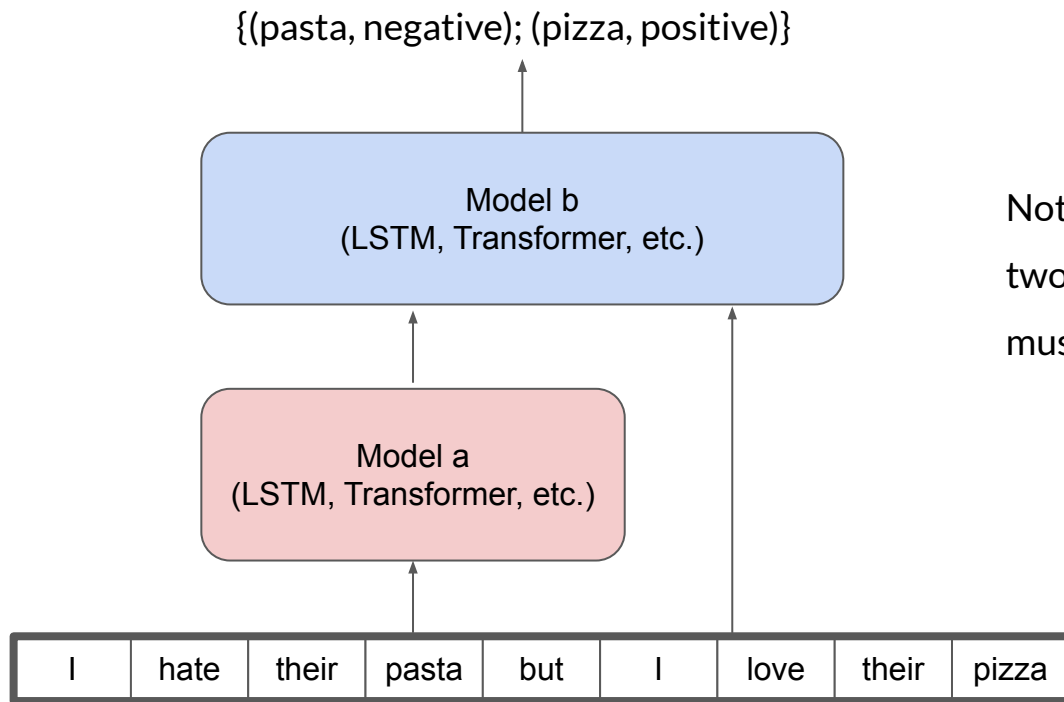


Homework 2 - Aspect-Based Sentiment Analysis

Modeling ABSA



Modeling ABSA



Note: It is not mandatory that **a** and **b** are two different models! But the two steps must be able to be evaluated separately!

ABSA: evaluation (model_ab)

The performance of a ABSA system is usually measured in terms of Macro F1.

"I love their pasta but I hate their Ananas Pizza."

Gold Term:	pasta	Ananas Pizza	Pred Term:	Ananas Pizza	their	pasta
Gold Sentiment:	positive	negative	Pred Sentiment:	negative	positive	neutral

$$\text{Precision} = 100 * \text{tp} / (\text{tp} + \text{fp})$$

$$\text{Recall} = 100 * \text{tp} / (\text{tp} + \text{fn})$$

$$\text{F1} = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$$

ABSA: evaluation (model_ab)

The performance of a ABSA system is usually measured in terms of Macro F1.

"I love their pasta but I hate their Ananas Pizza."

Even if aspect is correct!

Gold Term:	pasta	Ananas Pizza	Pred Term:	Ananas Pizza	their	pasta
Gold Sentiment:	positive	negative	Pred Sentiment:	negative	positive	neutral

Precision = $100 * tp / (tp + fp) = 100 * 1 / (1+2) = 33.33\%$

Recall = $100 * tp / (tp + fn) = 100 * 1 / (1+1) = 50\%$

F1 = $2 * precision * recall / (precision + recall) = 39,6\%$

True positive

False positive

False negative

ABSA: evaluation

For **model b** evaluation is the same but aspects are given.

"I love their pasta but I hate their Ananas Pizza."

Gold Term:	pasta	Ananas Pizza	Pred Term:	Ananas Pizza	their	pasta
Gold Sentiment:	positive	negative	Pred Sentiment:	negative	positive	neutral

Precision = $100 * tp / (tp + fp) = 100 * 1 / (1+2) = 50\%$

Recall = $100 * tp / (tp + fn) = 100 * 1 / (1+1) = 50\%$

F1 = $2 * precision * recall / (precision + recall) = 50\%$

True positive

False positive

False negative

ABSA: evaluation

Model c + d evaluation is the same as **model a + b**, but for a given set of categories.

For **model a + b** and **model c + d** the performances per **sentiment** and **category** will be given.

Performance for aspect extraction will be computed from **model a + b** (non aspect words do not count and only exact matches are considered).

Output will give **Precision**, **Recall** and **F1**, as well as **True Positives**, **False Negatives** and **False Positives**.

ABSA: evaluation

Example for **model b** (with a Random baseline):

MODEL: ASPECT SENTIMENT

Aspect Sentiment Evaluation

ALL	TP: 264;	FP: 397;	FN: 396						
	(m avg): precision: 39.94;		recall: 40.00;	f1: 39.97 (micro)					
	(M avg): precision: 25.47;		recall: 25.79;	f1: 25.58 (Macro)					
positive:	TP: 192;	FP: 152;	FN: 168;	precision: 55.81;	recall: 53.33;	f1: 54.55;			344
negative:	TP: 49; FP: 132;		FN: 107;	precision: 27.07;	recall: 31.41;	f1: 29.08;			181
neutral:	TP: 23; FP: 98; FN: 102;			precision: 19.01;	recall: 18.40;	f1: 18.70;			121
conflict:	TP: 0; FP: 15; FN: 19;			precision: 0.00;	recall: 0.00;	f1: 0.00;			15

ABSA: evaluation

Example for model a + b (with a Random baseline):

MODEL: ASPECT SENTIMENT + ASPECT EXTRACTION

Aspect Extraction Evaluation

Aspects TP: 37; FP: 729; FN: 622
precision: 4.83; recall: 5.61; f1: 5.19

Aspect Sentiment Evaluation

ALL TP: 15; FP: 753; FN: 645
(m avg): precision: 1.95; recall: 2.27; f1: 2.10 (micro)
(M avg): precision: 1.22; recall: 1.44; f1: 1.32 (Macro)

positive:	TP: 11; FP: 407; FN: 349;	precision: 2.63;	recall: 3.06;	f1: 2.83;	418
negative:	TP: 3; FP: 204; FN: 153;	precision: 1.45;	recall: 1.92;	f1: 1.65;	207
neutral:	TP: 1; FP: 126; FN: 124;	precision: 0.79;	recall: 0.80;	f1: 0.79;	127
conflict:	TP: 0; FP: 16; FN: 19; precision: 0.00;	recall: 0.00;	f1: 0.00;	16	

ABSA: evaluation

Example for model c + d (with a Random baseline):

Category Extraction Evaluation

ALL	TP: 193;	FP: 419;	FN: 351				
	(m avg): precision: 31.54;		recall: 35.48;	f1: 33.39 (micro)			
	(M avg): precision: 25.70;		recall: 29.36;	f1: 27.29 (Macro)			
anecdotes/miscellaneous:	TP: 80;	FP: 149;	FN: 111;	precision: 34.93;	recall: 41.88;	f1: 38.10;	229
price:	TP: 12;	FP: 64;	FN: 41;	precision: 15.79;	recall: 22.64;	f1: 18.60;	76
food:	TP: 92;	FP: 129;	FN: 132;	precision: 41.63;	recall: 41.07;	f1: 41.35;	221
ambience:	TP: 9;	FP: 77;	FN: 67;	precision: 10.47;	recall: 11.84;	f1: 11.11;	86

Category Sentiment Evaluation

ALL	TP: 88;	FP: 549;	FN: 575				
	(m avg): precision: 13.81;		recall: 13.27;	f1: 13.54 (micro)			
	(M avg): precision: 7.99;		recall: 7.46;	f1: 7.70 (Macro)			
positive:	TP: 75;	FP: 299;	FN: 301;	precision: 20.05;	recall: 19.95;	f1: 20.00;	374
negative:	TP: 9;	FP: 140;	FN: 158;	precision: 6.04;	recall: 5.39;	f1: 5.70;	149
neutral:	TP: 4;	FP: 64;	FN: 85;	precision: 5.88;	recall: 4.49;	f1: 5.10;	68
conflict:	TP: 0;	FP: 46;	FN: 31;	precision: 0.00;	recall: 0.00;	f1: 0.00;	46



Submission

What you will receive &
How to submit



What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive"],  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive"],  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

Input sentence

What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive",  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

Target aspects (there may be 0 or more)

What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive"],  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

[boundaries, surface form,
sentiment]

What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive"],  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

Categories data, only for
restaurants data for the extra
task

What you will receive: data format

The dataset is a JSON file where each entry contains the following fields:

```
{  
  "text": "I love their pasta but I hate their Ananas Pizza. ",  
  "targets": [  
    [13, 17], "pasta", "positive"],  
    [[36, 47], "Ananas Pizza", "negative"]  
  ],  
  "categories": [  
    ("food", "conflict")  
  ],  
}
```

Target categories (there may be 1 or more)

What you will receive: files

We will provide you a folder with the following structure:

```
nlp2021-hw2/  
  data/  
  hw2/  
    model.py  
    simple_test.py  
    stud/  
  model/  
  requirements.txt  
  test.sh
```

You are allowed to edit **only** the items in bold.

We will use Docker for evaluation. As far as you **do not change** any file but those we marked in bold, **if test.sh runs** on your side, it will run on ours as well. Find the code repository [here](#). You can use **simple_test.py** to test your predict functions.

What you will receive: datasets

In the `data/` folder you will find two datasets:

- `restaurants`,
- `laptops`

with the same annotation scheme.

Keep in mind that only `restaurants` is annotated for tasks **c + d**, so the evaluation will be performed using just `restaurants` data.

i.e. if you use just one model for **a + b + c + d**, make sure not to backpropagate `laptops` data for task **c + d** or it may affect performance. **Both datasets should be used to train model b and a + b and will be evaluated jointly.**

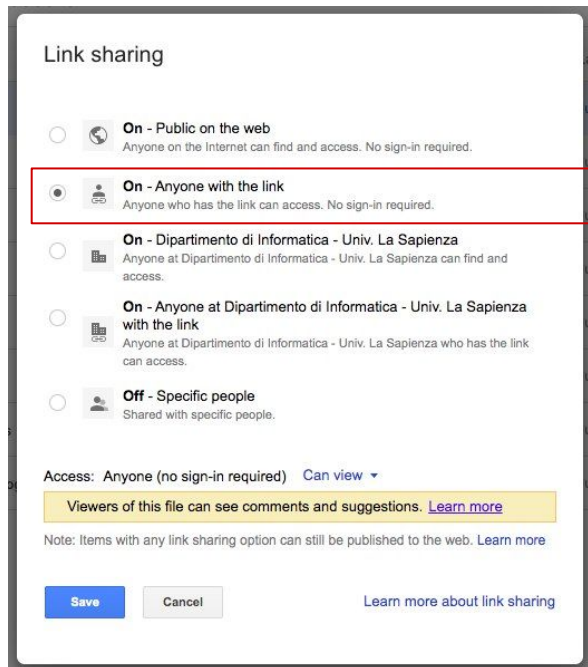
What we expect from you

- The zip folder we gave you (but populated :))
- Put your training code (if you used Colab, download the notebook **.ipynb** and place it) in **hw2/stud/**
- If you use any additional library, modify the requirements.txt file as needed (click [here](#) for info)
- Use the data (train, dev and test) in the data folder; **use each file as defined in the standard ML conventions** (train for training, dev for model selection, ...)
- Put everything your model needs (vocabulary, weights, ...) inside the **model/** folder, and be sure to properly load them in your model

What we expect from you

1. In `hw2/stud/implementation.py` implement the `StudentModel` class
 - Load your model and use it in the `predict` method
 - You **must respect the signature** of the `predict` method
 - You can add other methods (i.e. the constructor)
2. In `hw2/stud/implementation.py` implement the `build_model` function, initializing your `StudentModel` class.
3. Use `test.sh` to check that everything works
4. Add your `report.pdf` to the folder (yes, export it in pdf even if you are using Word!)
5. Name the zip folder `lastname_studentid_hw2.zip`
 - Ex: Luigi D'Andrea will submit a file named `dandrea_1234567_hw2.zip`

Submission instructions



The screenshot shows the 'Link sharing' dialog box in Google Drive. The 'On - Anyone with the link' option is selected and highlighted with a red rectangle. Below it, the 'On - Dipartimento di Informatica - Univ. La Sapienza' option is also visible. At the bottom, there is a 'Save' button and a 'Cancel' button. A yellow banner at the bottom of the dialog states: 'Viewers of this file can see comments and suggestions. [Learn more](#)'. A note below the banner says: 'Note: Items with any link sharing option can still be published to the web. [Learn more](#)'.

Link sharing

☐ **On - Public on the web**
Anyone on the Internet can find and access. No sign-in required.

☒ **On - Anyone with the link**
Anyone who has the link can access. No sign-in required.

☐ **On - Dipartimento di Informatica - Univ. La Sapienza**
Anyone at Dipartimento di Informatica - Univ. La Sapienza can find and access.

☐ **On - Anyone at Dipartimento di Informatica - Univ. La Sapienza with the link**
Anyone at Dipartimento di Informatica - Univ. La Sapienza who has the link can access.

☐ **Off - Specific people**
Shared with specific people.

Access: Anyone (no sign-in required) [Can view](#) ▼

Viewers of this file can see comments and suggestions. [Learn more](#)

Note: Items with any link sharing option can still be published to the web. [Learn more](#)

[Save](#) [Cancel](#) [Learn more about link sharing](#)

- Upload the zip on your **institutional** Drive and make it **link-shareable** and **public to anyone** (an automatic script will download it).
- Make sure it is accessible via an incognito page of your browser!
- Do **NOT** modify the folder structure
- You have to submit the homework through the submission form on Google Classroom. You will be asked to fill a form with the requested information and the **link** to the zip you uploaded on Drive.



Evaluation

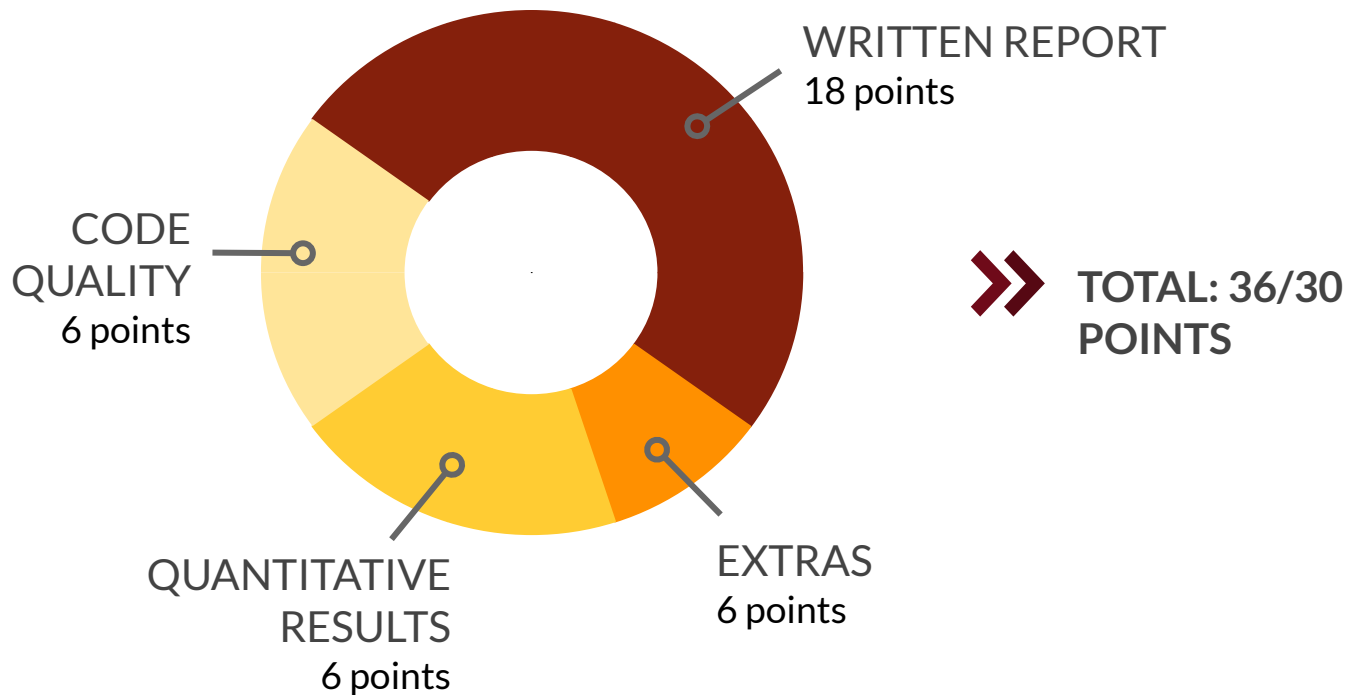
How your work will be evaluated



Homework 2 - Aspect-Based Sentiment Analysis



Evaluation Overview



Report: dos and don'ts

- **ACL 2021 paper template**
 - Available [here](#) (Word and LaTeX direct download) or [here](#) (Overleaf LaTeX template)
 - You can use either the LaTeX or the Word template, your choice
 - **DO NOT MODIFY** the template (margins, spacing, font size)
 - Use the non-anonymous flag, so you can enter your name
- **Max 3 pages**
 - For the report, including title, subtitles, etc.
 - This is a **STRICT RULE!**
- **Unlimited extra pages for images, tables and references**
 - Every image and table must have a caption (don't abuse them please :-))
 - Tables and images must be referenced in the report

Report: what you are expected to do

We expect a good report to be:

- **Readable and understandable**
 - We will not give penalties for English errors, but we expect the report to follow a clear flow. We don't want to read just a sequence of statements on what you did without showing the reasoning behind your choices
- **Well-structured and organized**
 - Take inspiration from the many papers available online and organize your report in well-defined sections (e.g. method, setup, experiments, results...)

Report: what you are not expected to do

We expect a good report **NOT** to include:

- Unnecessary **task** or **dataset descriptions**
 - just focus on your solution to the problem.
- **Code** copy-paste
 - Your code should be self-explanatory, so no need to show it in the report. You can add **pseudo-code** to show some particular algorithm, but **no code or screenshots** please!

Report: what you are not expected to do

We expect a good report **NOT** to include:

- Unnecessary **low-level implementation details**
 - Avoid any **low-level implementation/technical details** like “I used a dictionary to store these values”, “I had to use configuration X to solve this exception”, “I could not use Y because there was a dependency issue with Z”, etc.
 - Instead, **we are interested in high-level abstractions/strategies** you decide to use to tackle the homework, as well as the **intuitions behind your choices**.
E.g. use and description of a particular model, explanation of how and why an architecture works, etc.

Code and code Quality

Your project should conform to the following rules:

- You **MUST** use PyTorch.
 - TensorFlow and other deep learning frameworks are **NOT** allowed.
- PyTorch Lightning **MUST** be used as high level abstraction to train your model/s.
- HuggingFace Transformers and torchtext are now allowed
- **Frameworks** that use PyTorch (e.g. AllenNLP, etc.) are **NOT** allowed.
 - Libraries (such as tqdm, sklearn, NLTK) are fine, but since the line between a framework and a library is sometimes blurred, please ask in the Google Classroom group before using any external library: **any other library MUST be agreed with the TAs.**

Code and code Quality

Your project should conform to the following rules:

- **You are now allowed** to use tools/architectures that have been explained in the class, e.g.:
 - word embeddings (Word2Vec, GloVe, etc.) **are allowed**,
 - contextualized word embeddings (ELMo, etc.) **are allowed**,
 - Transformer-based models (BERT, BART, RoBERTa, XLM, etc.) **are allowed**.
- For any doubt, please ask the TAs on Google Classroom.
- **Comment** your code, please!

Baselines

We will evaluate the **performance of your model/s** on a SECRET test set.

To consider your submission, you must at least reach:

- 25% (Macro F1-score) for task **a**
- 30% (Macro F1-score) for task **b**

Otherwise, it will be considered FAIL.

Baselines (Extra!)

We will evaluate the **performance of your model/s** on a SECRET test set.

To consider the extra part in your submission, you must at least reach:

- 75% (Macro F1-score) for tasks **c + d**

Otherwise, it will not be considered for extra points.

Quantitative Results

We will evaluate the **performance of your model** on a SECRET test set.

The overall score considered for evaluation is the one of tasks **a + b**

You can get **from 0 to 6** points according to the following **thresholds**:

- $P < 0.30 \Rightarrow 0$
- $0.30 < P < 0.35 \Rightarrow 1$
- $0.35 < P < T3 \Rightarrow 2$
- $T3 < P < T4 \Rightarrow 3$
- $T4 < P < T5 \Rightarrow 4$
- $T5 < P < T6 \Rightarrow 5$
- $P > T6 \Rightarrow 6$

Thresholds will be defined based on an internal reference model and the **normalized distribution of YOUR scores!**

Extras

You can achieve **up to 6 points with some extras!**

An “extra” is whatever you decide to add to your model to make it better. For instance:

- implement parts **c** + **d** of the ABSA pipeline,
- use of pre-trained embeddings,
- use of NLP best practices,
- comparative analysis of results in your report,
- informative plots in your report,
- **new ideas** (including using external resources in a clever way, please see slide X)

and more, according to internal baselines. Don't forget to **explain your choices** in the report!
Extras that are not explained in the report will not be considered for evaluation.

Evaluation

- `test.sh` is identical to what we will be using
- If it does not run on your side, we will not correct your homework
- Note that, if you use **any kind of hard-coded paths**, this script **won't work**
- Use **paths relative to the project root folder**, e.g.:
 - **NO:** `/home/pincopallino/my_folder/model/weights.th`
 - **OK:** `model/weights.th`

Warnings

Things you should be aware of



Please be aware that

This is an **individual homework**! Collaboration among the students is **not** allowed.

We will check for **plagiarism** both manually and automatically.

It is **not allowed** to:

- Copy from other students
- Share your code with other students
- Copy from online resources (StackOverflow, GitHub, Medium and so on).

However, you are allowed to use material from **external sources** as long as it is **not central** to the homework.

- In this case, it is **MANDATORY** to cite such resources in the report

Please be aware that

- If we find out that you breached any of the above rules, you will **automatically FAIL** this homework and you will have to pass a **FULL EXAM**.
- **Plagiarism will imply further consequences at the Faculty level.**
- While we release the homework on GitHub, **DO NOT FORK THE PROJECT**.
- If you want to continue using GitHub for versioning, clone the project and re-upload it.
- If we realize you shared your code in any way (forking or otherwise), even without the intention of letting others copy, you will be failed automatically.

Use of external data

- For your experiments, **use the provided data** (train and dev) in the data folder; use each file as defined in the standard ML conventions (train for training, dev for model selection).
- **Use only the training set to train the model that you submit for evaluation.** If you train it on more data (dev set or any other external data), it will be a **FAIL**.
- You can use external resources (E.g. BabelNet, etc.); if you have doubts, please ask on the Google Classroom.

Tips

A few tips to organize your work:

- **Start as soon as possible!**
 - Training a neural network requires time, possibly hours, depending on your hardware
- **Don't wait until last day to test the submitting code!!!!**
- **Start small!**
 - If you don't get decent results with a very very simple neural network, there is a good chance that adding other things won't make your model perform better
 - Leave the “extras” as the last thing!
- Leave some time for **hyperparameter** tuning!
 - Sometimes good hyperparameter combinations can do wonders for your neural network
- Use **Google [Colab](#)** (free GPUs!)



Deadline

When to deliver what

Deadline

Submission date: **June Xth, 2021**
23:59:59 Italian time (UTC + 1)

Submit the homework through the submission form on Google Classroom. You have to fill the form with the requested information and a link to the zip folder of the homework on Google Drive.



Awards

Get a **Sapienza NLP™** t-shirt



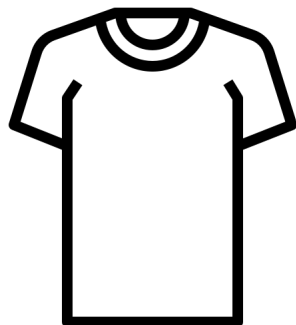
Homework 2 - Aspect-Based Sentiment Analysis



Win a Sapienza NLP t-shirt!

We will hand out amazing Sapienza NLP t-shirts to the **overall top-5** students!

The final ranking will be computed according to the scores on our **secret** test set.



That's not all

If your work is novel, interesting and original, we will gladly invite you to work together with us to extended on a fully-fledged paper for **TOP-TIER INTERNATIONAL CONFERENCE!**

Just over the last 12 months, the Sapienza NLP group published more than a dozen of papers!

Questions?

If you have a question that may interest your colleagues, **please ask it on [Google Classroom](#)**.

Otherwise, for personal or other questions, send an email to **ALL** of us (but please, only reach for things that can't be asked on the Google Classroom).

Our emails are:

campagnano@di.uniroma1.it

huguetcabot@babelscape.com