Assignment 2 - Semantic Role Labelling or Text Detoxification

1. Introduction

- Please read the document very carefully. If you have questions ask in the telegram group.
- Select ONE task: either the task from Section 2.1 OR the from Section 2.2.
- Use this template to complete your assignment and upload it to Canvas: https://colab.research.google.com/drive/1IyvayaR7KS9JyofdQ14b8J3kwJ8QCLrj?usp=sharing
- For both tasks you have to submit your solutions to Codalab (links can be found below).

Semantic role labelling: https://codalab.lisn.upsaclay.fr/competitions/531
Detoxification: https://codalab.lisn.upsaclay.fr/competitions/532

2. Tasks Description

Below two tasks are presented (you need to solve one of them).

2.1 Semantic Role Labelling

2.1.1 Introduction

Our task is motivated by human need to compare various objects: different models of mobile phones, cars, programming languages, countries, etc. This information need has been addressed in NLP research, but there is currently much room for the improvement of existing comparative question answering systems.

For example, the CAM (Comparative Argument Mining) system[1] is given a pair of objects to compare and retrieves arguments in favour each of them. It also extracts

[1] http://ltdemos.informatik.uni-hamburg.de/cam/

predicates (comparative characteristics of the objects, e.g. *easier*, *better*, *faster*, etc.) and **aspects** (features along which the objects are compared, e.g. *speed*, *screen*, *performance*, etc.) from the arguments (comparative sentences).
Aspects and predicates are extracted using hand-written templates which have low recall (fail to extract entities which do not conform to the templates) and occasionally extracts incorrect entities.

We would like to improve its performance by training a model which extracts objects, aspects, and predicates from a sentence. This model should be trained on sentences where words or phrases are labelled with entities.

Examples of sentences

Postgres is easier to install and maintain than Oracle.

[Postgres OBJECT] is [easier PREDICATE] to [install ASPECT] and [maintain ASPECT] than [Oracle OBJECT].

Instances can be multiword:

Advil works better for body aches and pains than Motrin.

[Advil OBJECT] works [better PREDICATE] for [body aches ASPECT] and [pains ASPECT] than [Motrin OBJECT].

Data format

The provided data files are in CoNLL format. Each line contains one word and its label, separated by a tab ("Word<TAB>label"), the end of the sentence is marked with an empty line. The labels are in BIO format, where each of the entity labels ("Object", "Aspect", "Predicate") is prepended with a prefix "B-" or "I-", indicating the **b**eginning of an entity (the first word of an entity) and the **i**nside of an entity (the second and all subsequent words). Words which are not a part of an entity are labelled with "O":

advil B-Object

works O

better B-Predicate

for O

body B-Aspect

aches I-Aspect

and O

pains B-Aspect

than O

motrin B-Object

. O

2.1.2 Task formulation

The data consists of comparative sentences (i.e. sentences which contain comparison of two or more objects). The data contains three types of entities:

- **Object** - objects which are being compared
- **Aspect** - features along which the objects are compared
- **Predicate** - words or phrases which implement the comparison (usually comparative adjectives or adverbs)

The dataset uses BIO labelling scheme:

- The first word of an entity is labelled with “B-<entity-type>” (*beginning* of an entity)
- The second and further words of an entity are labelled with “I-<entity-type>” (*inside* of an entity)
- Words which are not a part of an entity are labelled with “O” (*out* of entity)

Therefore, our dataset uses the following labels:

- O
- B-Object
- I-Object
- B-Aspect
- I-Aspect
- B-Predicate
- I-Predicate

Your task is to assign one of such labels to each of the words in the test set.

2.1.3 Evaluation metrics

The result will be evaluated with F_1-score:

$$2 \cdot \frac{precision \cdot recall}{precision + recall} = 2 \cdot \frac{tp}{tp + 0.5(fp + fn)}$$

We will consider scores for all individual classes except O. For multi-word entities we use the “relaxed” metric: if the borders of a predicted entity match those of the reference entity, we add 1 to *tp* quantity (number of *true positive* examples). If there is only a partial match, we add a number between 0 and 1 computed as the intersection length divided by the full entity length.

2.1.4 Method

Your task is to train a sequence labelling model on a provided labelled dataset. You can use neural architectures (LSTM, Transformer) as well as traditional ML algorithms for sequence labelling (CRF, HMM). We encourage you to experiment with different types of embeddings (e.g. context-free GloVe and fastText or context-informed ELMo and BERT).

In the context of this assignment, you will solve a sequence labelling task on the dataset of comparative sentences provided by the course team. You will need to train a model and submit your solution to the CodaLab competition: https://codalab.lisn.upsaclay.fr/competitions/531 .

2.1.5 Expected output

You are expected to:

1. **Develop a solution of the task** and provide a reproducible code in the form of an ipython notebook (preferably a link to a Google Colab[2] notebook):
    - You should use Python 3,
    - The notebook should contain code for installation of all dependencies,
    - The notebook should contain code for downloading all the additional datasets it uses,
    - The notebook should reproduce the results you submit to CodaLab:
        i. It should generate the output file of the required format and compute the scores for it - this result should be reached by running your notebook cell by cell without changing anything, including paths to files!
        ii. The scores should be close to the scores of your CodaLab submission.
        iii. If this reproducibility is not reached your grade will be lowered.
2. **Write a report** using the ACL latex template[3] which describes the method used in your solution. The paper should be formatted as a short paper i.e. will be at most 4 pages long with an unlimited number of pages for references and follow the structure of a usual paper as those found in the ACL Anthology.[4] Namely, sections such as Introduction, Methodology, Results, and Conclusion should be

[2] https://colab.research.google.com/notebooks/intro.ipynb#recent=true
[3] https://www.overleaf.com/latex/templates/acl-2020-proceedings-template/zsrkcwjptpcd
[4] https://www.aclweb.org/anthology/

present (cf this paper as an example[5]). The section ‘Related Work’ can be omitted in your reports.

3. **Push the (best) solutions** which you developed to the CodaLab competition: https://codalab.lisn.upsaclay.fr/competitions/531 so that they appear in the leaderboard. The name of your user / submission should be present in the report for verification.

2.2 Text Detoxification

2.2.1 Introduction

Global access to the Internet has enabled the spread of information all over the world and has given many new possibilities. On the other hand, alongside the advantages, the exponential and uncontrolled growth of user-generated content on the Internet has also facilitated the spread of toxicity and hate speech. Much work has been done in the direction of offensive speech detection. However, it has become essential not only to detect toxic content but also to combat it in smarter ways. While some social networks block sensitive content, another solution can be to detect toxicity in a text which is being typed in and offer a user a non-offensive version of this text. This task can be considered a style transfer task, where the source style is toxic, and the target style is neutral/non-toxic. The task of style transfer is the task of transforming a text so that its content and the majority of properties stay the same, and one particular attribute (style) changes. This attribute can be the sentiment, the presence of bias, the degree of formality, etc. Considering the task of detoxification, it has already been tackled by different groups of researchers, as well as a similar task of transforming text to a more polite form. However, all these works deal only with the English language. As for Russian, the methods of text style transfer and text detoxification have not been explored before.

2.2.2 Task formulation

You have a great chance to be the first participant in the competition of automatic detoxification of Russian texts to combat offensive language. Such a kind of textual style transfer can be used, for instance, for processing toxic content in social media.

[5] https://www.aclweb.org/anthology/P19-1325/

While much work has been done for the English language in this field, it has never been solved for the Russian language yet.

We define the detoxification task as the task of style transfer: from the toxic style to the non-toxic style. We want to rewrite the sentence and preserve the context.

We define the task of style transfer as follows. Let us consider two corpora $D^X = \{x_1, x_2, \ldots, x_n\}$ and $D^Y = \{y_1, y_2, \ldots, y_m\}$ in two different styles –s^X (toxic) and s^Y (non-toxic), respectively. The task is to create a model $f_\theta : X \to Y$, where X and Y are all possible texts with styles s^X and s^Y. The task of selecting the optimal set of parameters θ for f consists maximising the probability $p(y' \mid x, s^Y)$ of transferring a sentence x with the style s^X to the sentence y' which saves the content of x and has the style s^Y. The parameters are maximised on the corpora D^X and D^Y which can be parallel or non-parallel. We focus on the transfer $s^X \to s^Y$, where s^X is the toxic style, and s^Y is neutral.

2.2.3 Evaluation metrics

To perform a comprehensive evaluation of a style transfer model, we need to make sure that it (i) changes the text style, (ii) preserves the content, and (iii) yields a grammatical sentence. The majority of works on style transfer use individual metrics to evaluate the three parameters. In our competition we use the following metrics:

1) Style transfer accuracy: a classifier that evaluates the toxicity class of a rewritten sentence
2) ChF: character n-gram F score between the rewritten sentence and the manually detoxified reference

2.2.4 Methods

In the context of this assignment, you will solve a style transfer task on the dataset of comparative sentences provided by the course team. You will need to train a model and submit your solution to the CodaLab competition: https://codalab.lisn.upsaclay.fr/competitions/532 .

You are free to use any methods and/or models for style transfer or pretrained models for text generation (GPT, T5, etc.). Here are some baselines you may want to improve:

Duplicate this is a naive baseline that amounts to performing no changes to the input sentence. It represents a lower bound of the performance of style transfer models, i.e. it helps us check that the models do not contaminate the original sentence.

Delete: this method eliminates toxic words based on a predefined toxic words vocabulary. The idea is often used on television and other media: rude words are bleeped out or hidden with special characters (usually an asterisk). The main limitation of this method is vocabulary incompleteness: we cannot collect all the rude and toxic words. Moreover, new offensive words and phrases can appear in the language that can be also concatenated with different prefixes and suffixes. On the other hand, this method can preserve the content quite well, except for the cases when toxic words contain meaning that is essential for the understanding of the whole text.

Retrieve: This method is targeted at improving the accuracy of style transfer. For a given toxic sentence, we retrieve the most similar non-toxic text from a corpus of non-toxic samples. In this case, we get a safe sentence. However, the preservation of the content depends on the corpus size and is likely to be very low.

2.2.5 Expected output

Example of the input and model output are presented below:

Model	Sentence
Input	не дай бог моя дочь так оденется убью нахуй палкой (If, God forbid, my daughter goes out dressed like this, I'll fucking kill her with a stick)
Delete	не дай бог моя дочь так оденется убью палкой (If, God forbid, my daughter goes out dressed like this, I'll kill her with a stick)
Retrieve	не бросайте угла родного одной мы лежали больнице палате в в в те девчонкой была молодой годы (don't abandon your native corner same hospital we were ward in in in those girl was young years)

As output we expect a paraphrased (rewritten) toxic sentence in a more neutral (non-toxic) style. For each input sentence x_i we expect a corresponding rewritten sentence y_i.

Please, submit a textual file *results.txt* each rewritten sentence in a row. Make sure that the size of the output dataset is the same as the size of the input dataset. Please, submit the result file in the zip archive in Codalab: **https://codalab.lisn.upsaclay.fr/competitions/532** .

You are expected to:

1. **Develop a solution of the task** and provide a reproducible code in the form of an ipython notebook (preferably a link to a Google Colab[6] notebook):

[6] https://colab.research.google.com/notebooks/intro.ipynb#recent=true

    - You should use Python 3,
    - The notebook should contain code for installation of all dependencies,
    - The notebook should contain code for downloading all the additional datasets it uses,
    - The notebook should reproduce the results you submit to CodaLab:
        i. It should generate the output file of the required format and compute the scores for it - this result should be reached by running your notebook cell by cell without changing anything, including paths to files!
        ii. The scores should be close to the scores of your CodaLab submission.
        iii. If this reproducibility is not reached your grade will be lowered.
2. **Write a report** using the ACL latex template[7] which describes the method used in your solution. The paper should be formatted as a short paper i.e. will be at most 4 pages long with an unlimited number of pages for references and follow the structure of a usual paper as those found in the ACL Anthology.[8] Namely, sections such as Introduction, Methodology, Results, and Conclusion should be present (cf this paper as an example[9]). The section 'Related Work' can be omitted in your reports.
3. **Push the (best) solutions** which you developed to the CodaLab competition: https://codalab.lisn.upsaclay.fr/competitions/532 so that they appear in the leaderboard. The name of your user / submission should be present in the report for verification.

3. Evaluation criteria

Technical report		**Code**		**Results**		**Total**	
Methodology	Discussion of results	Readability	Reproducibility	Improved over the baseline	top-1 - 10 points top-20% - 5 points	**100% + bonus**	**Penalty for late submission**
5	5	5	5	5 or 10	0 or 5 or 10	**25** + (5 or 10)	1 day = 1 point

[7] https://www.overleaf.com/latex/templates/acl-2020-proceedings-template/zsrkcwjptpcd
[8] https://www.aclweb.org/anthology/
[9] https://www.aclweb.org/anthology/P19-1325/

* To get 100% for this task you need to achieve 25 points, but you can get additional 5 points if your method is in the top 20% (among all enrolled students) in the Codalab leaderboard and additional 10 points if your method is the top-1 in the Codalab leaderboard. These credits will be counted proportionally towards the final grade in the course.

For the both tasks, you are expected to provide:

1. **Technical report (10 points total).** Write a report in the provided Ipynb template[10] describing the method used in your solution. The report must have two parts:
    a. **Methodology (5 points)**: the main of your report with description of all methods that you tried and, most importantly, that worked the best for you. Here you can include some tricks of your preprocessing, description of the models and motivation of their usage, the description of the training process details (train-test split, cross-validation, etc.). So, everything valuable that will help us to understand the scope of your work and reproduce your pipeline.
    b. **Discussion of results (5 points)**: here we want to see the final table with comparison of the baseline and all tried approaches you decided to report. Even if some method did not bring you to the top of the leaderboard, you should nevertheless indicate this result and a discussion, why, in your opinion, some approach worked and another failed. Interesting findings in the discussion will be a plus.

2. **Code (10 points total)**. Develop yourself a solution of the task and provide a reproducible code in the provided template. Make sure that your code:
    a. Is using Python 3;
    b. Contains code for installation of all dependencies;
    c. Contains code for downloading of all the datasets used;
    d. Contains the code for reproducing your results (in other words, if a tester downloads your notebook she should be able to run cell-by-cell the code and obtain your experimental results).

    As a result, you code will be graded according to these criteria:
    a. **Readability (5 points)**: your code should be well-structured preferably with indicated parts of your approach (Preprocessing, Model training, Evaluation, etc.).
    b. **Reproducibility (5 points)**: your code should be reproduced without any mistakes with “Run all” mode (obtaining experimental part).

[10] https://colab.research.google.com/drive/1IyvayaR7KS9JyofdQ14b8J3kwJ8QCLrj?usp=sharing

- **Results (5 points + extra 5 or 10 points):** Push the (best) solutions which you developed to the **CodaLab** platform so that they appear in the respective public leaderboard. The name of your user / submission should be present in the report for verification.
    - You will get **5 points for outperforming the baseline**; then **additional 5 points for being in top 20%** at the public leaderboard on the private dataset OR **additional 10 points for being top 1** at the private leaderboard.

Additional notes:

Please follow these rules:

1. You should work on the model on your own and submit your own solution.
2. Use of data:
    a. The only labelled data you are allowed to use is the data provided in the CodaLab competition.
    b. You can use any unlabelled datasets you need, provided that they are open. You should specify the additional data you use in the model description (in CodaLab) and in the report.
3. In order to get the full mark you should submit your solution before the deadline. The solutions submitted after the deadline will also be checked, but only if they significantly outperform the baseline.