Final project

Natural Language Processing &Health
Due: April 10th, 2022

Students may do final projects solo or in teams of up to 3 people. We strongly recommend you do the final project in a team.

1 Choosing a project topic

1.1 Project Suitability

You can choose any topic related to machine learning for biomedical NLP or clinical NLP. That means your project should make substantive use of machine learning and substantive use of biomedical or clinical language data.

1.2 Project types

Here is a (non-exhaustive!) list of possible project types:

- Applying an existing machine learning model to a new task
- Implementing a complex system
- Proposing a model (or a new variation of an existing model)
- Experimental and/or theoretical analysis of machine learning models

1.3 Project ideas

- Relation extraction
- Named entity recognition
- Sentence similarity
- Text classification
- De-identification

1.4 Expectations

Your project should provide some kind of scientific knowledge gain, similar to typical BioNLP research papers. A typical case is that your project will show that your proposed method provides good results on a BioNLP task you are dealing with.

Given that you only have a few weeks to work on your project, it is not necessary that your method beats the state-of-the-art performance or works better than previous methods. But it should at least show performance broadly expected of the kinds of methods you're using.

In any case, your paper should try to provide reasoning explaining the behavior of your model. You will need to provide some qualitative analysis, which will be useful for supporting or testing your explanations. This will be particularly important when your method is not working as well as expected.

Ultimately, your project will be graded holistically, taking into account many criteria: originality, the performance of your methods, the complexity of the techniques you used, thoroughness of your evaluation, amount of work put into the project, analysis quality, writeup quality, etc.

1.5 Finding existing research

Generally, it's much easier to define your project if there is existing published research using the same or similar task, dataset, approaches, and/or evaluation metrics. Identifying existing relevant research (and even existing code) will ultimately save you time, as it will provide a blueprint of how you might sensibly approach the project. There are

many ways to find relevant research papers:

- You could browse recent publications at any of the top venues where BioNLP and/or clinical NLP is published: BioNLP workshop, clinical NLP workshop, ACL, EMNLP, TACL, NAACL, EACL, NIPS, ICLR, ICML (Not exhaustive!)
- In particular, publications at many BioNLP venues are indexed a
  - `https://aclweb.org/aclwiki/BioNLP_Workshop`
  - `https://clinical-nlp.github.io/2020/`

1.6 Finding datasets and tasks

There are lots of publicly-available datasets on the web. Here are some useful resources to find datasets

- 5 different tasks: `https://github.com/ncbi-nlp/BLUE_Benchmark`
- n2c2 NLP research data sets: https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/
- Chemical disease relation tasks: `https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr/`
- Disease recognition: `https://www.ncbi.nlm.nih.gov/research/bionlp/Data/disease/`
- Chemicals recognition: `https://ftp.ncbi.nlm.nih.gov/pub/lu/NLMChem/`
- Repository of biomedical corpora: `http://corpora.informatik.hu-berlin.de/`

1.7 Collecting your own data

It is possible to collect your own data for your project. However, data collection is often a time-consuming and messy process that is more difficult than it appears. Given the limited time frame, we generally don't recommend collecting your own data. If you really do want to collect your own data, make sure to budget the data collection time for your project. Remember, your project must have a substantial machine learning component, so if you spend all your time on data collection and none on building the models, we can't give you a good grade.

2 Project advice

2.1 Define your goals

At the very beginning of your project, it's important to clearly define your goals in your mind. In particular:

- **Clearly define the task.** What's the input, and what's the output? Can you give an example? If the task can't be framed as input and output, what exactly are you trying to achieve?
- **What dataset(s) will you use?** Is that dataset already organized into the input and output sections described above? If not, what's your plan to obtain the data in the format that you need?
- **What is your evaluation metric (or metrics)?** This needs to be a well-defined, numerical metric (e.g., F1 score), not a vague idea (e.g., 'summary quality').
- **What does success look like for your project?** For your chosen evaluation metrics, what numbers represent expected performance, based on previous research? If you're doing an analysis or theoretical project, define your hypothesis and figure out how your experiments will confirm or negate your hypothesis.

2.2 Processing data

You may need to do (additional) processing of your data (e.g., tokenization, pos tagging, or parsing). Here are some tools that may be useful:

- **Stanford core NLP**: a Python library providing tokenization, tagging, parsing, and other capabilities. `https://stanfordnlp.github.io/stanfordnlp/`
- **NLTK**, a lightweight Natural Language Toolkit package in Python: `http://nltk.org/`
- **spaCy**, another Python package that can do preprocessing, but also includes neural models (e.g., Language Models): `https://spacy.io/`

- Other software used in the lab.

2.3 Data hygiene

At the beginning of your project, split your data set into training data (most of your data), development data (also known as validation data) and test data. A typical train/dev/test split might be 70%, 10%, 20% (assigned randomly). Many BioNLP datasets come with predefined splits. If you want to compare against existing work on the same dataset, you should use the same split as used in that work. Here is how you should use these data splits in your project:

- **Training data**: Use this (and only this data!) to train/implement your model.
- **Development data**: This has two main uses.
  - The first is to compare the performance of your different models (or versions of the same model) by computing the evaluation metric on the development data. This enables you to choose the best hyperparameters and/or architectural choices that should be evaluated on the test data.
  - The second important usage of development data is to decide when to stop training your model. Two simple and common methods for deciding when to stop training are:
    * Every epoch (or every N training iterations, where N is predefined), record performance of the current model on the development set and store the current model. If development performance is worse than on the last previous iteration (alternatively, if it fails to beat the best performance M times in a row, where M is predefined), stop training and keep the best model.
    * Train for E epochs (where E is some predefined number) and, after each epoch, record the performance of the current model on the development set and store the current model as a checkpoint. Once the E epochs are finished, stop training and keep the best model.
- **Test data**: At the end of your project, evaluate your best-trained model(s) on the test data to compute your final performance metric. To be scientifically honest, you should only use the training and development data to select which models to evaluate on the test set.

The reason we use data splits is to avoid overfitting. If you simply selected the model that does best on your training set, then you wouldn't know how well your model would perform on new samples of data – you'd be overfitting to the training set. In NLP, powerful neural models are particularly prone to overfitting to their training data, which is especially important.

Similarly, if you look at the test set before you've chosen your final architecture and hyperparameters, that might impact your decisions and lead your project to overfit the test data. Thus, in the interest of science, it is extremely important that you don't touch the test set until the very end of your project. This will ensure that the quantitative performance that you report will be an honest, unbiased estimate of how your method will do on new samples of data.

2.4 Build strong baselines

A baseline is a simpler method to compare your more complex system against. Baselines are important so that we can understand the performance of our systems in context. For example, suppose you're building a machine learning model to do binary text classification (classification of sentences as positive or negative). The simplest baseline is the guessing baseline, which would achieve 50% accuracy (assuming the dataset is 50% positive and 50% negative). A more complex baseline would be a simple Naive Bayes classifier. You could also have simple neural baselines – for example, encoding the sentence using an average of word embeddings. Lastly, you should compare against simpler versions of your full model. Building strong baselines is very important. Too often, researchers and practitioners fall into the trap of making baselines that are too weak, or failing to define any baselines at all. In this case, we cannot tell whether our complex systems are adding any value at all. Sometimes, strong baselines perform much better than you expected, and this is important to know.

2.5 Evaluation

In your project, carrying out meaningful evaluation is as important as designing and building your neural models. Meaningful evaluation means that you should carefully compare the performance of your methods using appropriate evaluation metrics.

2.5.1 Quantitative evaluation - numerical performance measures

Choosing evaluation metrics. You must have at least one evaluation metric (which should be a numerical metric that can be automatically computed) to measure the performance of your methods. If there is existing published work on the same dataset and/or task, you should use the same metric(s) as that work (though you can evaluate on additional metrics if you think it's useful).

What to compare. You should use your evaluation metric(s) to (a) compare your model against your baselines, (b) compare different versions of your model, and (c) compare your model against previous work (extra points). When comparing against previous work, make sure to get the details right – for example, did the previous work compute the BLEU metric in a case-sensitive or case-insensitive way? If you calculate your evaluation metric differently from previous work, the numbers are not comparable!

2.5.2 Qualitative evaluation

The qualitative evaluation seeks to understand your system (how it works, when it succeeds, and when it fails) by measuring or inspecting key characteristics or outputs of your model. You will be expected to include some qualitative evaluation in your final report. Here are some types of qualitative evaluation:

- A simple kind of qualitative evaluation is to include some examples (e.g., input and model output) in your report. However, don't just provide random examples without comment – find interesting examples that support your paper's overall arguments, and comment on them.
- Error analysis is another important type of qualitative evaluation. Try to identify categories of errors.
- Break down the performance metric by some criteria. For example, if you think a text classification model is especially bad at classifying short sentences, show that by plotting the F1 score as a function of source sentence length.
- Compare the performance of two systems beyond the single evaluation metric number. For example, what examples does your model get right that the baseline gets wrong, and vice versa? Can these examples be characterized by some quality? If so, substantiate that claim by measuring or plotting the quality.
- If your method is successful, qualitative evaluation is important to understand the reason behind the numbers, and identify areas for improvement. If your method is unsuccessful, qualitative evaluation is even more important to understand what went wrong.

3 Grading sheet

- Introduction
  - Project definition (5%)
  - Related work (5%)
- Material and methods
  - Dataset(s)
    - Data description (5%)
    - Data split (5%)
  - Methods
    - baseline (10%)
    - Your system (20%)
- Results
  - Compare your model against baseline (10%).
  - Compare different versions of your model (10%).

  - Compare your model against previous work (extra 10%).
- Discussion Qualitative (20%)
- Conclusion (10%)
- Data and model sharing (extra 10%)

4 Preparation

We generally expect it to be **4** pages long for teams of 2 people (6 pages for teams of 3 people), including all tables and figures, but not including the references list at the end.

4.1 Formatting requirements

Templates: AMIA-Submission-Template.zip

The submission is

- A PDF file.
- A single column formatted document.
- Adherent with the page length restrictions. Please note that the page limit includes all tables and figures.
- Formatted for U.S. Letter (8.5 x 11 inch) paper size with one-inch margins left, right, top, and bottom.

And, the text within the Submission is formatted as follows:

- All submission rquired 1 line spacing. Must be no more than six lines per vertical inch.
- Title is 14-point bold, centered, title case (using initial capitals for each word in the title other than articles
- Below the title, are the names of the author(s), using 12-point Times New Roman typeface, single column, bold, centered, upper and lower case using appropriate capitals;
- The main text of the submission is single-spaced in 10-point Times New Roman typeface, justified, one-column format;

4.2 Reference format

References must be included in the PDF document. Links to web pages will not be accepted.

Bibliography and references must follow the **Vancouver Style** (`https://en.wikipedia.org/wiki/Vancouver_system`). Cite all references in the text, tables, or figure legends, using the following reference format: in the text, use eight-point superscript to indicate reference numbers; ten-point numbers in square brackets is an acceptable, although not preferred, alternative.

Under a heading "References" at the end of the submission, provide a list of references cited, in order of occurrence in the manuscript, and with titles using initial capital only. References must fit within the allotted page(s) for the respective submission categories. List all authors of any cited work when there are six or fewer authors; for more than six, list only the first three followed by "et al."

- References are listed in numerical order, and in the same order in which they are cited in text.
- A number is allocated to a source in the order in which it is cited in the text. If the source is referred to again, the same number is used.
- Use Arabic numerals (1,2,3,4,5,6,7,8,9)
- Use superscripts. For example, ...as one author has put it "the darkest days were still ahead".[1]
- Reference numbers should be inserted to the left or inside of colons and semi-colons.
- Reference numbers are generally placed outside or after full stops and commas

Example of a reference list

1. O'Campo P, Dunn JR, editors. Rethinking social epidemiology: towards a science of change. Dordrecht: Springer; 2012. 348 p.
2. Schiraldi GR. Post-traumatic stress disorder sourcebook: a guide to healing, recovery, and growth [Internet].

New York: McGraw-Hill; 2000 [cited 2019 Nov 6]. 446 p.
3. Halpen-Felsher BL, Morrell HE. Preventing and reducing tobacco use. In: Berlan ED, Bravender T, editors. Adolescent medicine today: a guide to caring for the adolescent patient [Internet]. Singapore: World Scientific Publishing Co.; 2012 [cited 2019 Nov 3]. Chapter 18.
4. Stockhausen L, Turale S. An explorative study of Australian nursing scholars and contemporary scholarship. J Nurs Scholarsh [Internet]. 2011 Mar [cited 2019 Feb 19];43(1):89-96.
5. Kanneganti P, Harris JD, Brophy RH, Carey JL, Lattermann C, Flanigan DC. The effect of smoking on ligament and cartilage surgery in the knee: a systematic review. Am J Sports Med [Internet]. 2012 Dec [cited 2019 Feb 19];40(12):2872-8.
6. Subbarao M. Tough cases in carotid stenting [DVD]. Woodbury (CT): Cine-Med, Inc.; 2003. 1 DVD: sound, colour, 4 3/4 in.
7. Stem cells in the brain [television broadcast]. Catalyst. Sydney: ABC; 2009 Jun 25.

4.3 Article structure

Divide your article into clearly defined and numbered sections. Subsections should be numbered 1.1 (then 1.1.1, 1.1.2, ...), 1.2, etc. (the abstract is not included in section numbering). Use this numbering also for internal cross-referencing: do not just refer to 'the text'. Any subsection may be given a brief heading. Each heading should appear on its own separate line.

Abstract

The abstract is a brief summary of the paper. It is typically one paragraph long, and is a concise summary of what was done and the principal results. It may be assumed that the reader has some knowledge of the subject, but the abstract should be intelligible without reference to the paper. Don't cite sections, tables, or figures in the abstract. The title of the paper is part of the abstract, so the opening sentence should be framed without repetition of the title. Write the abstract after you have written the rest of the paper; then you know what the paper claims to do and does.

1. Introduction

State the objectives of the work and provide an adequate background, avoiding a detailed literature survey or a summary of the results.

2. Material and methods

2.1 Dataset

Provide description of a dataset or a group of datasets. It contains facts about data, such as the number of documents, sentences, tokens in the training and test sets. Put the data in tabular form if appropriate.

2.2 Methods

Provide sufficient details to allow the work to be reproduced by an independent researcher. Methods that are already published should be summarized, and indicated by a reference. If quoting directly from a previously published method, use quotation marks and also cite the source. Any modifications to existing methods should also be described.

2.3 Experimental settings

Describe how the experiment was done and summarize the data taken. Describe the evaluation metrics, software, hardware, hyperparameters.

3. Results

In this section you should present your results. Results should be clear and concise.

4. Discussion

This is where you interpret your results. This should explore the significance of the results of the work, not repeat them. A combined Results and Discussion section is often appropriate. Avoid extensive citations and discussion of

published literature.

5. Conclusions

A short Conclusions section should follow the Discussion section. The Conclusions should present the main conclusions of the study and should clearly summarize the range of applicability of the methodology described in the article.

6. References