LING530F: Deep Learning for NLP
Assign. 2: (Text) Classification With RNNs
Format: Group Assignment
Due Date: 24-Oct, 2018
Weight: 20% of grade

Instructor: Muhammad Abdul-Mageed

1 Submission Instructions

General: Please take your time reading this assignment carefully, ensuring you understand it clearly. If there is any part that is not clear to you, please make sure you ask the instructor.

Identification: Please make sure your name, name of the assignment, and the course, are clearly marked in your Jupyter Notebook Submission as well as on the short paper pdf and in the LaTex.

1.1 Method of Submission & Deliverables

Please submit the following via Canvas by the deadline.:

1. A short paper (4 pages main body+ 2 pages references) in pdf format
2. The LaTex code (the ".tex" file) used for writing up the short paper in # 1 above. (See Section 1.2 below).
3. All your code in a single Jupyter notebook, including sample outputs where appropriate

1.2 LaTex Requirement

LaTex is a document preparation system that with a host of attractive features, as you may read for yourself through this link. It is a good idea to use overleaf, the "Collaborative Writing and Publishing" platform. LaTex beginners can check the video and slide tutroials available here. You can find many other LaTex tutorials online.

1.2.1 ACL 2018 LaTex Template

It is required to use the ACL 2018 LaTex Template. You can download the acl18-latex.zip (please use the updated acl_natbib.bst). Alternatively, you can directly access the Template via Overleaf here.

1.3 References & Credit

Please make sure you cite all your references clearly. This includes any papers you review, any tutorials you benefit from, any code you re-use or modify, etc. It is required to categorically and explicitly cite any material created by others that you consult. **Failing to abide by this crucial requirement will be treated as plagiarism**.

2 Objectives

The **(Text) Classification With RNNs** is designed to offer students opportunities to:

- **(1)** Acquire practical experience *designing* and *training* a fully-fledged, supervised deep learning classification system. This includes an extended list of important decisions (e.g., data splits, network architecture, regularization) crucial to understanding how deep learning works;
- **(2)** Familiarize themselves with related software (e.g., PyTorch), with an eye to eventually master use and creation of deep learning software;
- **(3)** Practice writing high-quality research-level papers in an environment identical to top academic conferences in the field.

3 Assignment

3.1 Background

3.1.1 Group Assignment

This assignment is a group project where each student will work with two other students. [1] Each group member is expected to contribute equally to the assignment. Students will be asked to submit a section in the paper Appendix detailing who did what. This will be taken into account during grading. Additionally, in cases where a group member do not his/her part of the work seriously or do not end up contribute as initially agreed, students will be given the option of letting the instructor know by sending an email to `muhammad.mageed@ubc.ca`. This is expected to happen only in very rare cases. In the great majority of the cases, all students are hard-working, passionate, and well-motivated. For these reasons, the instructor trusts this will be the case in this and other assignments. Students are also encouraged to form group with complementary skill sets. **Importantly, all group members are expected and must contribute to the engineering involved. It is not sufficient, for example, for a student to only contribute to the writing.**

3.1.2 Overview

For this assignment, students will build a fully-fledged (text) classifier using RNNs any of the variations of RNNs. This includes, e.g., Long-Short Term Memory Networks (LSTMs) or Gated Recurrent Units (GRUs). Students are welcome to go beyond this requirement by using mechanisms on top of these baselines. For example, students can (but are not required to) use an *attention* mechanism on one of the network layers. For this assignment, the students will be provided a host of tasks (each accompanied with a labeled text dataset) from which they can

[1] Groups of size larger than or less than 3 students can be considered, *conditioned on prior instructor permission*.

choose one task. Students interested in extending their work beyond their task of choice will be allowed 1 more page for describing this extension. (Students are not required to extend their work beyond their single task of choice, but might find this interesting and so this is allowed). Students will then write a short paper describing their work, delivering (1, 2) both the pdf and the LaTex ".tex" documents, along with (3) all their code in a single Jupyter notebook.

4 Tasks

Please choose one task. As a rule, students should consult with the webpage of each task for complete information and details. The information provided here is only partial, to avoid overload and duplication. Also, only partial description of Task 1 is provided here. All tasks are described in the respective tasks homepages and all links are provided here. Note that "test" sets for the 2019 tasks are not available. Also, the "development" for some of these 2019 tasks might not have labels. In these cases, you can just take "training data and split into 80% "train", 10% "dev", and 10% "test" to perform your work. You are encouraged to subscribe to the email group related to the task you will work on, if there is one (check online for more info.).

4.1 Task 1: Detecting 'Implicit Emotions'

This task was part of an EMNLP workshop as described here. The task aims at `‘‘developing models which can classify a text into one of the following emotions: *Anger, Fear, Sadness, Joy, Surprise, Disgust* without having access to an explicit mention of an emotion word."`

4.1.1 Task Description

This is copied and pasted from the task page:

Participants were given a tweet from which a certain emotion word is removed. That word is one of the following: "sad", "happy", "disgusted", "surprised", "angry", "afraid" or a synonym of one of them. The task was to predict the emotion the excluded word expresses: Sadness, Joy, Disgust, Surprise, Anger, or Fear.

With this formulation of the task, we provide data instances which are likely to express an emotion. However, the emotion needs to be inferred from the causal description, which is typically more implicit than an emotion word. We therefore presume that successful systems will take into account world knowledge in a structured or statistical manner. Examples are:

- "It's [#TARGETWORD#] when you feel like you are invisible to others."
- "My step mom got so [#TARGETWORD#] when she came home from work and saw that the boys didn't come to Austin with me."
- "We are so [#TARGETWORD#] that people must think we are on good drugs or just really good actors."

The shared task consisted of the challenge to build a model which recognizes that [#TARGETWORD#] corresponds to sadness ("sad") in the first two examples and with joy ("happy") in the third.

4.1.2 Dataset

The dataset for this task is available under IEST-2018 folder. Importantly, the "dev.csv" file has fake labels and you must replace these labels with the true labels from the file "trial-v3.labels". A sample paper exploiting this dataset is [1].

4.2 Task 2: EmoContext

This task asks you to classify the emotion of the utterance as one of the classes: "happy", "sad", "angry" or "others" with a given textual dialogue. This is a link to the task homepage. This is a link to the data. Note, no labels exist on "development" set, and so you can split the "training" set into 80% "train", 10% "dev", and 10% "test".

4.3 Task 3: Hyperpartisan News Detection

This task requires participants to determine whether a news article follows a "hyperpartisan argumentation, i.e., whether it exhibits blind, prejudiced, or unreasoning allegiance to one party, faction, cause, or person". This is the homepage for this task. Data for this task are available here. You should consult the homepage for details, but we have provided the data on the link above.

4.4 Task 4: RumourEval

This task aims at detecting the veracity of rumours and has two sub-tasks. *Task A:* requires classifying responses of a rumour according to stance. i.e. support, deny, query, and comment:

- **Support:** the author of the response supports the veracity of the rumour they are responding to.
- **Deny:** the author of the response denies the veracity of the rumour they are responding to.
- **Query:** the author of the response asks for additional evidence in relation to the veracity of the rumour they are responding to.
- **Comment:** the author of the response makes their own comment without a clear contribution to assessing the veracity of the rumour they are responding to.

Task B: "The goal of the second subtask is to predict the veracity of a given rumour. The rumour is presented as a post reporting or querying a claim but deemed unsubstantiated at the time of release. Given such a claim, and a set of other resources provided, systems should return a label describing the anticipated veracity of the rumour as true or false. The ground truth of this task is manually established by journalist and expert members of the team who identify official statements or other trustworthy sources of evidence that resolve the veracity of the given rumour." Please read online for more details.

4.5 Task 4: Arabic Dialect Identification

This is the homepage for this task. These are the different data splits: train, dev, and test. There are multiple tasks. Look at this paper for details [2].

4.6 Task 5: Open Task

If you would like to work on a task different from the ones listed above, please feel from to discuss with the instructor. It will be possible to provide permission for work on a task that involves (1) supervised classification with RNNs (or other deep learning methods) on (2) language or language-related data (e.g., speech signal, image and text). It is crucial that students are able to show that (3) they are working on this task only now and have not finished this work before taking this course and that (4) they have the required data. The instructor reserves the rights to ask detailed questions that ensure these four conditions apply, but any other conditions that might seem relevant or necessary.

5 Extending The Work

Students can optionally decide to work on additional task. Students are allowed to do so and are provided an extra page for each additional task. The instructor does not suggest taking on additional tasks if the extra work will need more than $3-4$ hours. Students can also decide to extend beyond variations of RNNs (e.g., work with CNNs or other methods like [variational] auto-encoders or generative adversarial networks [GANs]). If students decide to extend methods in this way, they will be allowed an extra page. In general, no more than 5 pages will be allowed. This requirement will be strictly enforced. Additional information that are necessary, can be added in the Appendixes, but the instructor does not guarantee reading Appendixes. This is similar to ACL not reacquiring reviewers to read Appendixes (although reviewers are encouraged to do).

6 Tools

It is strongly preferred to use PyTorch. There are a number of PyTorch tutorials here. You may like to benefit from AllenNLP. Learning and using PyTorch will help you understand deep learning better, and will enable you build research-level, powerful models on the longer run. Resources from last assignment are still relevant as enabling tools:

- Gensim word2vec tutorial, usage examples, and parallelization post(advanced).
- fastText GitHub page, tutorials, and frequently asked questions.

7 Useful Literature

In general, the following are good papers you can consult: [3, 4, 5].

8 Paper Writing

Your paper can have some or all of the following sections.

- Abstract
- Introduction
- Related Work
- Dataset
- Methods
- Experiments
- Results
- Conclusion

For the final assignment, the instructor will provide students some details about how they can write each section. For this short paper, students cannot include all these sections. So, a better way would be to include the most important among them such that the work is easily replicable. Most importantly, students will need to demonstrate understanding of the theoretical part of the course. For example, a detailed description of pre-processing, data splits (concepts like train, dev, and test sets and/or n-fold cross validation), model capacity (e.g., the architecture of the network used in terms of layers and number of units in each layer), activation functions, regularization, cost functions, baselines and meaningful comparisons (which baselines are chosen and why), etc. are very crucial. Justification of each of the important aspects of experiments is significant.

9 Grading

For this assignment, students will deliver a Jupyter notebook with all code, a pdf and the LaTex document used to Canvas as explained in Section 1.1. The grade will be based on completion of the work, its technical quality, clear explanation and justification of the details, the extent to which the work reflects understanding of deep learning and NLP in as much as the task is concerned, and the extent to which the paper follows norms of acadmic writing in deep learning and NLP. An "A" grade is warranted for organized and complete code that yields a competitive system (in the top 12-15% of all class submissions), showing an understanding of the task, theoretical knowledge, and mastery of the engineering involved.

References

[1] Hassan Alhuzali, Mohamed Elaraby, and Muhammad Abdul-Mageed. 2018. Ubc-nlp at iest 2018: Learning implicit emotion with an ensemble of language models .

[2] Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. pages 263–274.

[3] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. pages 6294–6305.

[4] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. volume 1, pages 2227–2237.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training .