CS 6740/IS 6300 Fall 2019 - assignment A3

Learning goals

1. experiment with the "inoculation method" from the Liu, Schwartz, and Smith NAACL 2019 paper.
2. practice playing around with a new research idea in an open-ended way
3. gain experience with some NLP task using data meant to be challenging
4. learn from how other students in the class approaches the same task

... under the constraint that I'd like you to be able to accomplish these goals without having to build up a lot of stuff from scratch, and **you should not spend more than 20-23 hours total over two weeks**.

What to do

We agreed to use the Build It Break it sentiment data. The sentence training data is to be used to create the two 80-10-10 "original" train/dev /test splits, with train and test preserving the original distribution (reported to be 3310 negative, 3610 positive). The concatenation of ~~the four test ground truth data sets~~ Team1, Team2's and Team3's (warning - contains some unconventional characters in four lines) ground truth is to be used as the challenge data, with a single ~~10-90~~ train-test split. (Liu et al want the challenge training to be small.) The splits and explanation of which original split **you, personally** should use is in the data readme.

The basic idea is to try experiment with inoculation-by-fine-tuning on this "original data"/"challenge data" pair; in your final report, you should include a training "curve" like in Figure 3 of the paper, but you should only pick 3 sizes for the fine-tuning examples. (Choose a reasonable set of sizes. An example of a bad choice would be {5, 7, 9} — you'd want your choices to cover a larger span. Would {25, 75, 100} be a good choice? {10, 100, 143 = all}?)

Choose **one** of the following two questions to investigate, or state your own question about the inoculation-by-fine-tuning methodology and investigate it.

Question A: Consider "Possible Outcome (2)" from Fig 1 of the paper, where there is what we might call a "model weakness" gap. Will a "good" algorithm yield less of a model weakness gap than a "not as good (although not completely terrible)" algorithm?

> Recall that we discussed in class that one possibility you can use for an algorithm class to work with is the NLTK Naive-Bayes text classifier with a choice of top-k most-common words as features; the code we saw in lecture was from Chapter 6 "learning to classify text" from Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit (for Python 3, a Python 2 version is here).

You'll want to install the NLTK package; instructions here.

You'll want to massage the data to be a list of pairs (2-tuples), where each 2-tuple has a list of words as the first element and the name of the class label for the sentence as the second element. For example:

```
      train_docs = [ (['this', 'is', 'awesome'], 'pos'), (['this', 'is', 'terrible' ], 'neg'))
```

To make a set of features functions corresponding to the top 345 (say) most frequent words in your training data, first, get a list word_features_345 of the top 345 most frequent words. Then, you can define

Here is a link to a page where you can view "diffs" between any two versions: use the "compare selected versions" feature to highlight precisely what text was added or deleted.

Version	Published	Changed By	Comment
CURRENT (v. 9)	**Oct 27, 2019 13:23**	**Lillian Lee**	correct one examp
v. 8	Oct 25, 2019 13:14	**Lillian Lee**	post the splits info
v. 7	Oct 25, 2019 13:03	**Lillian Lee**	placeholder of spli
v. 6	Oct 25, 2019 10:49	**Lillian Lee**	Discarding Team
v. 5	Oct 25, 2019 01:47	**Lillian Lee**	placeholder for sp
v. 4	Oct 25, 2019 01:07	**Lillian Lee**	mention CMS dea
v. 3	Oct 25, 2019 01:01	**Lillian Lee**	make change colu
v. 2	Oct 25, 2019 00:52	**Lillian Lee**	specify sentence,
v. 1	Oct 25, 2019 00:47	**Lillian Lee**	

```
def document_features(document):
        document_words = set(document)
         features = {}
         for word in word_features_345:
                 features['contains({})'.format
(word)] = (word in document_words)
         return features
```

Then, you can do (assuming you want a NaiveBayes classifier that uses the top 345 words as features):

```
featuresets = [(document_features(d), c) for (d,
c) in train_docs]
classifier = nltk.NaiveBayesClassifier.train
(train_set)
```

then, assuming test_set is in the right format, too, you can do

```
>>> print(nltk.classify.accuracy(classifier,
test_set))
# some numerical result comes out
 >>> classifier.show_most_informative_features
(5)

# 5 most informative features are printed
```

> But we also decided you could instead use a neural classifier against another neural classifier (taking care to think about the pre-training issue — how does that affect the use of inoculation as studying a challenge dataset?), or an "old-style' classifier against a neural classifier, or what have you. The choice is up to you, just try to pick something sensible, and justify your choices in the writeup!

Question B: How much difference does it make if instead of adding new challenge data, you added the same amount of original data (which would need to be held out) as a "pseudo-challenge dataset"?

Discussion and collaboration rules

You are encouraged to discuss and ask questions or even just post fun examples you discovered in the data on CampusWire , but you should do the experimental work on your own.

Deadlines

- Monday November 4, 11:59pm: submit an informal progress report on CMS. Include in it any questions you might have for me. The intent is just to give me something to skim before meeting with you, so bulleted lists and some sort of preliminary performance numbers would be fine. (You don't have to submit the "final_report.pdf" yet; I'll give extensions to allow the upload of it later on.)
- Tuesday November 5, in class: individual ~9-minute meetings with me (will be scheduled for the usual lecture time) to discuss your progress report.
- Thursday November 7, in class: present your findings of interest so far to the class (5 minutes of presentation per student, 5 minutes for questions). Use your judgment regarding "interesting", noting that negative results ("no matter what I tried, I didn't see a generalization gap") can be of interest. You can either use slides (on your own laptop, or you can send them to me beforehand to display) or make handouts. but I do require some visual aid (how else will you be able to show learning "curves", for instance?).

- Friday, November 8, 11:59 pm: submit to CMS your final report of what you did, what design choices you made and why, what you learned, and what more you might have wanted to try if more time were allotted. 4-6 pages is probably a good rough length guideline.

Grading criteria

- Demonstration of understanding the concept of inoculation-by-fine-tuning as described in the Liu et al. paper.
- Thoughtfulness in designing your experiments.
- Thoughtfulness in analyzing the results of your experiments. Note that thoughtfulness can be demonstrated by explaining why something you did turned out to be a bad idea
- Demonstration of good-faith effort in running the experiments, doing the assignment, and meeting the deadlines.

I will consider granting extra credit for (thoughtful) participation on CampusWire.