CMPUT 274 — Tangible Computing I **Version 1.1-20**
Weekly Exercise #5 — New Training Data

Introduction

The purpose of this exercise is to gain experience and intuition in creating training data for our classifier, and to understand the nature of the precision, recall, and accuracy metrics of goodness. Developing a high-quality training dataset can be a very challenging problem.

This Weekly Exercise is different from the other exercises because it involves writing no new code. You are to use the code in `ooclassifierbasev3.py`, as provided and discussed in lecture.

This Weekly Exercise can be viewed as a new, expanded version of the Worksheet for Lecture 15 at:

`https://docs.google.com/document/d/13fNwRa5GyI0BnYLL3OxnPNqocrrwGN1qfjEm_ocKriU/edit?usp=sharing`

There may be unspecified details for this exercise for which a **reasonable design decision** can be made (i.e., does **not** contradict an existing specification), stated explicitly (e.g., in a comment in the code or input data), and implemented.

This assignment will require you to know something about:

1. Machine-learned classifiers
2. Training Data
3. Precision
4. Recall
5. Accuracy

Task I: Create a New Training Dataset

As you have seen, the OO Classifier (`ooclassifierbasev3.py`) is capable of doing simple sentiment analysis on any arbitrary training dataset. So far, we have used the classifier with a "weather" and with a "help" dataset.

You are required to create a new training dataset of your own. Your dataset will be in a text file (just like the weather and help datasets), and must be compatible with `ooclassifierbasev3.py`. We will actually run your training data with that classifier, as part of the marking process.

Furthermore, your new training dataset must be in a text file called `new.training.txt` and:

1. **Positive training instances:** Contain at least 20 (and no more than 40) *positive training instances*.

   In the "weather" dataset, an example of a positive training instance is `#weather nice weather eh`.

2. **Negative training instances:** Contain at least 20 (and no more than 40) *negative training instances*.

In the "weather" dataset, an example of a negative training instance is `#negative good food`.

3. Every training instance must be properly labelled and have between 1 and 15 words.

   In the "weather" dataset, instance `#weather nice weather eh` has `#weather` as the label and the training instance has 3 words.

4. **Positive features:** The `pos-features` environment variable must be set correctly in your training dataset file. Choose your positive features (aka target words) appropriately to meet the requirements of the rest of this exercise (see below).

   In the "weather" dataset, the first three words of `pos-features` are `outside today weather`.

5. The `pos-label` environment variable must be set correctly in your training dataset file.

   In the "weather" dataset, `#weather` is the value of `pos-label`.

As a "Solo Effort", you must create your own training dataset **without collaboration or help** from anyone else, other than the instructor or an TA. You may use sentences or words from online sources, with citation (e.g., by providing an URL as a comment inside the training data file).

Task I is worth 40/100 of your mark for the exercise. An approximate rubric for this task is:

1. 35 to 40, out of 40: The training dataset meets all the requirements and shows an excellent variety of sentences and a wide variety of words.

   All training instances clearly capture the positive and negative sentiment of the dataset, whatever it might be (e.g., "get the emergency team" is **not** labelled as `#negative` for a "help" sentiment analysis dataset).

2. 20 to 34, out of 40: The training dataset meets all the requirements.

   But, there are several repetitions of similar ideas in several training instance (e.g., one instance is "get my wife" and another instance is "get my spouse", and other instances are simple variations of these instances). Or, there are multiple repetitions of the same words (e.g., multiple uses of the word "help"). Or, some of the training instances do not match the positive and negative sentiments of the dataset.

   Of course, the same ideas and words can be repeated. But if the repetition is seen in more than 20% of the training instances, it limits the variety and diversity of the training instances.

   By this standard, the "weather" and "help" datasets used in lecture would likely receive between 20 to 34, out of 40.

3. less than 20, out of 40: The training dataset does not meet all the requirements. And/or there are significant flaws with the variety and diversity of the training instances. And/or, some of the training instances clearly violate the positive and negative sentiments of the dataset.

Task II: Sentiment Analysis and Positive Features

Find a set of positive features that results in an accuracy of at least 0.70 on your training dataset when used with the classifier. Be sure to set the `pos-features` environment variable correctly in `new.training.txt`.

We will test your training dataset by running:

```
python3 ooclassifierbasev3.py new.training.txt
```

Read the Wikipedia page at:

https://en.m.wikipedia.org/wiki/Sentiment_analysis

In a text file `question.1.txt` (to be submitted) answer the following Question #1 (which has multiple parts) using 300 words or less. Use proper English sentences (no point form), proper grammar, and spelling. Write in well-structured and coherent paragraphs. Make sure your paragraphs have proper topic sentences.

Question #1:

In what way does your positive features reflect the sentiment analysis intended by your training dataset? How did you choose the positive features? What strategy did you use? Did you randomly try words? Did you have a more structured strategy? Explain briefly.

NOTE: You are not required to write code or an algorithm to answer this question.

Task II is worth 30/100 of your mark for the exercise. Although CMPUT 274 is not a course in essay writing, the quality of the writing does affect your mark. An approximate rubric for this task is:

1. 20 to 30, out of 30: The answer to the question is well-written, grammatically correct, and coherent. Furthermore, the answer articulates a clear strategy and demonstrates a full understanding of the relationship between classifiers, sentiment analysis, the training data, positive features, and accuracy.

   Specific positive features are referred to and discussed in the answer, and why they were selected, and why other possible positive features were not selected.

2. 10 to 19, out of 30: The answer to the question is understandable, but may suffer from grammar errors, or may be confusing.

   The explanation of the strategy is described with insufficient details and concrete examples to convince the instructor that there is a clear understanding of the problem of training dataset creation, and the selection of positive features.

3. less than 10, out of 30: There are significant problems with the grammar and/or coherence of the answer. And/or the explanation of the strategy is too vague or shows a lack of understanding of the problem of training dataset creation, and the selection of positive features.

Task III: Accuracy and Precision

Read the Wikipedia page at:

https://en.m.wikipedia.org/wiki/Precision_and_recall

Starting with your training data in `new.training.txt`, do **not** change the positive features, but add new “realistic” training instances until precision is above 0.90, but accuracy is below 0.65.

Submit this new training dataset in file `task.3.txt`. We will test your training dataset by running:

```
python3 ooclassifierbasev3.py task.3.txt
```

In a text file `question.2.txt` (to be submitted) answer the following Question #2 (which has multiple parts) using 300 words or less. Use proper English sentences (no point form), proper grammar, and spelling. Write in well-structured and coherent paragraphs. Make sure your paragraphs have proper topic sentences.

Question #2:

What was your strategy in creating the new training instances required for this task? What kind of training instances can increase precision but decrease accuracy? Explain briefly. In your explanation, explicitly refer to some of the new training instances.

NOTE: You are not required to write code or an algorithm to answer this question.

Task III is worth 30/100 of your mark for the exercise. Although CMPUT 274 is not a course in essay writing, the quality of the writing does affect your mark. The rubric for this task is the same as for Task II, but with respect to the strategy of adding new training instances and including an understanding of the relationship between precision and accuracy.

Submission Guidelines:

Submit all of the required files (and no other files) as **one** properly formed compressed archive called either `training.tar.gz`, or `training.tgz`, or `training.zip` (for full marks, please do **not** use `.rar`):

- when your archive is extracted, it should result in exactly *one directory* called `training` (use this exact name) with the following files in that directory:
- `new.training.txt`
- `task.3.txt`
- `question.1.txt`
- `question.2.txt`
- your `README` (use this exact name) conforms with the Code Submission Guidelines (Sections 1.1.1, and Section 2.3), in terms of your *personal identification information only*
- No other files should be submitted.

Note that your files and functions must be named **exactly** as specified above.

A new tool has been developed by the TAs to help check and validate the format of your `tar` or `zip` file *prior* to submission. To run it, you will need to download it into the VM, and place it in the same directory as your compressed archive (e.g., `training.zip`).

You can read detailed instructions and more explanation about this new tool in Submission Validator: Instructions (at the top of the Weekly Exercises tab), or run:

```
python3 submission_validator.py --help
```

after you have downloaded the script to see abbreviated instructions printed to the terminal.

If your submission passes this validation process, and all validation instructions have been followed properly, you will not lose any marks related to the format of your submission. (Of course, marks can still be deducted for correctness, design, and style reasons, but not for submission correctness.)

When your marked assignment is returned to you, there is a 7-day window to request the reconsideration of any aspect of the mark. After the window, we will only change a mark if there is a clear mistake on our part (e.g., incorrect arithmetic, incorrect recording of the mark). At any time during the term, you can request additional feedback on your submission.