CS6742: NLP and Social Interaction, Spring 2011
2/15/2011: Assignment A4, characterizing triggers

Introduction In this assignment, we build on the data gathered in A3 to address the question, "what triggers a specific type of response in a conversation?". Although we frame this as a prediction task, our main goal is to discover what characterizes these "stimulating" tweets.[1] Per class pre-requisites, this assignment assumes basic experience in training an off-the-shelf supervised learning algorithm.

Due dates: **10pm Sunday Feb 20th** and **10pm Wednesday Feb 23th**. Steps 1 through 5 described below need to be completed by **10pm Sunday Feb 20th**, and an **initial summary** of results must be submitted at this time (`A4initial_summary.pdf` on CMS assignment "A4 checkpoint"). On the lecture of Tuesday Feb 22th, we will jointly discuss these initial results and share ideas; be prepared to also (infomally) present any issues you encountered. You will then have the chance to improve your results on Steps 1 through 5 up until the deadline for Step 6. A **final summary** describing the results of Steps 1 through 5 (please clearly indicate updates to the initial summary) and of Step 6 must be submitted (`A4final_summary.pdf` on CMS assignment "A4 final"). by **10pm Wednesday Feb 23th**,. The results will be discussed in class on Thursday Feb 24th.

Submit the deliverables and the summaries through the course CMS: http://cms.csuglab.cornell.edu.

Step 1: Collect a set of negative examples Start this step early. You might also want to start working on the next steps before completing Step 1 (use a small sample to develop your code).

Using the technique developed in A3, collect a set of (initial tweet, reply tweet) pairs in which the replies do **not** exhibit the indicators (e.g., ":)") you are working with. No filtering based on the content of the initial tweet should be done. The size of this *negative set* should be the same size as *positive set* (i.e., the set of initial-reply tweets you collected in A3), and *negative set* should contain no duplicates.

Deliverables: Two plaintext files `A4initial.txt` and `A4reply.txt` containing the pairs in the negative set in the same format used in A3.

To be placed in your summary: Start by briefly describing the "specific type of response" you collected in A3, the indicators employed and what filtering was done to the tweets returned by the Search API [2]. Then explain how you gathered the negative set (what call to the search API, or what other component of the Twitter API).

Step 2: Data exploration Compare the *initial* tweets in the positive and negative sets. Address at least these four questions:

- What is the average initial-tweet length in the two sets?
- What is the average word length in the two sets?
- What words are much more frequent in the positive set than they are in the negative set?
- Are URLs more frequent in the the positive or in the negative set?

To be placed in your summary: Answers to these (and potentially other) questions.

[1] The grading will reflect this by rewarding effort in coming up with interesting features and analyzing their relative effectiveness, rather than the accuracy obtained on the prediction task.

[2] This can be taken from the A3.i summary and updated by explaining any additional filtering that you might have done for A3.ii.

(OVER)

Step 3: Baseline As a simple baseline we will use the following rule:

> **Echo-baseline rule:** given an (initial tweet, reply tweet) pair, if the initial tweet contains an indicator, then predict that the reply will also contain an indicator (i.e., it is in the positive set); otherwise predict that the reply will not contain an indicator.

Test how accurate this baseline is by computing:

a) the percentage of pairs in the positive set for which this rule (correctly) predicts that the reply is of the desired type

b) the percentage of pairs in the negative set for which this rule (incorrectly) predicts that the reply is of the desired type

c) the percentage of pairs in the combined positive and negative set in which the baseline correctly predicts the type of the reply.

To be placed in your summary: These percentages.

Step 4: Feature representation Represent each initial tweet as a feature vector (you can build on the observations made in Step 2). You are encouraged to explore more features, but here is a minimal required set of features to be used:

- most frequent 200 words,
- emoticons and twitter-specific lexicon,
- tweet length,
- binary feature indicating whether a tweet contains an URL or not.

Two of these features are sufficient for the first deadline.

To be placed in your summary: Description of the features you selected and how you represent them. Example: "Feature: top 200 words. Representation: binary value for each of the top 200 words (1 if word is in tweet, 0 if word is not in tweet)."

Step 5: Apply supervised classifier Train a supervised classifier for the task: given an initial tweet (more precisely, its vector representation[3]), predict whether its reply is going to be of the desired type or not (did the pair come from the positive or negative set).

Since the emphasis of this assignment is not on machine learning, we do not want you to spend time tweaking the specific algorithm and parameters. That is why we strongly recommend using the off-the-shelf Weka classifiers available at http://www.cs.waikato.ac.nz/ml/weka/, in particular the Naive Bayes or the J48 (C4.5) Decision Tree classifier. Apart from having a GUI that facilitates feature exploration, Weka also provides vector normalization, automatic train-test splitting, an easy way to ignore a subset of the features, and an automatic feature-selection filter (the last two will be useful in Step 6).[4]

[3] Make sure to normalize the feature vectors.

[4] You are free to use any other classification tool, but please send us an email beforehand.

Use 80% of your data for training and 20% for testing (80% "percentage split" in Weka).

Deliverable: A plaintext file `A4classifier_output.txt` containing the Weka classifier output.[5]

To be placed in your summary: Describe data pre-processing. Precision on the train and test set of the classifier of your choice.

Step 6: Analyze the effectiveness of the features In this final step, analyze the effectiveness of your features. The outputs of the Weka Decision Tree and Naive Bayes classifiers already provide insight into which features (or combination of features) are more useful. Which individual feature performs the best, and does it perform better than the baseline? What are the top performing features? Can you provide an interpretation of these results?

To be placed in your FINAL summary: Address these questions. Optional: anything else you would have liked to have tried.

[5] If you are using the GUI you can right-click on the result in the "Result list" and select "Save result buffer".