README for classify.py
COMS 4705-HW1 Stance Classification

Jing Qian (jq2282@columbia.edu)

September 26, 2019

Dear Instructor, I want to use **one late day** for this homework. Thank you!

1. Model Description

1.1. Best model and corresponding parameters

For topic "abortion", my best model is Ngrams model with SVM classifier, getting 62% average score in 5-fold cross validation. The best parameters I found for such model is: Ngrams combining unigram, bigram and trigrams; 500 features selected from Ngrams; use LinearSVC() and set loss to "hinge", C to 1 and max_iter to 1000, class_weight to None. In fact, the best parameters I found for Linear SVC() are all default values, I think that may be the reason that they are set to default. Although the combination of unigram, bigram and trigrams generates a lot of features, it turns out 500 features are good to go. I think it is because of the limited size of our dataset. More features tend to overfit, not perform well on test set.

The Other Feature model for "abortion" gets 61% average score in 5-fold cross validation. The best parameters I found for such model is: 500 features are selected; use LinearSVC() and set loss to "hinge", C to 1 and max_iter to 1000, class_weight to None.

For topic "gay rights", my best model is Ngrams model with Naive Bayes classifier, getting 63% average score in 5-fold cross validation. The best parameters I found for such model is: Ngrams combining unigram, bigram and trigrams; 50 features selected from Ngrams; use MultinomialNB() and set additive smoothing parameter to 1, which is the default value. Also, we could infer that the "gay rights" topic is more separable because it reaches higher accuracy and needs many fewer features.

The Other Feature model for "gay rights" also gets 63% average score in 5-fold cross validation. The best parameters I found for such model is: 50 features are selected; use LinearSVC() and set loss to "hinge", C to 1 and max_iter to 1000, class_weight to None.

1.2. Classifiers and features tried

1.2.1. Feature selection

For Ngrams, I used 500(or 50) features selected from the combination of unigram, bigram and trigram. For Other Feature, I added three feature types to Ngrams: repeated punctuation, part-of-speech tags and LIWC. I used 500(or 50) features selected from the combination of unigram, bigram, trigram, three LIWC features ('words_pronom', 'words_per_sen', 'words_over_6'), repeated punctuation (??, ??????, !!!, and ?!) and part-of-speech tags ('count_noun', 'count_verb', 'count_adj').

In the feature selection, I tried tuning the ngram range, which doesn't affect much. In the Other Feature model, I tried adding different combination of 6 LIWC features and the rest columns in the dataset and found the three LIWC features ('words_pronom', 'words_per_sen', 'words_over_6') would lead to the highest performance.

I found that in the "abortion" topic, the extracted ngrams features include a lot of numbers. So I tried removing all numbers in the count vectorization, which decreased the accuracy.

I tried feeding all the features extracted from Ngrams into the classifier directly and found that selecting k best features instead of using all features increases the model accuracy significantly. I think it is because of the noise carried by irrelavent features. In the script, I did parameter search for k.

1.2.2 Classifier selection

In this assignment, we are asked to use Naive Bayes and SVM classifiers. For the Naive Byes, I use MultinomialNB(), which is the most widely used Naive Bayes classifier for text classification. I also tried ComplementNB(), which is good for imbalanced dataset and BernoulliNB(), which is good for short document. I found MultinomialNB outperforms a liitle bit than the other two classifiers. The only feature needs to tune for MultinomialNB is the smoothing parameter *alpha*.

For SVM, I use LinearSVC(), which is slower and has more parameters to tune than Naive bayes, but much faster than other SVM classifiers. I also tried SVC() and SGDClassifier(). The latter two outperform LinearSVC() by 1 2% accuracy for some parameters I tried. However, these two took much longer runtime than LinearSVC() so I did not do the full parameter search for these two classifiers but use LinearSVC() instead. In SVC(), I tried two kernels: "linear" and "rbf" (Gaussian). Kernel "rbf" did a little better than "linear" kernel in SVC(). Considering the efficiency, I use LinearSVC() directly which corresponds to the linear kernel in SVC() but more flexible and quicker to tune.

2. Error Analysis

2.1.Abortion

Three examples that did not work in my best model:

1. ID: C1417. Text: "One of my friends says she 's pro-abortion , and I pretty much agree . I 'm not for a choice , though I " m for that too , but I " m also for abortion in the way that it should be used more frequently ." It is labelled "pro" but wrongly predicted as "con". I think the classifier made a mistake because it has a "I'm not" before an important feature "choice". We know that pro-abortion people are usually called pro-choice, then the opposite of pro-choice is a negative attitude towards abortion.

2. ID: C1569. Text: "When a fetus hasnt been born yet it isnt a person to me . If somethign can not live without something for it to hook up to for life support it isnt alive . It isnt alive until it is born . It could be way more hazardous to the mother than the child . " It is labelled "pro" but wrongly predicted as "con". I think this text is filled with negative words, which mislead the classifier to think it is negative.

3. ID: C53. Text: "Palinin , Help me with something if you can . As I read back through this debate something odd jumped out at me . You dismiss and ignore every point I raise and then expect me to answer every point you raise ; why is that ? waiting for a response ." The text is labelled "pro" but wrongly predicted as "con". I think this text is really hard for any people to figure out whether it is pro or con without the context and the classifier predicted it as "con" possibly due to the negative words.

2.2. Gay rights

Three examples that did not work in my best model:

1. ID: C3046. Text: "Well in the good old day you did n't have to be that specific . It used to be a given that a mother and father is to raise a child . " It is labelled "con" but wrongly predicted as "pro". I think this text doesn't show a direct "con" attitude towards "gay rights" and has the positive word "good" inside, so the classifier mistook it as a positive attitude.

2. ID: C1789. Text: "Ah , yes . ReventonRage did pointed out a contradiction . However , both of you and him is forgetting the point . The point is that in one perspective or sense , it is wrong to be a homosexual . And in another sense , it is right to be a homosexual . I gave the reason why ...." It is labelled "pro" but wrongly predicted as "con". This author repeated a "con" statement from other people and it has one of the top 20 features "wrong", so the classifier mistook it as a negative attitude.

3. ID: C3014. Text: "Meaning that children will start to see it as normal and choose to be gay . How many times I 'm a gon na have to tell you this ? " It is labeeled "con" but wrongly predicted as "pro". This author stated a scene that children get influenced and justify gay, which s/he opposed. And the classifier didn't understand but thought it is positive.

2.3. Possible additional features

From the examples and analysis above, we could see that the possible reasons that the models made mistakes including: vague expression, missing context, citation of opposite opinions, rhetorical questions and negative words especially before important keywords. I think we may introduce features "#name say/-think" and quotation marks considering citation, features "why/how" and question marks considering rhetorical questions.

3. Feature Analysis

Top 20 features for topic "abortion" are: ['abortionist', 'arguments', 'awareness', 'choice', 'foetus', 'force', 'human', 'illegal', 'kill', 'living human', 'lord', 'potential', 'potential life', 'pregnancy', 'proposition', 'shall', 'survive outside', 'unto', 've', 'world'].

Top 20 features for topic "gay rights" are: ['argument', 'authority', 'evolution', 'female homosexuals', 'knowledge', 'million dollars', 'original', 'population', 'right', 'scenario', 'society', 'special', 'species', 'stated', 'survival', 'survival species', 'true', 'whomever', 'whomever want', 'wrong']

Although some top features look similar from two topics, like "world" (abortion) vs. "society" (gay rights) and "living human" (abortion) vs. "survival species", I think the most of the top features depend on the topic. For example, "pro-life" and "pro-choice" are the most featured words in the discussion of "abortion", so "choice" and "life"-related words are the top feature in the "abortion". However, they are not in the top features of "gay rights". On the other hand, "homosexuals" as the top feature in "gay rights", is not in "abortion". So I didn't use the same features for all topics and I don't think that would work well.