Ling 406: Introduction to Computational Linguistics

Term Project, Spring 2023

DUE DATE: May 12, 2023 (by midnight)

(This is a hard deadline! No extensions allowed)

Sentiment Classifier [100 points]

Implement a sentiment analyzer that automatically classifies input text as either positive or negative. Since we provide training data, it makes sense to use a supervised machine learning approach (you can use NLTK, scikit-learn, and/or any other machine learning platform/toolkit/libraries). Those of you who have a good handle of machine learning can also use deep learning models.

Approach:

First, you should produce a baseline system (i.e., a simple, first-stab approach that you are fairly confident will produce a measurable result). This system should be as simple as possible and should prove the feasibility of your plan. However, as for baselines, its performance does not necessarily need to be very high.

Then, you should conduct a series of experiments to improve your system. Try to learn from your results and revise your experiments as you go. Close-to-perfect results are nice, but we are also looking for good methodology in your experimentation. The idea is to start with the baseline system and find new good/informative features to improve it. So, in the end, we would like to see a performant system. However, as said before, we are more interested in the research process/methodology you use to build such a system.

Project questions:

Here are the questions intended to guide you in this process:

1) [20 points] In building the baseline system, what shallow text features make sense for the task as a first-stab approach? (a language modeling approach is a standard way to tackle this problem using, for example, a bag-of-words / a unigram model approach). Start with design decisions related to text preprocessing -- i.e., data cleaning (i.e., text normalization, stop words, etc.), tokenization, stemming/lemmatization, etc.), as we've shown in class.

2) [20 points] What machine learning models are suitable for conducting sentiment analysis? (i.e., compare 3-4 learning algorithms of your choice). Which performs best given your features?
3) [20 points] How would you improve the baseline model? (i.e., what text features are most beneficial to the task of sentiment analysis?) Experiment with at least 4 different features that go beyond a language modeling representation. For instance, consider deign decisions related to negation, POS tagging, dependency relations, etc.
4) [20 points] How does the size of the various feature sets you are experimenting with influence the performance of sentiment analysis? Compare the performance of different feature sets under the same feature selection scenario and machine learning algorithm. For instance, you have several options here:

   a) start with the feature set in the baseline model, and then add new features one at time (also called the *incremental approach*). For each new such addition, measure the performance on each of the machine learning models you selected at 2). Which of the machine learning models you chose at 2) works best with your features?

   b) another way is to add all new features to the baseline model, then compute their performance with a *leave-one-out approach* (as you did for Hw#3).

For this question you have to compute the performance, then compare and analyze the results. Which is the best combination of features (i.e., best feature set) with which machine learning model and why?

You have to answer questions 1) - 4) above and explain in a paper report all the steps you took to build the system. The paper report is worth 20 points (see Deliverables).

Dataset:

For data, there are many resources on this topic. However, for this project we will use the movie review dataset from Cornell: http://www.cs.cornell.edu/people/pabo/movie-review-data/. This is the polarity dataset v2.0 (3.0Mb) (includes README v2.0): 1000 positive and 1000 negative processed reviews, introduced in Pang/Lee ACL 2004. Released June 2004. (provided on the github space for this project).

Resources:

Here are some standard sentiment resources, such as sentiment lexicons, that you might want to use to come up with more informative features. Those presented in class are a good start. We also give you access to EmoLex, an Emotion Lexicon of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and sentiment values (negative and positive). The annotations were manually done by crowdsourcing. Please do NOT use without authors’ permission outside the context of this class!

Below are some suggested papers for you to read before you begin (they will give you a better idea about the task and feature suggestions):

Angiani, G., Ferrari, L., Fontanini,T., Fornacciari, P., Iotti, E., Magliani, F., &Manicardi, S. (2016). A Comparison betweenPreprocessing Techniques for SentimentAnalysis in Twitter. In KDWeb.

Zhuang, L., Jing, F., & Zhu, X. Y.(2006, November). Movie review mining andsummarization. In Proceedings of the 15thACM international conference on Informationand knowledge management (pp. 43-50).

Shu, L., Xu, H., & Liu, B. (2017).Life long learning CRF for supervised aspect extraction. arXiv preprint arXiv:1705.00251.

Yessenov, Kuat, and Sasa Misailovic. "Sentiment Analysis of Movie Review Comments" Massachusetts Institute of Technology, Spring 2009.
http://people.csail.mit.edu/kuat/courses/6.863/report.pdf

Satarupa Guha, Aditya Joshi, Vasudeva Varma. SIEL: Aspect Based Sentiment Analysis in Reviews. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 759–766,Denver, Colorado, June 4-5, 2015.
https://www.aclweb.org/anthology/S15-2129.pdf

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. Mohammad. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 437–442,Dublin, Ireland, August 23-24, 2014. https://www.aclweb.org/anthology/S14-2076.pdf

Mulyo, B. M., & Widyantoro, D. H.(2018). Aspect-Based Sentiment AnalysisApproach with CNN. Proceeding of the ElectricalEngineering Computer Science and Informatics,5(5), 142-147.

Deliverables:

This is an independent project (you must work alone although you can engage in design discussions with fellow colleagues — i.e., discussion on possible features, learning models, etc. is allowed.)

This is what you have to deliver (via github):

1) all the code you wrote to process the data (i.e., preprocessing code, etc. - outside of any other platform you used) plus a Readme file where you explain what the code does and how to run it.

2) You have to answer questions 1) - 4) above and explain in a paper report (report.pdf) all the steps you took to build the system. Each project report has to have the following structure:

- *Introduction* (consider the problem in a broader context: why is it important to work on sentiment analysis today, what applications can benefit from it, etc.)

- *Problem definition* (what is sentiment analysis and how do you define it in the context of Computational Linguistics: i.e., how do you define the task and what kind of input/output such a system has)

- *Previous work* (not comprehensive, but show you know something about this problem; cite accordingly)

- *Approach* (what computational approach did you use; what model(s) have you tested; what dataset(s) did you employ? did you perform some data preprocessing? if yes, what, how and why?; What text representation(s) have you used? What are your features/feature sets you played with?; etc.)

- *Results* (analysis of results; metrics used (standard ones are expected (precision, recall, F-measure, Accuracy) but if you use others, explain them); analysis of results - which feature set and machine learning model performed best for this task and why?)

- *Discussion and Conclusions* (what have you learned from this project; what possibilities of improvement are there for this problem and this approach; i.e., if you had to do it again, what would you change?)

NOTE: You can use any formatting style to write your report. However, we recommend the standard format used in the Computational Linguistics: style files (Latex, Word) are available here: http://acl2020.org/downloads/acl2020-templates.zip. If you prefer to use Overleaf, you can find the template here: https://www.overleaf.com/latex/templates/acl-2020-proceedings-template/zsrkcwjptpcd.

Extra-credit:

Students interested in extra-credit will have two options. You can choose both (for full extra-credit) or just one of them.

1) Option 1 [10 points]:

Write a more comprehensive Previous Work, and Discussion and Conclusions sections of the report. For previous work, you should identify 3 more recent papers (published no earlier than 2014) relevant to the project topic and compare and summarize the techniques used and their results. Moreover, you should also prepare a more thorough

Discussion and Conclusions section. Thus, in addition to the items listed above for this section, you should present a more detailed discussion of potential improvements (i.e., what linguistic representations are needed for this task? what challenges still remain to be solved and what solution do you suggest?)

2) Option 2 [30 points]:

Run the baseline (project question 1)) and the improved system (project question 4)) on a different dataset: Champaign-Urbana Yelp restaurant reviews (a collection of 10391 reviews of restaurants in the Champaign-Urbana area scraped from Yelp by John Hall, a former Linguistics student).

Note that this dataset is annotated with a star rating: {1star, 2star, 3star, 4star, 5star}

1star_count = 1172

2star_count = 1358

3star_count = 1795

4star_count = 2947

5star_count = 3119

You have to compare the results of your system on the two datasets (the movie reviews and the restaurant reviews). In order to do so, you will need to collapse all the reviews with at least 3 and a half stars into positive and the rest into negative target classes.

You have to answer the project questions 1) - 4) again for the new dataset and then compare the results on the two datasets at each step. For instance, which dataset is more challenging for sentiment analysis detection? What features work best for one dataset and not so well for the other? How about the best learning model(s)?