Mini Project 1: Document Classification Using a Polarity Lexicon

1 Goal

The goal of this project is to automatically predict the polarity of reviews. The suggested dataset you will work with contains pre-processed *positive* and *negative* reviews such as the following (emphasis on some words by me):

> kolya is one of the *richest* films i've seen in some time . zdenek sverak plays a confirmed *old* bachelor (who's likely to remain so) , who finds his life as a czech cellist increasingly *impacted* by the five-year *old* boy that he's taking care of . though it ends rather *abruptly*-- and i'm *whining* , 'cause i wanted to spend more time with these characters-- the acting , writing , and production values are as *high* as , if not *higher* than , comparable american dramas . this father-and-son *delight*-- sverak also wrote the script , while his son , jan , directed-- won a golden globe for *best* foreign language film and , a couple days after i saw it , walked away an *oscar* .

This example review is *positive*.

2 Method

One way to easily determine polarity is to look for individual words in the review that express strong positive or negative sentiment. The example contains words such as *richest* or *best*. To automate this process, we will rely on pre-compiled lists of positive and negative words. Example:

- Positive: *awesome, great, delight, rich, high, nice, higher, oscar, best, values*
- Negative: *bad, horrible, awful, impacted, abruptly, whining, old*

We then compute a score for each document as follows:

$$\text{score}(d) = \#\text{ of positive terms} - \#\text{ of negative terms}$$

If the score is larger than 0, we have a positive document. If it is smaller than 0, it is negative. For 0, we can make a random choice. In the above example, using only the lexicon listed above, we would obtain the following result:

$$\text{score} = 6 - 5 = 1$$

As the score is larger than 0, our prediction is that the document is positive. We evaluate the method by calculating accuracy, i.e., the ratio of documents where the prediction was correct.

3 Your task

You will write a program that predicts the polarity for a dataset of reviews as shown above. This dataset is the one used by Pang et al. (2002). You can download a copy at

`https://www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz`

The dataset consists of two folders, `pos` and `neg`, each containing 1000 positive and negative documents, respectively. This data will serve as your test data.

To make predictions, you need a polarity lexicon. You can obtain the one by Hu and Liu (2004) at

`http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`

This dataset consists of two files, `positive-words.txt` and `negative-words.txt`, containing several thousand positive and negative words, respectively. The files also contain some header information that you have to remove or ignore when processing.

You will now write a program that takes each document, calculates the score (see above) using the polarity lexicon, and checks whether the resulting prediction is correct.

4 Submission

For this project, you need to submit to me the following:

- Your working code
- Your predictions for the documents
- A short report (1-2 pages) describing the results you obtained:
  - Overall accuracy
  - error analysis of one correctly classified document and a wrongly classified document + an explanation for why why it was wrongly classified.

References

Hu, M., Liu, B., 2004. Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 168–177.

Pang, B., Lee, L., Vaithyanathan, S., July 2002. Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 79–86.
URL http://www.aclweb.org/anthology/W02-1011