WI - Mini Project 2

Group: Mikkel Larsen & Bruno Thalmann

Web Intelligence

Autmn Semester 2014

Indhold

1 Sentiment Analysis

1.0.1 Tokenization and Feature Extraction

The sentiment tokenization is very important because it catches a lot of the significant sentiment clues. Especially catching emoticons can mean a lot, because a lot of people use that for expressing how they feel about something. Capitalization and length of a word is also important to catch, fx 'this book is reeeeeeeealy BAD'.

In our implementation we did not take any of the topics mentioned above into account. We wanted to focus on getting a running program before improving its results. So our tokenization basically just splits the words, but the implementation is ready for more in depth tokenization.

1.0.2 Feature Extraction

Feature extraction is about which words to use and how to handle negation. We have chosen to use all words and our handling of negation can be seen in the next section.

Handling of Negation

Is used for identifying when a part of a sentence is negative. Fx 'I didn't like this product vs I really like this product'. Our strategy of handling negation is using the KISS principle[1]. The idea is to append '_NEG' to any word appearing between a negation and a punctuation mark. The negation words we have used are the following:

never, no, nothing, nowhere, noone, none, not, havent, hasnt, hadnt, cant, couldnt, shouldnt, wont, wouldnt, dont, doesnt, didnt, isnt, arent, aint

these are taken directly from the KISS principle. Our punctuation are also from there and are:

., :, ;, !, ?

An example of a string that has been negated. Before:

i don't think i will enjoy it: it might be too spicy.

After:

i don't think_NEG i_NEG will_NEG enjoy_NEG it_NEG: it might be too spicy.

[1] Das and Chen 2001; Pang, Lee & Vaithyanathan 2002

The classification is about learning from a set of data. When the learned model then is given some new data it should give back a classification of te given data based on the learned model.

1.0.3 Naive Bayes Classifier

Based on bayes theorem(who would have thought that..): $p(C|X) = p(X|C)p(C)/p(X)$.