Sentiment analysis in practice

Mike Thelwall
University of Wolverhampton, UK

Contents

- Creating a gold standard
- Feature selection
- Cross-validation

Recap

- The objective of commercial opinion mining is to automatically identify positive and negative sentiment from text, often about a product
- Examples:
  - "The film was fun and I enjoyed it."
    - -> positive sentiment
  - "The film lasted too long and I got bored."
    - -> negative sentiment

Gold standard

- A gold standard is a large set of texts with correct sentiment scores
- It is used for
  - Training machine learning algorithms
  - Testing all sentiment analysis algorithms
- Normally created by humans
- Time-consuming to create

Extract from gold standard

Positive	*Negative*	*Text*
2	-2	Hey witch what have you been up to?
3	-1	OMG my son has the same birthday as you! LOL!
1	-4	I regret giving my old car up. I couldn't afford four new tyres.
3	-1	Hey Kevin, hope you are good and well.

-1/1 = neutral; 5 = strongly positive; -5 = strongly negative

Gold standard hints

- Need random sample of 1000+ texts
  - Coded by 3+ independent coders, if possible
  - Use Krippendorff's alpha to assess agreement
  - Some disagreement is normal
  - Use code book to guide coders
  - Need to pilot test
  - Need to select reliable coders
- Or use Amazon's Mechanical Turk??

Test data: Inter-coder agreement

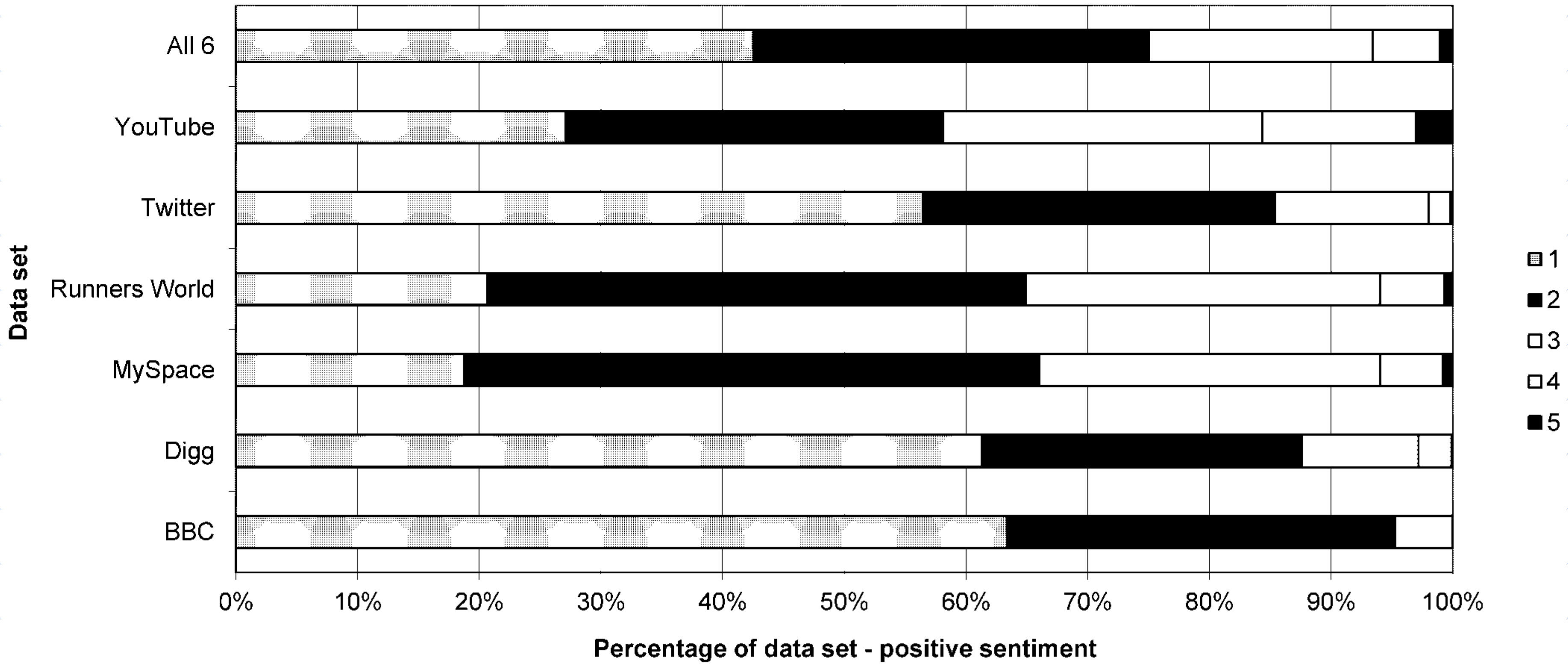
Test data = 1041 MySpace comments coded by 3 independent coders

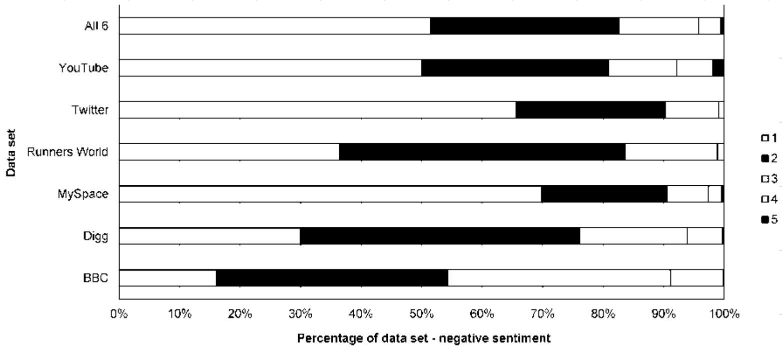
Krippendorff's inter-coder weighted alpha = 0.5743 for positive and 0.5634 for negative sentiment

Only moderate agreement between coders
but it is a hard 5-category task

Comparison for 1041 MySpace texts	**+ve agree-ment**	**-ve agree-ment**
Coder 1 vs. 2	**51.0%**	**67.3%**
Coder 1 vs. 3	**55.7%**	**76.3%**
Coder 2 vs. 3	**61.4%**	**68.2%**

Six social web gold standards





To test on a wide range of different Social Web text

Alternative gold standards

- Ratings coded with texts by authors
  - E.g., Movie reviews with overall movie ratings 1 star (terrible) – to 5 stars (excellent)

Audience Reviews for Black Swan View All

A bastard child of a huge gang of masters of psychologycal horror and drama. The plot is too obvious and full of gimmicks but Aronofsky dinamic eye and Portman's full immersion in the role makes it an entertaining trip with some sublime parts.

March 20, 2011

A goody-goody ballerina (Natalie Portman) must learn to tap into her dark side so she can dance the role of the seductive Black Swan; with the help of a free-spirited dancer (Mila Kunis) she does the job, maybe a little too well. The backstage melodrama drags a bit early on, but there's some wonderfully executed ... more

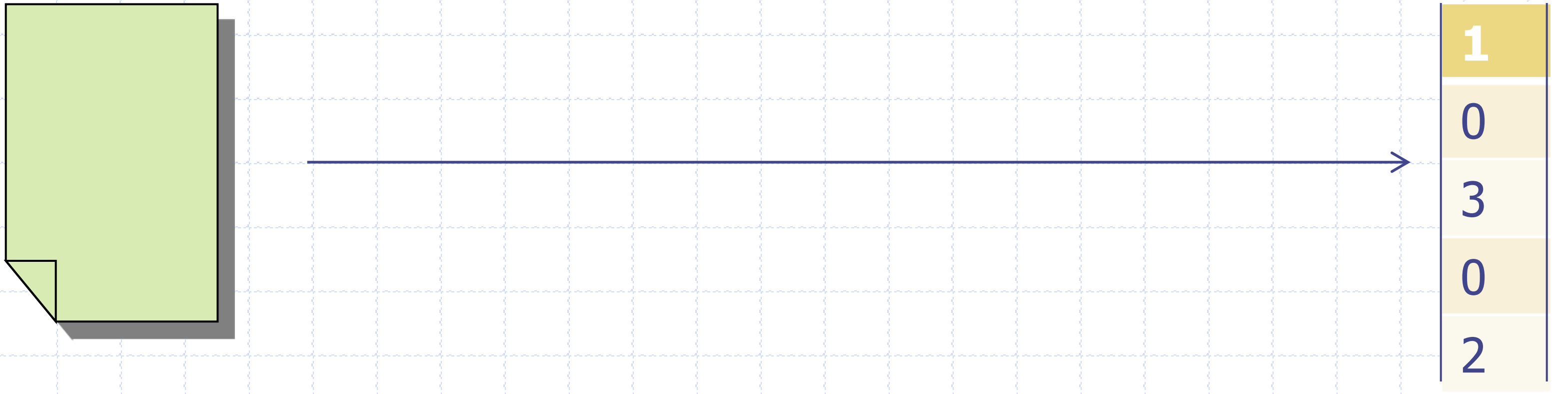
From rottentomatoes.com

Alternative gold standards

- Ratings inferred from text features
  - E.g., smiley at end indicates positive :) or negative :(
  - Not reliable? –smileys may mark sarcasm, irony. e.g., `I hate you :)`
- Automatic methods are cheap and can generate large training data

Feature selection

- Machine learning algorithms take a set of *features* as inputs
- Features are things extracted from texts
- Documents are converted into *feature vectors* for processing



Types of feature

- Features can be:
  - Individual words (unigrams = bag of words), pairs of words (bigrams), word triples (trigrams) etc.(n-grams)
  - Words can be stemmed or part-of-speech tagged (e.g., verb, noun, noun phrase)
  - Meta-information, such as the document author, document length, author characteristics

Feature types: unigrams

- Features: i, hate, anna, love, you
- Alphabetical: anna, hate, i, love, you
- d1 feature vector: ( )
- d2 feature vector: ( )

d1 I hate Anna.

d2 I love you.

Feature types: bigrams

- Features: i hate, hate anna, i love, love you
- Alphabetical: hate anna, i hate, i love, love you
- d1 feature vector:
- d2 feature vector:

d1 I hate Anna.

d2 I love you.

Feature types: trigrams

- Features:
- Alphabetical:
- d1 feature vector:
- d2 feature vector:

d1 I hate Anna.

d2 I love you.

Feature types: 1-3grams

- Alphabetical Features: anna, hate, hate
- d1 feature vector:
- d2 feature vector:

d1 I hate Anna.

d2 I love you.

ARFF files *Attribute-Relation File Format*

- ARFF file format is for machine learning
- Lists names and values of features

@attribute Polarity{-1,1}
@attribute Words numeric
@attribute love numeric
@attribute hate numeric
@attribute you numeric
@data
 1, 2, 1, 1, 0
-1, 2, 0, 1, 1

ARFF files– *another example*

@attribute Positive{1,2,3,4,5}
@attribute Bigrams numeric
@attribute love_you numeric
@attribute i_hate numeric
@attribute you_are numeric
@data
1, 3, 1, 1, 1
4, 2, 0, 1, 1

Task: make ARFF file for trigram data

Answer

Feature types: Alternatives

- Punctuation
- Stemmed or lemmatised text instead of original words
- Semantic information or part-of-speech
- Text length (number of terms in text)

Feature selection

- Sometimes machine learning algorithms work better if fed with only the *best* features
- Feature selection is using a process to select the best features
  - Normally those that discriminate best between classes
  - The value of each feature is estimated using a heuristic metric, such as Information Gain, Chi-Square or Log Likelihood

Feature quality

- The best features are those that most differentiate between positive and negative texts
  - "excellent" is a feature if 90% of texts in which it is found are positive
  - "and" is a feature if 50% of texts in which it is found are positive
- Frequent features are also more useful

Automatic feature selection

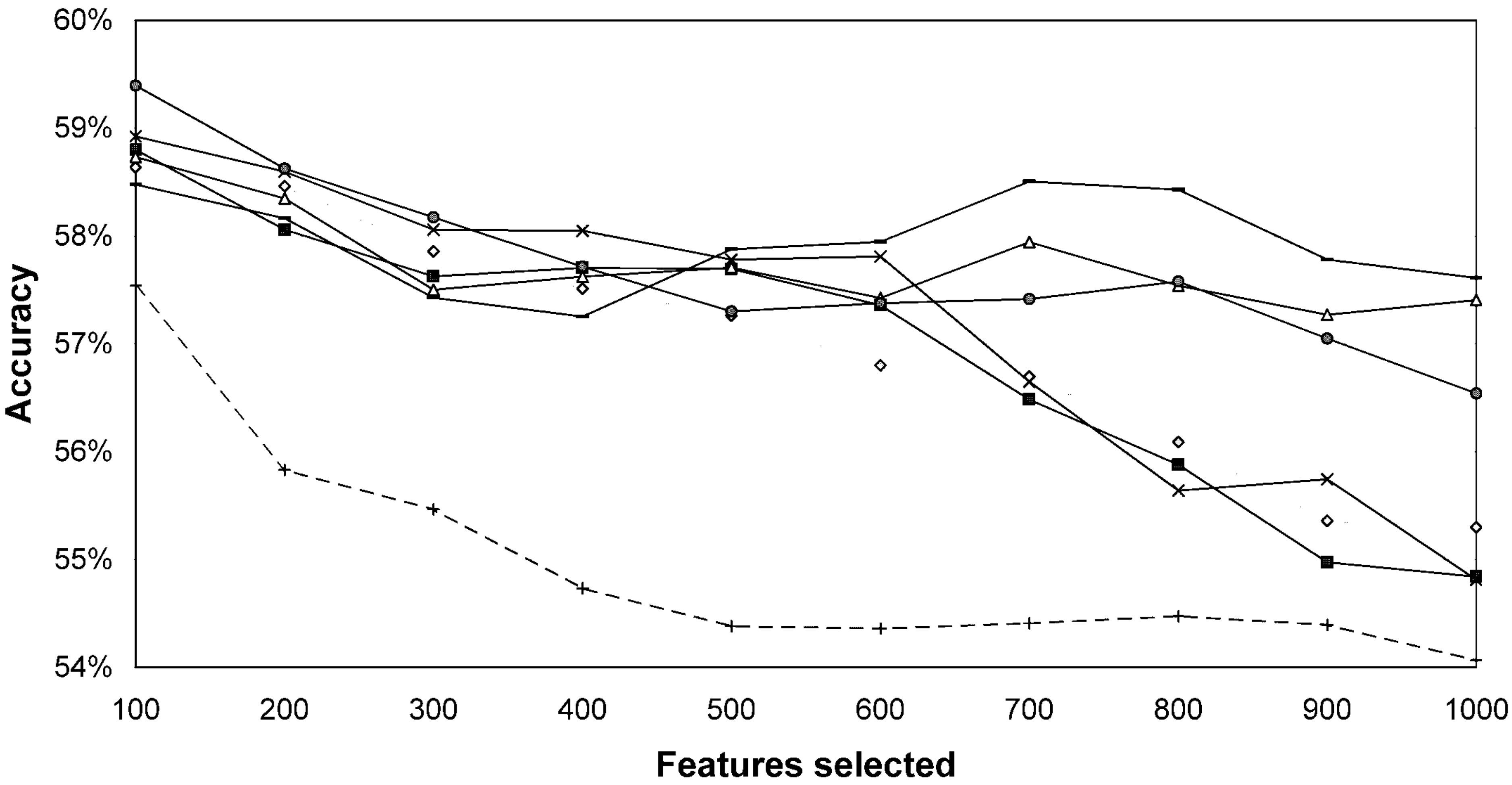
- Use a heuristic to rank features in terms of likely value for classification
  - E.g., Information Gain
- Select the top *n* features, e.g., *n* = 100, 1000
- In practice, experiment with different *n* or use largest feasible *n*

Simple example

Feature	Information Gain
I love	0.8
is excellent	0.7
excellent	0.6
dislike	0.5
not excellent	0.4
don’t really like	0.3
is strong	0.2
and it	0.1
then	0.0

What feature set size might give the best result for this data?

Why is the IG value for “and it” not zero?



n line represents a different features set with the SVM machine learn algorithm

The diagram shows that accuracy varies with feature set size

Cross-validation

- "10-fold cross validation"
  - Standard machine learning assessment technique
- Train opinion mining algorithm on 90% of the data
- Test it on the remaining 10%
- Repeat the above 10 times for a different 10% each time
- Average the results

10-Fold cross-validation

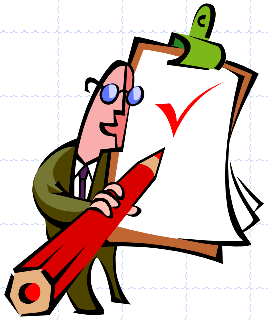
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data
- Data data data data data data data data data data

Round	Accuracy
1	81%
2	82%
3	81%
4	83%
5	81%
6	84%
7	82%
8	80%
9	84%
10	81%

Overall accuracy = ________

10-fold cross-validation

- Maximises the amount of "training" data
- Maximises the amount of "test" data



Alternative accuracy measures

- Binary or trinary tasks
  - precision, recall, f-measure
- Scale tasks
  - Near accuracy (e.g., prediction is within 1 of the correct value)
  - Correlation
    - The best measure, as uses all the data fully
  - Mean percentage error

SentiStrength vs. 693 other algorithms/variations

Results:+ve sentiment strength

Algorithm	Optimal #features	Accuracy	Accuracy +/- 1 class	Correlation
SentiStrength	**-**	**60.6%**	**96.9%**	**.599**
Simple logistic regression	700	**58.5%**	96.1%	**.557**
SVM (SMO)	800	**57.6%**	**95.4%**	**.538**
J48 classification tree	700	**55.2%**	**95.9%**	.548
JRip rule-based classifier	700	**54.3%**	96.4%	**.476**
SVM regression (SMO)	100	**54.1%**	97.3%	**.469**
AdaBoost	100	**53.3%**	**97.5%**	**.464**
Decision table	200	**53.3%**	96.7%	**.431**
Multilayer Perceptron	100	**50.0%**	*94.1%*	**.422**
Naïve Bayes	100	**49.1%**	**91.4%**	**.567**
Baseline	-	**47.3%**	**94.0%**	**-**
Random	-	**19.8%**	**56.9%**	**.016**

Results:-ve sentiment strength

Algorithm	Optimal #features	Accuracy	Accuracy +/- 1 class	Correlation
SVM (SMO)	100	**73.5%**	**92.7%**	**.421**
SVM regression (SMO)	300	**73.2%**	**91.9%**	**.363**
Simple logistic regression	800	**72.9%**	**92.2%**	**.364**
SentiStrength	-	**72.8%**	**95.1%**	**.564**
Decision table	100	**72.7%**	**92.1%**	**.346**
JRip rule-based classifier	500	**72.2%**	**91.5%**	**.309**
J48 classification tree	400	**71.1%**	**91.6%**	**.235**
Multilayer Perceptron	100	**70.1%**	**92.5%**	**.346**
AdaBoost	100	**69.9%**	**90.6%**	**-**
Baseline	-	**69.9%**	**90.6%**	**-**
Naïve Bayes	200	**68.0%**	**89.8%**	**.311**
Random	-	**20.5%**	**46.0%**	**.010**

Example differences/errors

- *THINK* 4 THE ADD
  - Computer (1,-1), Human (2,-1)
- 0MG 0MG 0MG 0MG 0MG 0MG 0MG 0MG!!!!!!!!!!!!!!!!!!!!N33N3R!!!!!!!!!!!!!!!!
  - Computer (2,-1), Human (5,-1)

SentiStrength 2

- Sentiment analysis programs are typically domain-dependant
- SentiStrength is designed to be quite generic
  - Does not pick up domain-specific non-sentiment terms, e.g., G3
- SentiStrength 2.0 has extended negative sentiment dictionary
  - In response to weakness for negative sentiment

Thelwall, M., Buckley, K., Paltoglou, G. (submitted).
High Face Validity Sentiment Strength Detection for the Social Web

SentiStrength 2 (unsupervised) tests

Social web sentiment analysis is less domain dependant than reviews

Data set	Positive Correlation	Negative Correlation
YouTube	0.589	0.521
MySpace	0.647	0.599
Twitter	0.541	0.499
Sports forum	0.567	0.541
Digg.com news	0.352	0.552
BBC forums	0.296	0.591
All 6	0.556	0.565

Summary

- Creating a gold standard is time-consuming but necessary – unless you can borrow one
- Machine learning algorithms use vectors of numbers extracted from the text – normally word/bigram/trigram frequencies
- Feature selection is important for effective machine learning
- Cross-validation allows data re-use – it is the best way to test an algorithm

Bibliography

- Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation, 39*(2-3), 165-210. [**creating a gold standard**]