AUTOMATIC DETECTION OF SATIRE AND SARCASM

Computational Approaches to Creative Language,
SS 2010, 22 June

Olga Nikitina

Intro

The task is novel - no stable framework

Similar tasks: text classification, sentiment analysis, opinion mining

Main problems:
- few uniform surface markers,
- context-sensitive,
- requires world knowledge
- culture-dependent

Models that are used for detection

1. Computational models of ironic environment with detection mechanism based on logical reasoning. Huge ontologies are required ([Utsumi 1996]).

2. Classifiers with bag-of-word features that detect unusual combination of words ([Burfoot, Baldwin 2009]).

3. Classifiers that use special surface markers that are known to come with specific ironic \ sarcastic contexts.

4. Classifiers based on a large number of surface patterns from both sarcastic and neutral documents ([Tsur et al 2010]).

Automatic satire detection

An SVM classifier + bag-of-word weighted features

<u>Additional targeted lexical features</u>

1. Headlines
    For each unigram in the headline, add a new feature.
2. Profanity
    Does the article contain profanity?
3. Slang and informal language
    3 features:
        Exact number of words marked as "slang"
        Is the number of such words higher than a certain upper boundary?
        Is the number of such words lower than a certain lower boundary?

Automatic satire detection

<u>Semantic validity</u>

A tool for detection of describing well-known entities in unfamiliar setting

How many documents are there on the Web that contain the same set of named entities?

-> Detect made-up entities
-> Detect unusual combinations of entities

Automatic satire detection

	Precision	Recall	F-score
all-to-satire	0.063	1.000	0.118
BIN	0.943	0.500	0.654
BIN + lex	0.945	0.520	0.671
BIN + val	0.943	0.500	0.654
BIN + all	0.945	0.520	0.671
BNS	0.944	0.670	0.784
BNS + lex	0.957	0.660	0.781
BNS + val	0.945	0.690	0.798
BNS + all	0.958	0.680	0.795

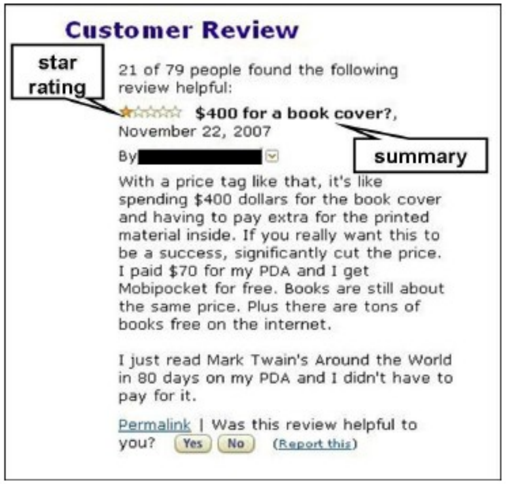
Good precision for all models: simple bag-of-words features are effective
Comparatively low recall: approx. 50% of satire articles cannot be recognized by these features only
Best F-score in BNS: feature weighting improves quality
Semantic validity enhances recall, but only with carefully weighted features.

Sarcasm recognition

Sarcasm is more various than a standard definition supposes:

- "[I] Love The Cover" (book)
- "Where am I?" (GPS device)
- "Trees died for this book?" (book)
- "Be sure to save your purchase receipt" (smart phone)
- "Are these iPods designed to die after two years?" (music player)
- "Great for insomniacs" (book)
- "All the features you want. Too bad they don't work!" (smart phone)
- "Great idea, now try again with a real product development team" (e-reader)
- "Defective by design" (music player)

Sarcasm recognition



<u>Data</u>
reviews of Amazon products

80 sarcastic sentences
(level of sarcasm from 3 to 5)
+ 505 neutral sentences
(level of sarcasm from 1 to 2)

<u>Automatic expansion of the training set</u>

Seed: "*This book was really good – until page 2!*"
Found: "*Gee, I thought this book was really good until I found out the author didn't get into Bread Loaf!*"
Accompanying: "*It just didn't make much sense.*"

In total:
471 sarcastic + 5020 neutral sentences

Sarcasm recognition

<u>Pattern-based features</u>

<u>Sentence:</u>
Garmin apparently does not care much about product quality or customer support

<u>Patterns:</u>
[company] CW does not CW much
does not CW much about CW CW or

Filter out those patterns that appear only for 1 product
Filter out those that appear both in sentences with sarcasm level 1 and 5

Pattern matching: 1 – exact, 0.1 – sparse, 0.1*n/N – incomplete, 0 – no match

Sarcasm recognition

<u>Punctuation-based features</u>

1. Sentence length in words
2. Number of “!” characters
3. Number of “?” characters
4. Number of quotes
5. Number of cpitalized\all capitals words

Sarcasm recognition

KNN-classifier with Euclidean distance as a measure for data points similarity, k = 5

Label (=sarcasm level) of the test sentence is a weighted average of the k closest training set vectors

$$Label(v) = [\frac{1}{k} \sum_i \frac{Count(Label(t_i))\,Label(t_i)}{\sum_j Count(label(t_j))}]$$

$$Count(l) = Fraction\,of\ vectors \in the\,training\,set\,with\,label\,l$$

Sarcasm recognition

<u>Evaluation</u>

5-fold cross validation

	Precision	Recall	Accuracy	F-score
punctuation	0.256	0.312	0.821	0.281
patterns	0.743	0.788	0.943	0.765
pat+punct	0.868	0.763	0.945	0.812
enrich punct	0.400	0.390	0.832	0.395
enrich pat	0.762	0.777	0.937	0.769
all	0.912	0.756	0.974	0.827

<u>High accuracy</u>: biased seed data (sarcastic sentences are rare).
<u>Low precision and recall for punctuation</u>: different means of expressing sarcasm in written text and online communication
<u>Combination of features</u> gives the best performance.

Sarcasm recognition

Gold-standard evaluation

	Precision	Recall	False Pos	False Neg	F-score
Star-sentiment	0.5	0.16	0.05	0.44	0.242
SASI	0.766	0.813	0.11	0.12	0.788

<u>Low recall for start-sentiment</u>: it fails to recognize subtle sarcasm
<u>High performance of SASI</u>: it does not over-fitting the data

Sarcasm recognition

Insights into sarcasm marking strategies:

- Surface markers (“yeah, great!”) are included in patterns

- Some combinations of punctuators + other features are also good markers (although punctuation alone is weak)

- Written cues are good, but show low recall and low precision -> they are ambiguous

- Context can be captured by patterns, since they are not limited to sentences

Literature

[Burfoot, Baldwin 2009] –Clint Burfoot, Timothy Baldwin, Automatic satire detection: are you having a laugh? In: *Proceedings of the ACL-IJCNLP 2009 Conference short papers, pp. 161-164, Suntec, Singapore, 4 August 2009*

[Tsur et al 2010] – Oren Tsur, Dmitry Davidov, Ari Rappoport , ICWSM – A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews, 2010

[Utsumi 1996] - A. Utsum. A unified theory of irony and its computational formalization. Coling-96, pp. 962-967, 1996

Measures

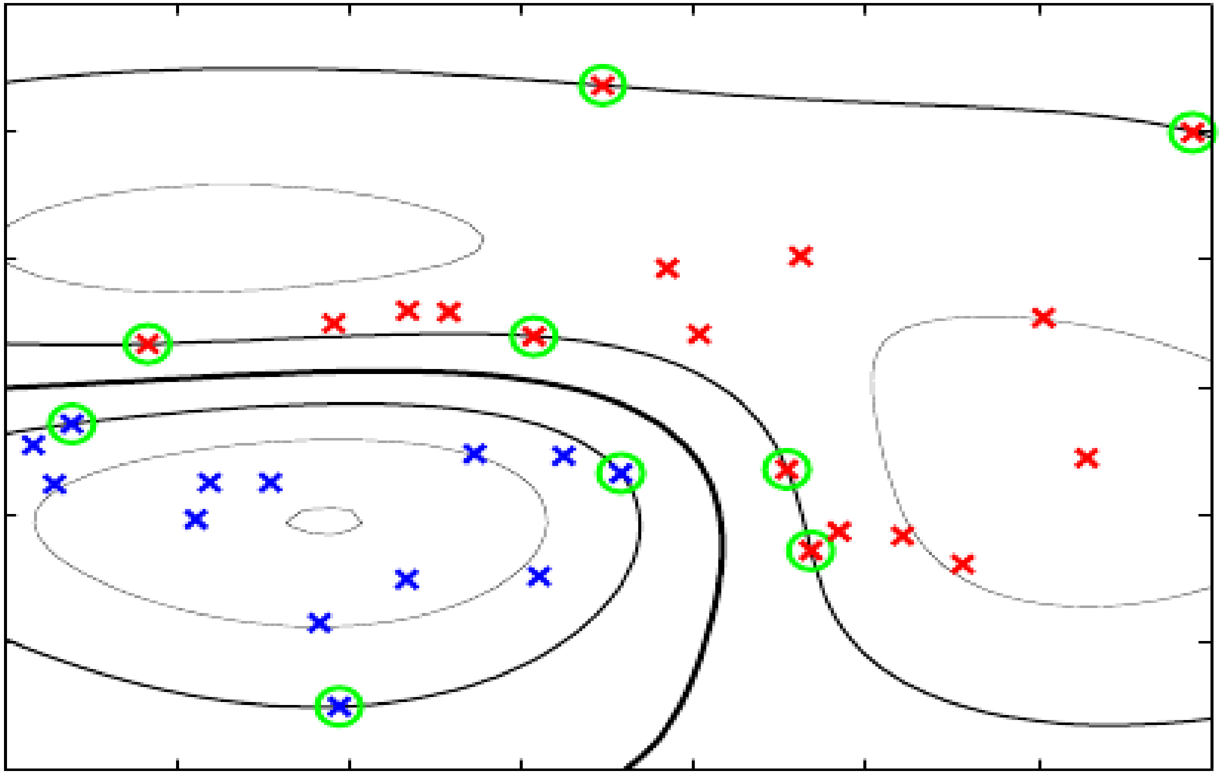
		Gold standard	
		True	False
Test outcome	Positive	True positive	False positive
	Negative	False negative	True negative

$$Accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}$$

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

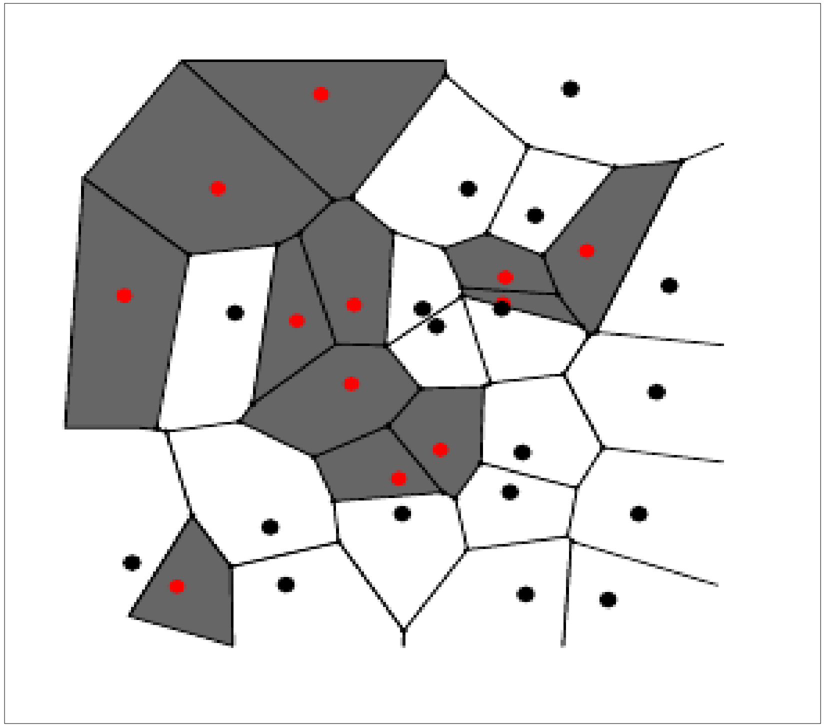
$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Support Vector Machines



Somehow represent your data.
Find the boundary between classes by minimizing the generalization error
To do it, maximize the distance between periphery data points and the boundary.
Such data points are called *support vectors*. The distance is called *margin*.
The main idea is that the decision about the boundary depends mostly on support vectors and is not influenced by other data points.

K-nearest Neighbors



Somehow represent your data.
The class of each test data point is the same as the class of its nearest train neighbor.
If you take k nearest neighbors, put the data to the same class as the majority of neighbors.