

# Thumbs Up? Sentiment Classification using Machine Learning Techniques

Pang, Lee, Vaithyanathan - EMNLP 2002

Slides by: Dor Cohen, Itai Gat

IE @ Technion

October 16, 2018

# Agenda

## 1 Introduction

- Topic classification
- Sentiment analysis

## 2 Problem

- Problem definition
- Data
- Human baseline

## 3 Methods

- Bag of words
- Naive bayes
- Maximum entropy
- SVM

## 4 Results

- Results

## 5 Reproduce results

## 6 Conclusions

# Introduction

# Topic classification

- Recent (2002) works sort documents according to their **subject**
  - ▶ e.g., sports vs. politics

# Topic classification

- Recent (2002) works sort documents according to their **subject**
  - ▶ e.g., sports vs. politics
- Yet crucial part of online posted articles is their **sentiment**
  - ▶ provide useful insights for readers automatically
  - ▶ e.g., product review is negative or positive

# Topic classification

Current (2002) techniques for non-topic text categorization

- Source style with features as stylistic variation (Biber, 1988)
  - ▶ e.g., author, publisher (NY times vs. Daily News)
- Genre of text (Finn et al., 2002)
  - ▶ e.g., editorial 'subjective' genre
- Is subjective language used? (Wiebe et al., 2001)
- Does text contains opinion expressing?

# Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - ▶ Q: What are our expected challenges?

# Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - ▶ Q: What are our expected challenges?
  - ▶ A: Topics are identifiable by key words alone, while sentiment requires more **understanding**



# Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - ▶ Q: What are our expected challenges?
  - ▶ A: Topics are identifiable by key words alone, while sentiment requires more **understanding**
- e.g., "How could anyone sit through this movie?"
  - ▶ Can you mark any negative word?

# Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - ▶ Q: What are our expected challenges?
  - ▶ A: Topics are identifiable by key words alone, while sentiment requires more **understanding**
- e.g., "How could anyone sit through this movie?"
  - ▶ Can you mark any negative word?
- Previous (2002) techniques:
  - ▶ Cognitive linguistic models (Sack, 1994)
  - ▶ Discriminant word lexicons (Tong, 2001)
  - ▶ Semantic orientation of words (Turney and Littman, 2002)

# Problem

# Problem definition

- Find a mapping from text document to binary label

# Problem definition

- Find a mapping from text document to binary label
  - ▶ Supervised learning

# Problem definition

- Find a mapping from text document to binary label
  - ▶ Supervised learning
- For  $m$  numeric features we define the mapping as:

# Problem definition

- Find a mapping from text document to binary label
  - ▶ Supervised learning
- For  $m$  numeric features we define the mapping as:

## Definition (Binary classifier)

$$f : X \rightarrow y$$

where  $X \in \mathbb{R}^m, y \in \{0, 1\}$

# Problem definition

- Find a mapping from text document to binary label
  - ▶ Supervised learning
- For  $m$  numeric features we define the mapping as:

## Definition (Binary classifier)

$$f : X \rightarrow y$$

where  $X \in \mathbb{R}^m, y \in \{0, 1\}$

- Evaluating the mapping is done by loss function



# Problem definition

- Find a mapping from text document to binary label
  - ▶ Supervised learning
- For  $m$  numeric features we define the mapping as:

## Definition (Binary classifier)

$$f : X \rightarrow y$$

where  $X \in \mathbb{R}^m, y \in \{0, 1\}$

- Evaluating the mapping is done by loss function
- e.g., Zero-one loss:  $L(x, y, f_w) = \mathbf{1}\{f_w(\hat{x}) \neq y\}$ 
  - ▶  $w$  denotes learned parameters

# Data: IMDB Movie Reviews

- Lucky for us: user rating provides us **supervised** learning
- Converted into three categories (or topics):
  - ▶ *Positive*, *negative*, (and *neutral* - not used)
- Avoid bias issues:
  - ▶ 20 reviews per author per sentiment
  - ▶ 752 negative vs 1301 positive
  - ▶ total of 144 reviewers

# Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)

# Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - ▶ Topics can be related, while with opinions people tend to express strong feelings

# Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - ▶ Topics can be related, while with opinions people tend to express strong feelings
- Hypothesis: certain words indicate on sentiment type

# Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - ▶ Topics can be related, while with opinions people tend to express strong feelings
- Hypothesis: certain words indicate on sentiment type
- Test: decision procedure - count positive vs. negative words

# Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - ▶ Topics can be related, while with opinions people tend to express strong feelings
- Hypothesis: certain words indicate on sentiment type
- Test: decision procedure - count positive vs. negative words

Human	Proposed words	Accuracy	Ties <sup>1</sup>
1	positive (5): dazzling, brilliant.. negative (5): suck, terrible..	58%	75%
2	positive (11): gripping, spectacular.. negative (6): cliched, boring..	64%	39%

**Table:** Baseline results for human word lists, data is balanced (700 vs. 700)

---

<sup>1</sup>Documents percentage where sentiments rated equally

# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)



# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)
  - ▶ Not necessarily the reason for low accuracy!

# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)
  - ▶ Not necessarily the reason for low accuracy!
- Authors propose their own list of words
  - ▶ Backed up with preliminary **data analysis** (including test set)

# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)
  - ▶ Not necessarily the reason for low accuracy!
- Authors propose their own list of words
  - ▶ Backed up with preliminary **data analysis** (including test set)

Human	Proposed words	Accuracy	Ties
3+Stats	positive (7): love, wonderful.. negative (7): bad, worst, '?', '!',...	69%	16%

**Table:** Results where words (total 14) were chosen based on data statistics

# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)
  - ▶ Not necessarily the reason for low accuracy!
- Authors propose their own list of words
  - ▶ Backed up with preliminary **data analysis** (including test set)

Human	Proposed words	Accuracy	Ties
3+Stats	positive (7): love, wonderful.. negative (7): bad, worst, '?', '!',...	69%	16%

**Table:** Results where words (total 14) were chosen based on data statistics

(2018) Reproduce data analysis, feature occurrences binarized:

- $P(\text{love} | \widehat{d} = \text{positive}) = 0.1796$  ,  $P(\text{love} | \widehat{d} = \text{negative}) = 0.1334$

# Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed words list is relatively short (effect is 0 vs. 0 ties)
  - ▶ Not necessarily the reason for low accuracy!
- Authors propose their own list of words
  - ▶ Backed up with preliminary **data analysis** (including test set)

Human	Proposed words	Accuracy	Ties
3+Stats	positive (7): love, wonderful.. negative (7): bad, worst, '?', '!',...	69%	16%

**Table:** Results where words (total 14) were chosen based on data statistics

(2018) Reproduce data analysis, feature occurrences binarized:

- $P(\text{love}|\widehat{d} = \text{positive}) = 0.1796$  ,  $P(\text{love}|\widehat{d} = \text{negative}) = 0.1334$
- $P(\text{worst}|\widehat{d} = \text{positive}) = 0.0252$  ,  $P(\text{worst}|\widehat{d} = \text{negative}) = 0.0937$

# Methods

# Bag of words

- $d_1$ : "The audio quality really stinks."

# Bag of words

- $d_1$ : "The audio quality really stinks."
- Extract features:
  - ▶ Unigrams: {the, audio, ... }
  - ▶ Bigrams: {the audio, audio quality, really stinks, ... }
  - ▶ N-gram!



# Bag of words

- $d_1$ : "The audio quality really stinks."
- Extract features:
  - ▶ Unigrams: {the, audio, .. }
  - ▶ Bigrams: {the audio, audio quality, really stinks, .. }
  - ▶ N-gram!

## Definition (Bag of words framework)

Let  $\{f_1, .. f_m\}$  denote set of  $m$  features that can appear in document.

Let  $n_i(d)$  be the number of times  $f_i$  occurs in document  $d$ .

Then each document  $d$  is represented by  $\vec{d} := (n_1(d), ..., n_m(d))$ .

# Bag of words

- $d_1$ : "The audio quality really stinks."
- Extract features:
  - ▶ Unigrams: {the, audio, .. }
  - ▶ Bigrams: {the audio, audio quality, really stinks, .. }
  - ▶ N-gram!

## Definition (Bag of words framework)

Let  $\{f_1, .. f_m\}$  denote set of  $m$  features that can appear in document.

Let  $n_i(d)$  be the number of times  $f_i$  occurs in document  $d$ .

Then each document  $d$  is represented by  $\vec{d} := (n_1(d), ..., n_m(d))$ .

## Unigram example

- $\vec{d}_1 = (The : 1, audio : 1, quality : 1, really : 1, stinks : 1)$

# Bag of words

- $d_1$ : "The audio quality really stinks."
- Extract features:
  - ▶ Unigrams: {the, audio, ... }
  - ▶ Bigrams: {the audio, audio quality, really stinks, ... }
  - ▶ N-gram!

## Definition (Bag of words framework)

Let  $\{f_1, \dots, f_m\}$  denote set of  $m$  features that can appear in document.

Let  $n_i(d)$  be the number of times  $f_i$  occurs in document  $d$ .

Then each document  $d$  is represented by  $\vec{d} := (n_1(d), \dots, n_m(d))$ .

## Unigram example

- $\vec{d}_1 = (\text{The} : 1, \text{audio} : 1, \text{quality} : 1, \text{really} : 1, \text{stinks} : 1)$
- $d_2$ : "Stinking quality, really stinks". Question:  $\vec{d}_2 = ?$

# Bag of words

- $d_1$ : "The audio quality really stinks."
- Extract features:
  - ▶ Unigrams: {the, audio, ... }
  - ▶ Bigrams: {the audio, audio quality, really stinks, ... }
  - ▶ N-gram!

## Definition (Bag of words framework)

Let  $\{f_1, \dots, f_m\}$  denote set of  $m$  features that can appear in document.

Let  $n_i(d)$  be the number of times  $f_i$  occurs in document  $d$ .

Then each document  $d$  is represented by  $\vec{d} := (n_1(d), \dots, n_m(d))$ .

## Unigram example

- $\vec{d}_1 = (\text{The} : 1, \text{audio} : 1, \text{quality} : 1, \text{really} : 1, \text{stinks} : 1)$
- $d_2$ : "Stinking quality, really stinks". Question:  $\vec{d}_2 = ?$
- $\vec{d}_2 = (0, 0, 1, 1, 2)$

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- In practice to estimate  $P(d|c)$ , we **naively** assume  $f_i$  are conditionally independent.

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- In practice to estimate  $P(d|c)$ , we **naively** assume  $f_i$  are conditionally independent.
  - ▶ Hence  $\widehat{P(d|c)} = \prod_{i=1}^m P(f_i|c)^{n_i(d)}$



# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- In practice to estimate  $P(d|c)$ , we **naively** assume  $f_i$  are conditionally independent.
  - ▶ Hence  $\widehat{P(d|c)} = \prod_{i=1}^m P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you can think about?

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- In practice to estimate  $P(d|c)$ , we **naively** assume  $f_i$  are conditionally independent.
  - ▶ Hence  $\widehat{P(d|c)} = \prod_{i=1}^m P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you can think about?
- $A_1$ : Some estimates are zero, can smooth (e.g., add-one smoothing)

# Naive Bayes classifier

- Assign class which maximizes probability:  $c^* = \operatorname{argmax}_c P(c|d)$
- Recap:

## Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- In practice to estimate  $P(d|c)$ , we **naively** assume  $f_i$  are conditionally independent.
  - ▶ Hence  $\widehat{P(d|c)} = \prod_{i=1}^m P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you can think about?
- $A_1$ : Some estimates are zero, can smooth (e.g., add-one smoothing)
- $A_2$ : Short documents vs. long documents (propose: tf-idf)

# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$  - normalization function

# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$  - normalization function
- $F_{i,c}$  is a *feature/class function*

# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$  - normalization function
- $F_{i,c}$  is a *feature/class function*
  - ▶ Defined:  $F_{i,c}(d, c') = 1$  if feature  $i$  **appears** on document  $d$  and its estimated class is  $c$ , o.w.  $F_{i,c}(d, c) = 0$

# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$  - normalization function
- $F_{i,c}$  is a *feature/class function*
  - ▶ Defined:  $F_{i,c}(d, c') = 1$  if feature  $i$  **appears** on document  $d$  and its estimated class is  $c$ , o.w.  $F_{i,c}(d, c) = 0$
- $\lambda_{i,c}$  - feature-weight parameters
  - ▶ large values imply  $f_i$  is a strong indicator for class  $c$



# Maximum entropy classifier

aka logistic regression

## Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$  - normalization function
- $F_{i,c}$  is a *feature/class function*
  - ▶ Defined:  $F_{i,c}(d, c') = 1$  if feature  $i$  **appears** on document  $d$  and its estimated class is  $c$ , o.w.  $F_{i,c}(d, c) = 0$
- $\lambda_{i,c}$  - feature-weight parameters
  - ▶ large values imply  $f_i$  is a strong indicator for class  $c$

## Fit procedure

- Training data used to estimate distribution  $F$
- $\lambda$ 's are set to maximize entropy of induced distribution

# Support vector machines

- Goal: Find hyperplane  $w$  which separates classes with margin large as possible

# Support vector machines

- Goal: Find hyperplane  $w$  which separates classes with margin large as possible
- In this setting we define  $w$  as:

## Definition (SVM hyperplane)

Let  $c_j \in \{1, -1\}$  be the class of document  $d_j$  then:

$$w := \sum_j \alpha_j c_j \vec{d}_j, \alpha_j \geq 0$$

- $\alpha_j$  are obtained by solving dual optimization problem.

# Support vector machines

- Goal: Find hyperplane  $w$  which separates classes with margin large as possible
- In this setting we define  $w$  as:

## Definition (SVM hyperplane)

Let  $c_j \in \{1, -1\}$  be the class of document  $d_j$  then:

$$w := \sum_j \alpha_j c_j \vec{d_j}, \alpha_j \geq 0$$

- $\alpha_j$  are obtained by solving dual optimization problem.

Alternative fitting procedure:

## Definition (Gradient descent)

$$w = w - \alpha * \frac{\partial L(X, w)}{\partial w}, \text{ Update till convergence}$$

# Results

## Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

## Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%

## Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%
- Topic-based classification reached 90%+ accuracy



## Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%
- Topic-based classification reached 90%+ accuracy
  - ▶ Settings were multi-class
  - ▶ We conclude that sentiment analysis is harder

# Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%
- Topic-based classification reached 90%+ accuracy
  - ▶ Settings were multi-class
  - ▶ We conclude that sentiment analysis is harder
- The frequency vs. presence of features seems to make the difference

# Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	<b>82.9%</b>

**Table:** 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%
- Topic-based classification reached 90%+ accuracy
  - ▶ Settings were multi-class
  - ▶ We conclude that sentiment analysis is harder
- The frequency vs. presence of features seems to make the difference
  - ▶ Hence from this point authors use presence (**binarized occurrences**)

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	<b>81.6%</b>

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.



# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	<b>81.6%</b>

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	<b>81.6%</b>

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse
- Part-of-speech: "I love this movie" vs. "This is a love story"

# Results and discussion

## Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	<b>82.9%</b>
3	uni+bigrams	32330	80.6%	80.8	<b>82.7%</b>
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	<b>81.9%</b>
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	<b>81.6%</b>

**Table:** 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse
- Part-of-speech: "I love this movie" vs. "This is a love story"
- Position based on dividing text into quarters.

Reproduce results

# Reproduce results

(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	<b>82.9%</b>	NA

# Reproduce results

(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	<b>82.9%</b>	NA
unigrams	16165	77.48%	<b>81.52%</b>	80.66%	<b>82.75%</b>

Table: Original vs. our results

# Reproduce results

(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	<b>82.9%</b>	NA
unigrams	16165	77.48%	<b>81.52%</b>	80.66%	<b>82.75%</b>

Table: Original vs. our results

- MLP is 2-layer neural network with 100 Relu neurons
- **No tuning** was used (sklearn 0.19.2 default parameters)
  - ▶ Plus not all described processing steps applied
- Notebook is available [▶ here](#)

# Classifier comparison

(2018)

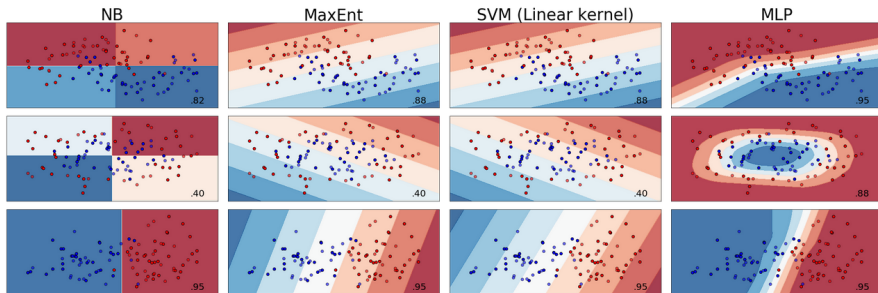
- Let's observe our classifiers decision boundaries for some toy datasets



# Classifier comparison

(2018)

- Let's observe our classifiers decision boundaries for some toy datasets



- Accuracy is reported

# Conclusions

# Conclusions

- Unigrams presence setting achieves the best performance
  - ▶ Apply feature selection algorithms

# Conclusions

- Unigrams presence setting achieves the best performance
  - ▶ Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

# Conclusions

- Unigrams presence setting achieves the best performance
  - ▶ Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

## Review example

*"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."*

# Conclusions

- Unigrams presence setting achieves the best performance
  - ▶ Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

## Review example

*"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."*

- Difficult for bag-of-words classifiers.

# Conclusions

- Unigrams presence setting achieves the best performance
  - ▶ Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

## Review example

*"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."*

- Difficult for bag-of-words classifiers.
- Authors suggest determining the **focus** of each sentence, if is on/off topic.

Thank you for participating!  
Questions?