Sarcasm detection for sentiment analysis:
From linguistic to machine learning approaches

윤수원
서울시립대학교

Structure of Lecture

1. **Sentiment Analysis**

2. **Linguistic Analysis of Senti-words and Sarcasm**

3. **Machine Learning approaches for Sarcasm detection**

Structure of Lecture

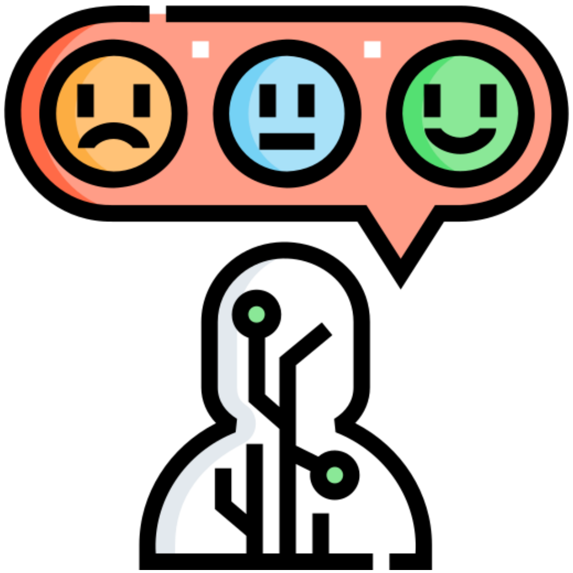
1. **Sentiment Analysis**

2. **Linguistic Analysis of Senti-words and Sarcasm**

3. **Machine Learning approaches for Sarcasm detection**

Definition of **sentiment analysis**

- Sentiment analysis is a **natural language processing (NLP) technique.**
- It involves the use of computational methods to **analyze the emotional tone** expressed in a piece of text, such as a **review, comment, or social media** post.
- The main objective of sentiment analysis is to identify whether the expressed sentiment is **positive, negative, or neutral.**



How is Sentiment Analysis Possible?

- The text is analyzed for the presence of **specific words, phrases, or linguistic patterns** that indicate the writer's emotions or attitude towards a particular topic, product, service, or event.

Common types of indicators

Positive and Negative Keywords

- Words and phrases that directly convey positive or negative sentiments
- "good," "excellent," "happy," "bad," "terrible," "disappointing," etc.

Intensifiers and Diminishers

- Words that amplify or reduce the strength of a sentiment
- "very," "extremely," "slightly," "quite," etc.

Polarity Shifters

- Words that change the sentiment of a statement
- negation words (e.g., "not," "no," "never")

Repetition

- Repeated words or phrases may highlight the writer's sentiment.

Types of Sentiment Analysis

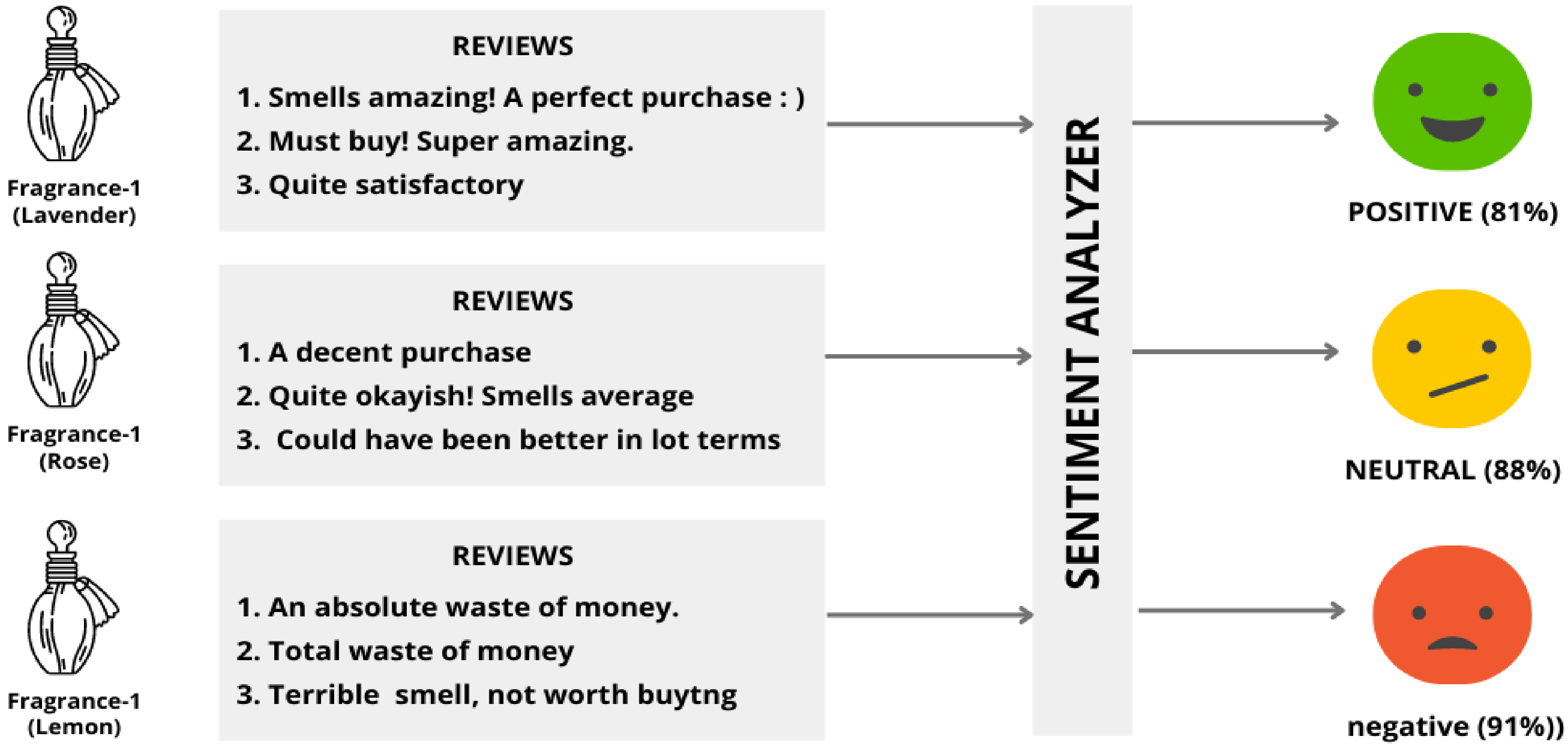
- **Document-level analysis** determines the overall sentiment of the document.
- **Sentence-level analysis** examines sentiments within each sentence.
- **Aspect-based analysis** focuses on specific aspects mentioned in the text.
- **Entity-level analysis** identifies sentiments towards specific targets.
- **Comparative analysis** compares sentiments between different entities or aspects, providing insights into relative preferences.

Why Conduct Sentiment Analysis?

- The insights gained from sentiment analysis can be **valuable for making data-driven decisions**, such as improving customer satisfaction and identifying areas for improvement or intervention.

- Sentiment analysis is widely used in **marketing, customer service, market research, and social media monitoring**, among other fields.

Example



Sentiment Analysis Using Python, Analytics Vidhya
https://www.analyticsvidhya.com/blog/2022/07/sentiment-analysis-using-python/

- **Fragrance-1 (Lavender)** has highly **positive** reviews.
- **Fragrance-2 (Rose)** happens to have a **neutral** outlook.
- **Fragrance-3 (Lemon)** has an overall **negative** sentiment.

Ways to Perform Sentiment Analysis in Statistics Programs/Languages

- **Python** with NLTK (Natural Language Toolkit)
  – Python provides powerful libraries like NLTK that offer tools for text processing, tokenization, and sentiment analysis.
- **R** with SentimentAnalysis and TextMining Packages
  – R programming language has packages like SentimentAnalysis and TextMining that enable sentiment analysis on text data.
- **Java** with Stanford NLP Library
  – The Stanford Natural Language Processing (NLP) library offers Java-based tools for sentiment analysis tasks.

Improvement methods

- **Sarcasm and Irony**
  - Contextual analysis may be needed to identify sarcastic or i ronic statements, where the literal meaning is different fro m the intended sentiment.
- **Contextual Clues**
  - Understanding the overall context of the text can help in re cognizing sentiment, as certain words or phrases may have different connotations depending on the context.

Example #1: Sentiment Analysis

Data source: https://www.gutenberg.org

Project Gutenberg | About ▾ | Search and Browse ▾ | Help ▾

Quick search | Go! | Donation | PayPal

Project Gutenberg › 66,745 free ebooks › 333 by William Shakespeare

Romeo and Juliet by William Shakespeare

Download This eBook

	Format	Size			
	Read this book online: HTML	198 kB			
	EPUB (with images)	268 kB			
	EPUB (no images)	143 kB			
	Kindle (with images)	812 kB			
	Kindle (no images)	496 kB			
	Plain Text UTF-8	165 kB			
	More Files…				

Example #1: Sentiment Analysis

Pre-processing: tedious, technical, but unavoidable, necessary task.
(Some data scientists jobs are mostly about pre-processing.)

Example #1: Sentiment Analysis

Pre-processing: tedious, technical, but unavoidable, necessary task.
(Some data scientists jobs are mostly about pre-processing.)

<u>Summary of pre-processing and sentiment analysis in R code</u>

1. Data download in R code: Romeo and Juliet (dplyr's tibble format, cf. data.frame)
2. Data pre-processing: remove auxiliary texts & tokenize
3. Break down sentences to words and remove stop words
4. Assign sentiment (positive/negative/neutral) for each word
5. Raw sentiment score for each sentence (# positive/negative words)

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg
   1:          1     0     0 FALSE FALSE
   2:          2     0     0 FALSE FALSE
   3:          3     0     0 FALSE FALSE
   4:          4     0     0 FALSE FALSE
   5:          5     0     0 FALSE FALSE
  ---
3642:       3642     1     0  TRUE FALSE
3643:       3643     0     1 FALSE  TRUE
3644:       3644     0     1 FALSE  TRUE
3645:       3645     0     0 FALSE FALSE
3646:       3646     0     1 FALSE  TRUE
```

<u>Optional steps</u>

6. Conversion to data.table format (data.table package)
7. Add rows for neutrals by inner join
8. Assign zero score on neutral sentences
9. Identify positive / negative sentences
10. Generate hypothetical true sentiments of sentences (10% flip)

Example #1: Sentiment Analysis

Pre-processing: tedious, technical, but unavoidable, necessary task.
(Some data scientists jobs are mostly about pre-processing.)

Summary of pre-processing and sentiment analysis in R code

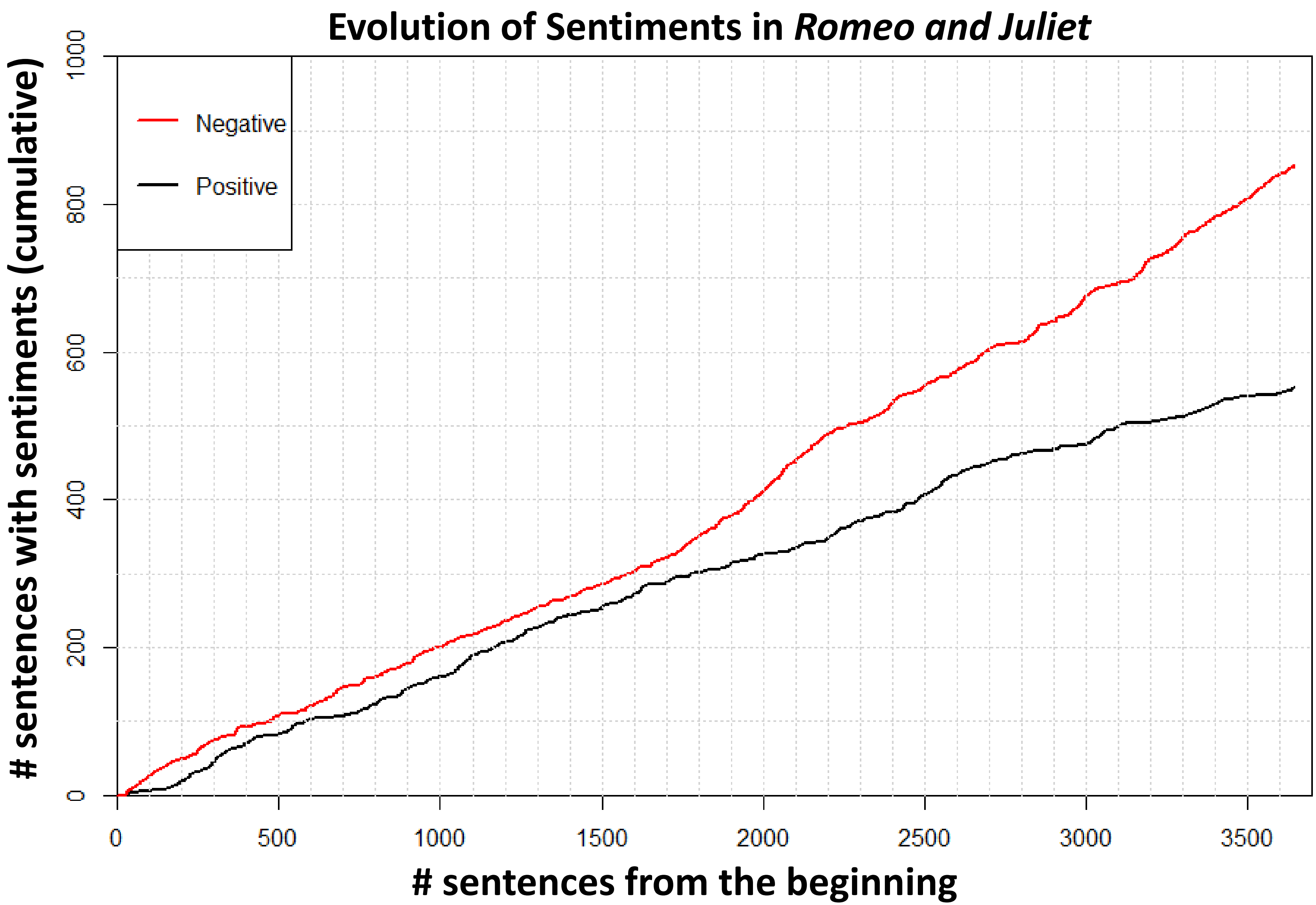
1. Data download in R code: Romeo and Juliet (dplyr's tibble format, cf. data.frame)
2. Data pre-processing: remove auxiliary texts & tokenize
3. Break down sentences to words and remove stop words
4. Assign sentiment (positive/negative/neutral) for each word
5. Raw sentiment score for each sentence (# positive/negative words)

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg
   1:          1     0     0 FALSE FALSE
   2:          2     0     0 FALSE FALSE
   3:          3     0     0 FALSE FALSE
   4:          4     0     0 FALSE FALSE
   5:          5     0     0 FALSE FALSE
  ---
3642:       3642     1     0  TRUE FALSE
3643:       3643     0     1 FALSE  TRUE
3644:       3644     0     1 FALSE  TRUE
3645:       3645     0     0 FALSE FALSE
3646:       3646     0     1 FALSE  TRUE
```

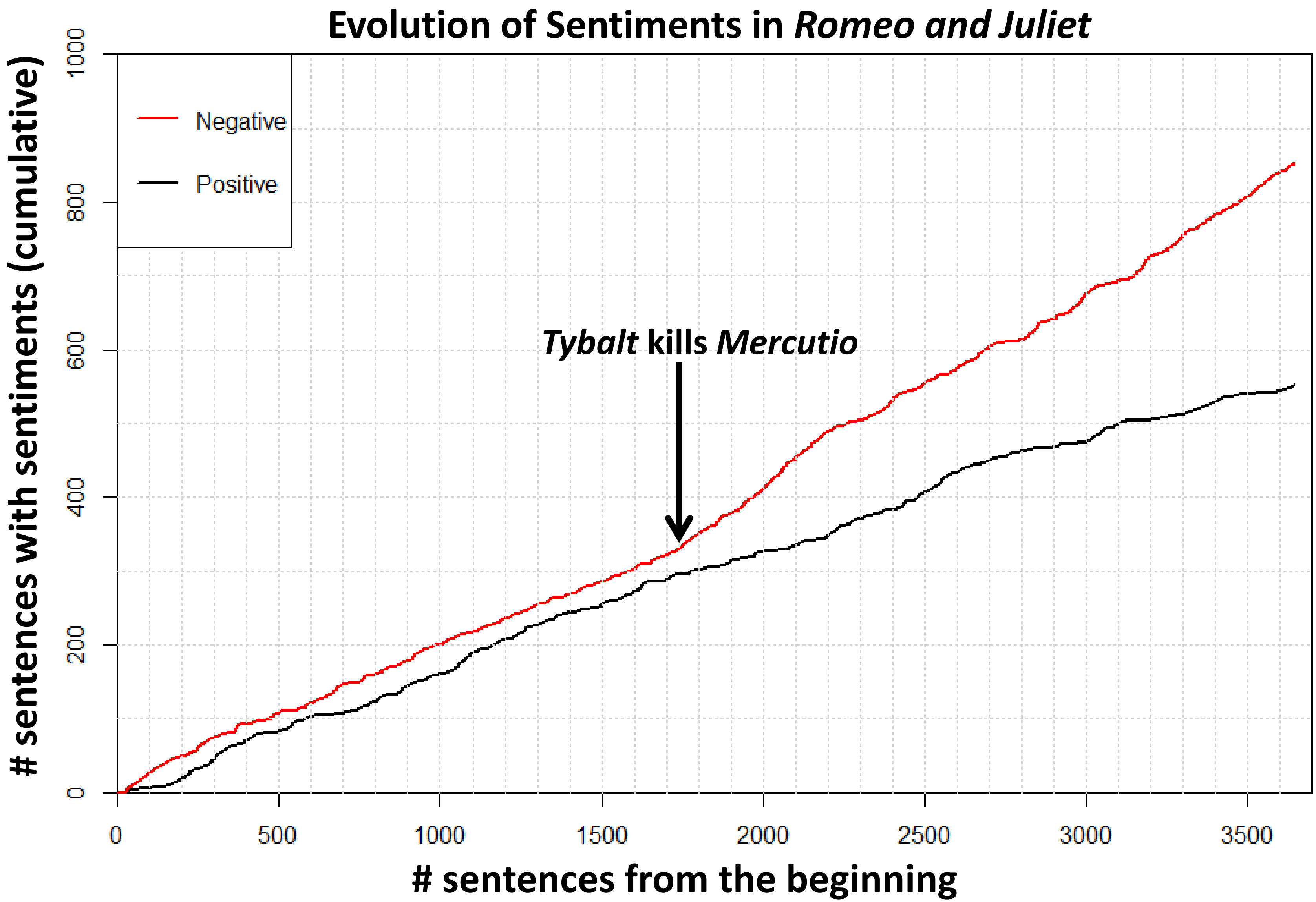
Sentiment	**condition**	**pos**	**neg**	**Count**	**%**
Positive	#pos > #neg	T	F	551	15.1
Negative	#pos < #neg	F	T	853	23.4
Neutral	#pos = #neg	F	F	2242	61.5

Total 3646 sentences

Example #1: Sentiment Analysis



Example #1: Sentiment Analysis



Example #1: Sentiment Analysis

Act III, Scene i.

Mercutio curses both the Capulets and Montagues when he dies

<ROMEO>

Courage, man; the hurt cannot be much.

<MERCUTIO>

No, ’tis not so deep as a well, nor so wide as a church door, but ’tis enough, ’twill serve. Ask for me tomorrow, and you shall find me a grave man. I am peppered, I warrant, for this world. A plague o’ both your houses. Zounds, a dog, a rat, a mouse, a cat, to scratch a man to death. A braggart, a rogue, a villain, that fights by the book of arithmetic!—Why the devil came you between us? I was hurt under your arm.

Example #1: Sentiment Analysis

<u>Q: Changes in sentiment? How much? How significant?</u>

Subset #1: 1st -1734th sentences

Subset #2: 1735th-3646th sentences

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

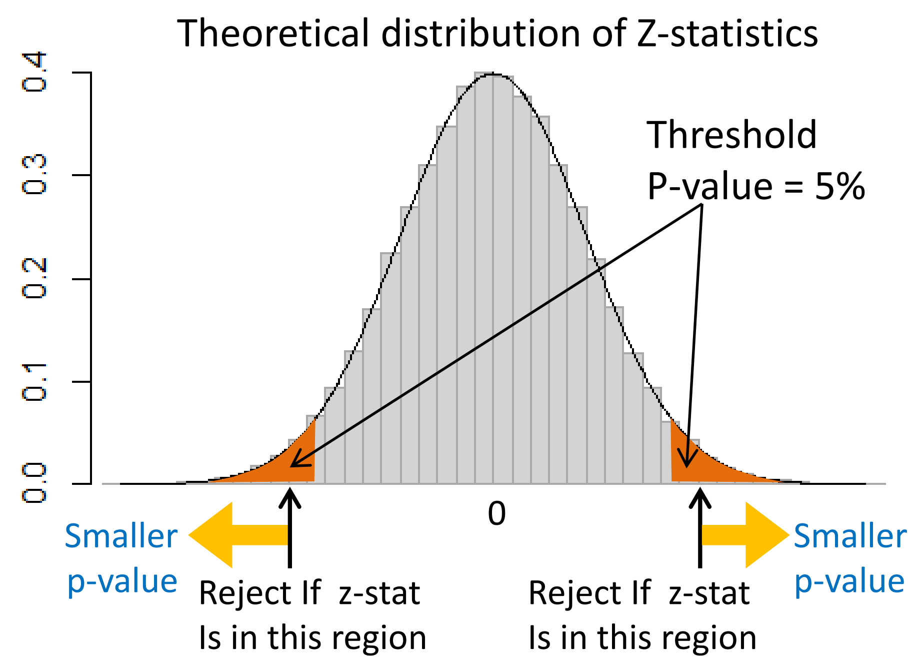
Subset #1: 1^{st}-1734^{th} sentences
Subset #2: 1735^{th}-3646^{th} sentences

Negative sentiment increases and positive sentiment decreases, significantly.

A: Two-sample Binomial tests (normal approximation, two-sides)

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085 ↑	-0.036 ↓
Z-statistic	6.085	-3.051
P-value	1.17×10^{-9}	0.0023

Reject the null since z-stat is too extreme



Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg true.pos true.neg
   1:          1     0     0 FALSE FALSE    FALSE    FALSE
   2:          2     0     0 FALSE FALSE    FALSE    FALSE
   3:          3     0     0 FALSE FALSE    FALSE    FALSE
   4:          4     0     0 FALSE FALSE    FALSE    FALSE
   5:          5     0     0 FALSE FALSE    FALSE    FALSE
  ---
3642:       3642     1     0  TRUE FALSE     TRUE    FALSE
3643:       3643     0     1 FALSE  TRUE     TRUE    FALSE
3644:       3644     0     1 FALSE  TRUE    FALSE     TRUE
3645:       3645     0     0 FALSE FALSE    FALSE    FALSE
3646:       3646     0     1 FALSE  TRUE     TRUE    FALSE
```

Prediction error

Predicted values

True (reference) values

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Uninformative classifier:
Sensitivity + Specificity = 1

Confusion matrix

Predicted \ **True**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

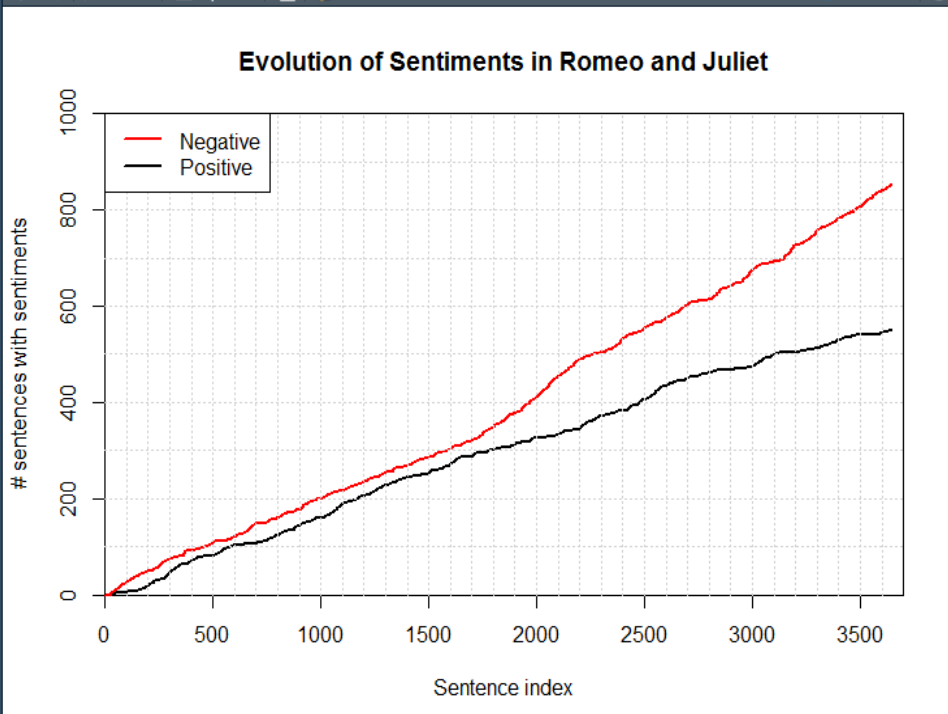
Sensitivity = Recall = Power = True Positive Rate = 762/(762+50) = **0.9384**
= 1 - (Type-II error) = 1 – (False Negative Rate) = 1 – 0.0616

Specificity = True Negative Rate = 2743/(2743+91) = **0.9679**
= 1 - (Type-I error) = 1 – (False Positive Rate) = 1 - 0.0321

This "Positive" means "Negative sentiment", not "Non-negative sentiment"

Demonstration of Analysis with R

```
>
> ##################################################
> ######## Data visualization and analysis #########
> ##################################################
>
> # accumulate # of positive/negative sentences
> score.pos = cumsum(rnj_sentiment_count$pos)
> score.neg = cumsum(rnj_sentiment_count$neg)
>
> # plot of cumulative numbers of sentences with sentiments
> plot(score.pos, type="l", lwd=2,
+      main = "Evolution of Sentiments in Romeo and Juliet",
+      xlab = "Sentence index", ylab = "# sentences with sentiments",
+      xlim=c(0,3700), ylim=c(0,1000), xaxs="i", yaxs="i")
> lines(score.neg, col="red", lwd=2)
> grid(nx=37,ny=10)
> legend("topleft", legend =c("Negative", "Positive"), lty=c(1,1),
+        lwd=c(2,2), col=c("red","black")  )
> # Since MERCUTIO was killed by TYBALT, negative words have dominated.
>
```




```
# generate hypothetical true sentiments of sentences
set.seed(123) # fix the initial seed for random number generation
true.flip = rbinom(nrow(rnj_sentiment_count),1,0.10) # flip sentiment of 10%
rnj_sentiment_count[, true.pos:= as.logical((1-true.flip)*pos + true.flip*neg) ]
rnj_sentiment_count[, true.neg:= as.logical((1-true.flip)*neg + true.flip*pos) ]

# final data set for analysis
rnj_sentiment_count

##################################################
######## Data visualization and analysis #########
##################################################

# accumulate # of positive/negative sentences
score.pos = cumsum(rnj_sentiment_count$pos)
score.neg = cumsum(rnj_sentiment_count$neg)

# plot of cumulative numbers of sentences with sentiments
plot(score.pos, type="l", lwd=2,
     main = "Evolution of Sentiments in Romeo and Juliet",
     xlab = "Sentence index", ylab = "# sentences with sentiments",
     xlim=c(0,3700), ylim=c(0,1000), xaxs="i", yaxs="i")
lines(score.neg, col="red", lwd=2)
grid(nx=37,ny=10)
legend("topleft", legend =c("Negative", "Positive"), lty=c(1,1),
       lwd=c(2,2), col=c("red","black")  )
# Since MERCUTIO was killed by TYBALT, negative words have dominated.



# Two-sample Binomial tests by Normal approximation
N = nrow(rnj_sentiment_count) # the number of sentences
N1 = 1734 # the number of sentences in the first subset
N2 = N - N1 # the number of sentences in the second subset

# Two-sample Binomial test for negative sentences
x = rnj_sentiment_count$neg

sn1 = x[1:N1] # split the negative sentences
sn2 = x[(N1+1):N2] # split the negative sentences

p1.hat = mean(sn1) # estimate for probability in the first subset
p2.hat = mean(sn2) # estimate for probability in the second subset

d.hat = p2.hat-p1.hat # estimate for probability difference

phat.pool = mean(x) # pooled estimate for probability under the null hypothesis
```

Popular Sentiment Lexicon Database for English

MPQA Multi-Perspective Question Answering

SentiWordNet

SenticNet
Helping machines to learn, leverage, love.

- SocialSent (Hamilton et al., 2016)

https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

Structure of Lecture

1. Sentiment Analysis

2. Linguistic Analysis of Senti-words and Sarcasm

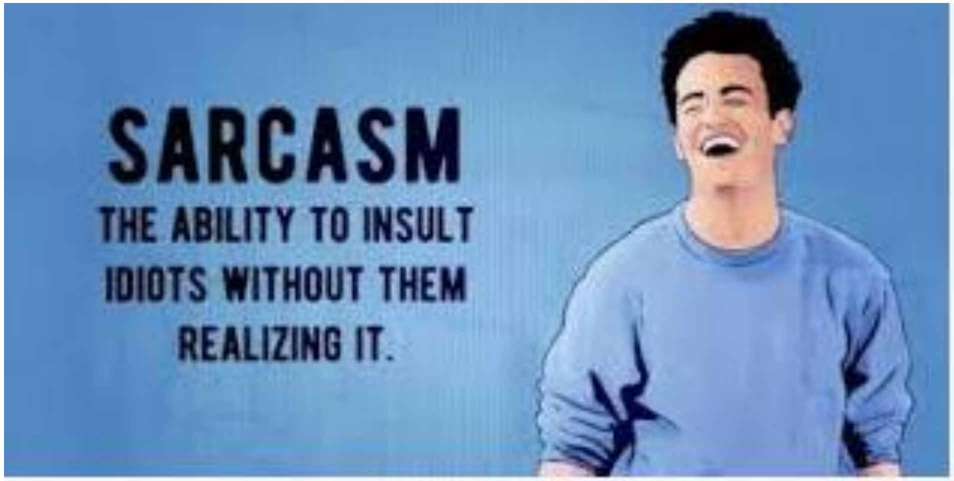
3. Machine Learning approaches for Sarcasm detection

Recall: Improvement methods

- **Sarcasm and Irony**
  - Contextual analysis may be needed to identify sarcastic or ironic statements, where the literal meaning is different from the intended sentiment.
- **Contextual Clues**
  - Understanding the overall context of the text can help in recognizing sentiment, as certain words or phrases may have different connotations depending on the context.

Q: Why sarcasm?

Q: Why sarcasm?



https://houseofrodan.com/products/sarcasm-is-my-only-defense-t-shirt-1?variant=37530524090524
https://m.blog.naver.com/syette828/221168828491

Q: How to detect sarcasm?

https://www.huffpost.com/entry/why-sarcasm-is-so-great_b_7887342

Q: How to detect sarcasm?

https://www.pinterest.com/pin/295126581803212247/

Q: How to detect sarcasm?

A1: Linguistic cues

I. Senti-words in Korean: racial slurs

1. The meaning of racial slurs (Yoon 2015)

- **The expressive dimension of slurs**

(1) That *bastard* Frederic is famous.

(2)

Ku ppalkayngi	nom-un	yumyenghay.
that commie.neg.att	jerk.neg.att-Top	famous

'That commie jerk is famous.'

(1) independence.

- **Expressive items do not participate in denial because they are not part of the descriptive meaning (McCready 2010)**

A:	Ku-nun	kkamtwungi-ya.
	he-Top	black.person.neg.att(=nigger)-Decl
	'He is a nigger.'	
B:	Ani-ya.	
	not-Decl	
	'That's not true.'	
≠	Hukin-un	nappuci-anh-a.
	black.people-Top	bad-Neg-Decl
	'Black people are not bad.'	

(2) Nondisplacebility

Con-un	ku-ka	ppalkayngi-la-nunkesul	al-koiss-ta.
John-Top	he-Nom	commie.**neg.att**-be-C	know-Asp-Decl

'John knows that he is a commie.'
But John respects the N. Korean.
#But I respect the N. Korean.

- **Certain racial slurs exhibit quite systematic variations in terms of the negative attitude**

(22) The expressive index (EI) **I** for 'black people' in Korean

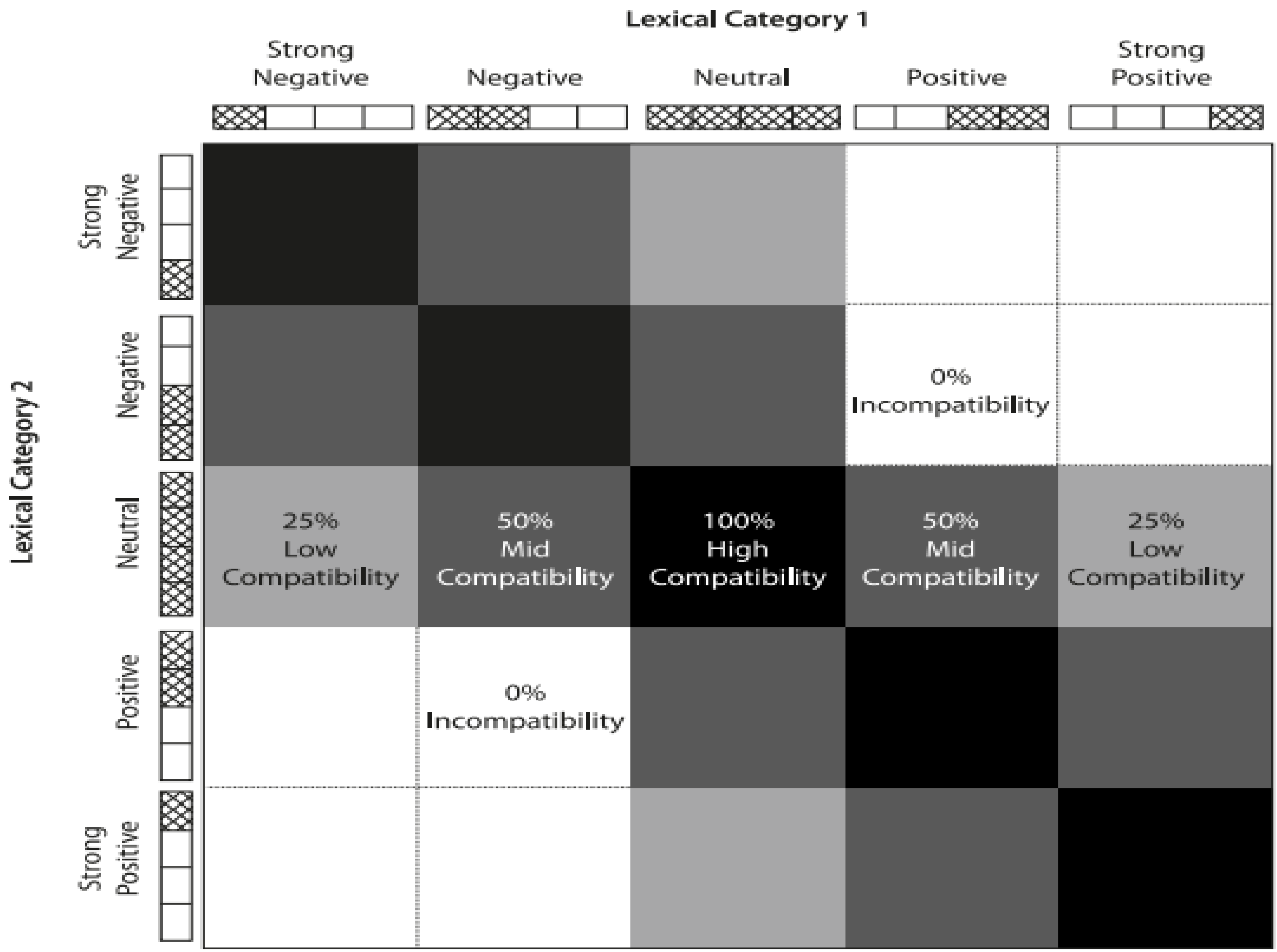
a. *hukin*: [−1,1]: a neutral descriptive term, 'black people'

b. *hukhyeng*: [0,1]: a rather friendly positive term, 'lit. black brother', 'afro bro'

c. *kemtwungi*: [−1,0]: a weak negative term, 'lit. darkie'

d. *kkamtwungi*: [−1,−.5]: a strong negative term, 'lit. blackie', 'nigger'

Compatibility Condition Model (CCM; Yoon 2015)



Evidence 1: compatibility condition in Korean

(26) a. Kunye-nun ✔alumtawun /# phyengpemhan /# hyungchukhan
she-TOP beautiful / normal / hideous
cathay-lul tulenayss-ta.
figure.POS-ACC revealed-DECL

b. Kunye-nun ✔alumtawun / ✔phyengpemhan / ✔hyungchukhan
she-TOP beautiful / normal / hideous
mosup-ul tulenayss-ta.
figure.NEU-ACC revealed-DECL

c. Kunye-nun #alumtawun / #phyengpemhan / ✔hyungchukhan
she-TOP beautiful / normal / hideous
molkol-ul tulenayss-ta.
figure.NEG-ACC revealed-DECL
'She revealed a beautiful/normal/hideous figure.'
(Giannakidou and Yoon 2011: 645, (67))

Evidence 2: compatibility condition in Korean

Table 1
The compatibility of ethnic slurs and expressive nouns.

epithets for 'guy'	slurs			
	ppalkayngi 'commie' *kkamtwungi* 'nigger/blackie' [−1,−.5]	*kemtwungi* 'darkie' [−1,0]	*hukin* 'black people' [−1,1]	*hukhyeng* 'black brother' [0,1]
saykki 'bastard' [−1,−.5]	high compatibility			
nom,casik 'jerk' [−1,0]	mid compatibility			
namca 'man/guy' [−1,1]	low compatibility			
ssi 'Mr./Ms.' [0,1]	incompatibility			
pwun, nim 'sir' [.5,1]				

Table 6
Co-occurrences of slurs and expressive nouns in Korean National Corpus *The Sejong Corpus* (results attained by the search program the *kkokkoma*).

epithets for 'guy'	slurs			
	ppalkayngi 'commie', *kkamtwungi* 'blackie/nigger' [−1,−.5]	*kemtwungi* 'darkie' [−1,0]	*hukin* 'black person' [−1,1]	*hukhyeng* 'black brother' [0,1]
saykki 'bastard' [−1,−.5]	36	0	0	0
nom/casik 'jerk' [−1,0]	13	2	0	0
namca 'guy' *sonyen* 'boy' [−1,1]	0	0	11	0
ssi 'Mr./Ms.' [0,1]	0	0	0	0
pwun, nim'sir' [.5,1]	0	0	0	0
total occurrences of slurs	257	80	32	612

Evidence 3: compatibility condition in Korean

Table 2
Compatibility of slurs and case markers.

case markers	slurs			
	ppalkayngi 'commie' *kkamtwungi* 'nigger/blackie' [−1,−.5]	*kemtwungi* 'darkie' [−1,0]	*hukin* 'black person' [−1,1]	*hukhyeng* 'black brother' [0,1]
ttawi-ka 'Nom.ANTI.HON' *ttawi-eykey* 'Dat.ANTI.HON'[−1,−.5]	high compatibility	mid compatibility		
ka 'Nom.NEU' *eykey* 'Dat.NEU' [−1,1]	low compatibility			
kkeyse 'Nom.HON' *kkey* 'Dat.HON' [.5,1]	incompatibility			

Evidence 4: compatibility condition in Korean

Table 3
Compatibility of slurs and (anti-)honorific markers.

(anti-)honorific markers	slurs			
	ppalkayngi 'commie' *kkamtwungi* 'nigger/blackie' [−1,−.5]	*kemtwungi* 'darkie' [−1,0]	*hukin* 'black person' [−1,1]	*hukhyeng* 'black brother' [0,1]
-*peli* 'NEG.ATT' [−1,−.5]	high compatibility	mid compatibility		
ø 'NEU.ATT' [−1,1]	low compatibility			
-*si* 'SUBJ.HON' [.5,1]	incompatibility			

Sentiment Analysis of Taste terms

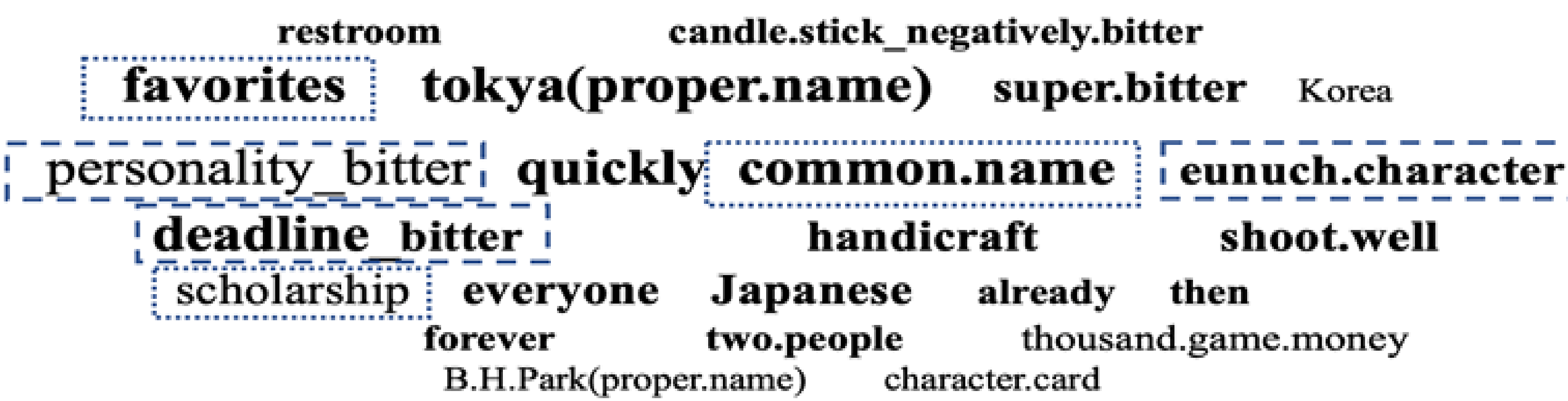
Fig. 2. Word cloud of *ssapssal* ‘*bitter.pos.att*’

(Translation in English)

Sentiment Analysis of Taste terms

Fig. 11. Word cloud of *ssupssul* 'bitter.neg.att'

(Translation in English)



Sarcasm detection 1:
Mismatch of positive and negative sentiments

https://www.lianedavey.com/sarcasm/

Juxtaposition of opposite attitudes? sarcasm, irony, or hyperbole

Sarcasm detection 1:

Mismatch of positive and negative sentiments

(42) a. Ppalkayngi-pwun: 6490 hits on Google search (June 27, 2014)
commie.**neg.att**-sir.**hon**
b. Ppalkayngi-nim: 32,700 hits on Google search (June 27, 2014)
commie.**neg.att**-sir.**hon**
'The ($_{CI}$dishonorable) commie, the ($_{CI}$honorable) being.'

(43)	Ne-na	cal-ha	sey-yo!
	you.**anti.hon**-or.**anti.hon**	well-do	subj.**hon**-Decl.**hon**
	'Mind your own ($_{CI}$bloody) business!'		

Flip-flop of bipolar emotional index: strengthened emotion or intimacy

(44) That fucking bastard Burns got promoted again!
(45) That's really fucking brilliant!

(46) Hiya, bitches! (to extremely close friends)

Sarcasm detection 2 :
Mismatch of honorific and anti-honorific attitudes (E.H. Oh, p.c.)

https://getyarn.io/yarn-clip/5e49a740-955a-4722-94e0-fcb96aead95e
https://tenor.com/search/your-highness-gifs

Sarcasm detection 2 :

Mismatch of honorific and anti-honorific attitudes

(1) "Oh, please, Your Highness, grace us with your infinite wisdom.“

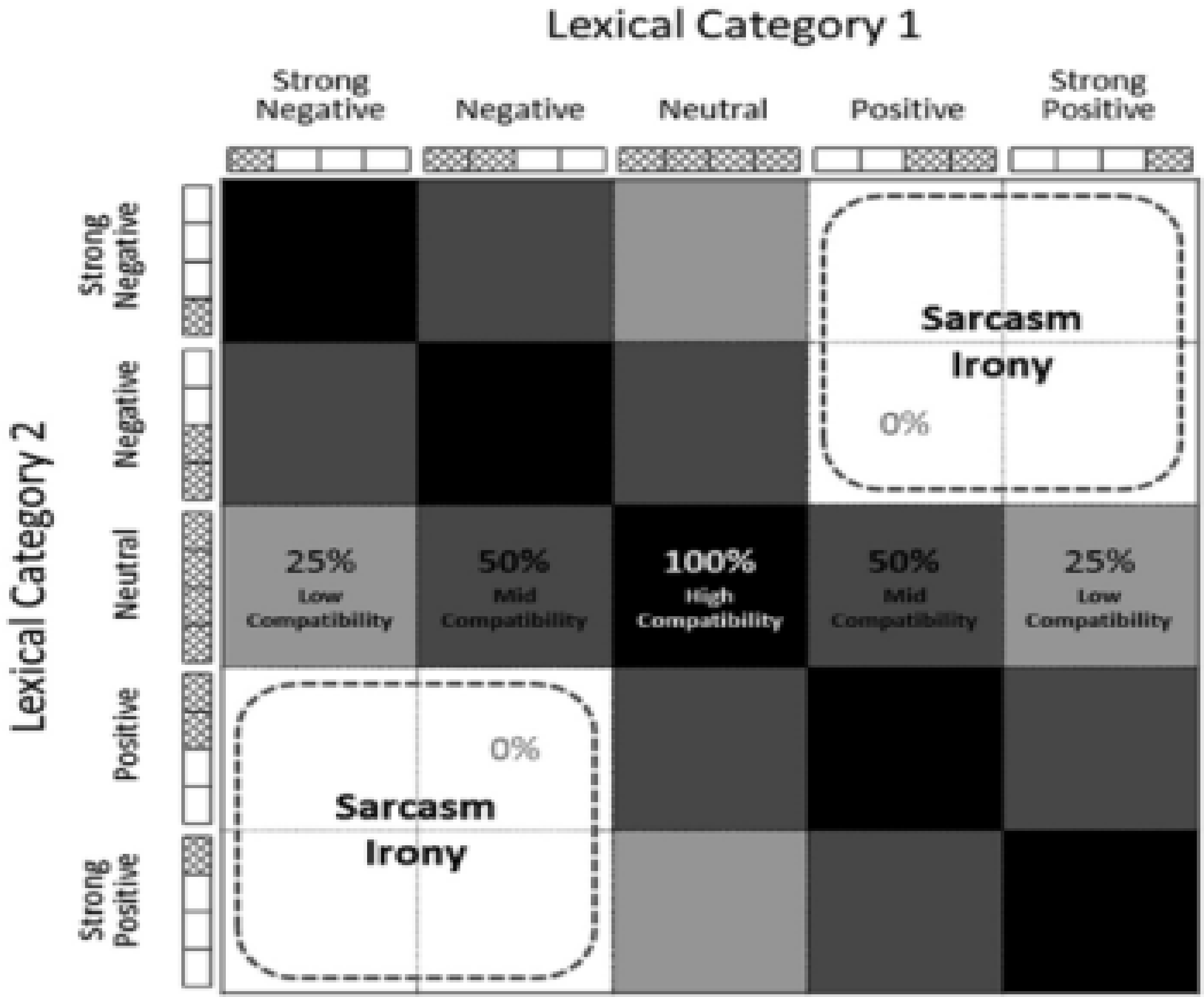
(2) "I'm truly honored to be in the presence of the great and mighty Professor Know-It-All.“

(3) "Well, Captain Obvious, thank you for enlightening us with your profound insight.“

(4) "I bow to you, Master of Punctuality, for gracing us with your timely presence.“

(5) "Your culinary skills are truly unmatched, Chef Extraordinaire. I couldn't even tell it was takeout.“

Sarcasm and Irony areas in CCM (Yoon 2015)



Sarcasm detection 3: punctuations

Single & Fabulous!   vs.   Single & Fabulous?

https://www.televisionofyore.com/recaps-of-sex-and-the-city/sex-and-the-city-season-2-episode-4

Sarcasm detection 3: punctuations

(1) Well, that's just great.

(2) Oh, of course you're right!

(3) Brilliant! You locked us out of the car again.

(4) You're sooo funny...

(5) Oh, that's just what I needed today: more work.

(6) Oh, I totally believe you.

Sarcasm detection 4: interjections

e.g. yeah, ah, oh

https://tenor.com/search/sarcastic-yeah-gifs
https://twitter.com/ourinspiring/status/944529527648157696
https://makeameme.org/meme/oh-yeah-8jf3sy

Sarcasm detection 4: interjections

e.g. yeah, ah, oh

https://gifdb.com/gif/oh-really-sarcastic-tamar-braxton-38v4ehqlxxs78ezj.html
http://www.quickmeme.com/meme/3rl87d
https://makeameme.org/meme/oh-yeah-8jf3sy

Sarcasm detection 4: interjections

(1) Oh, great. Another flat tire.

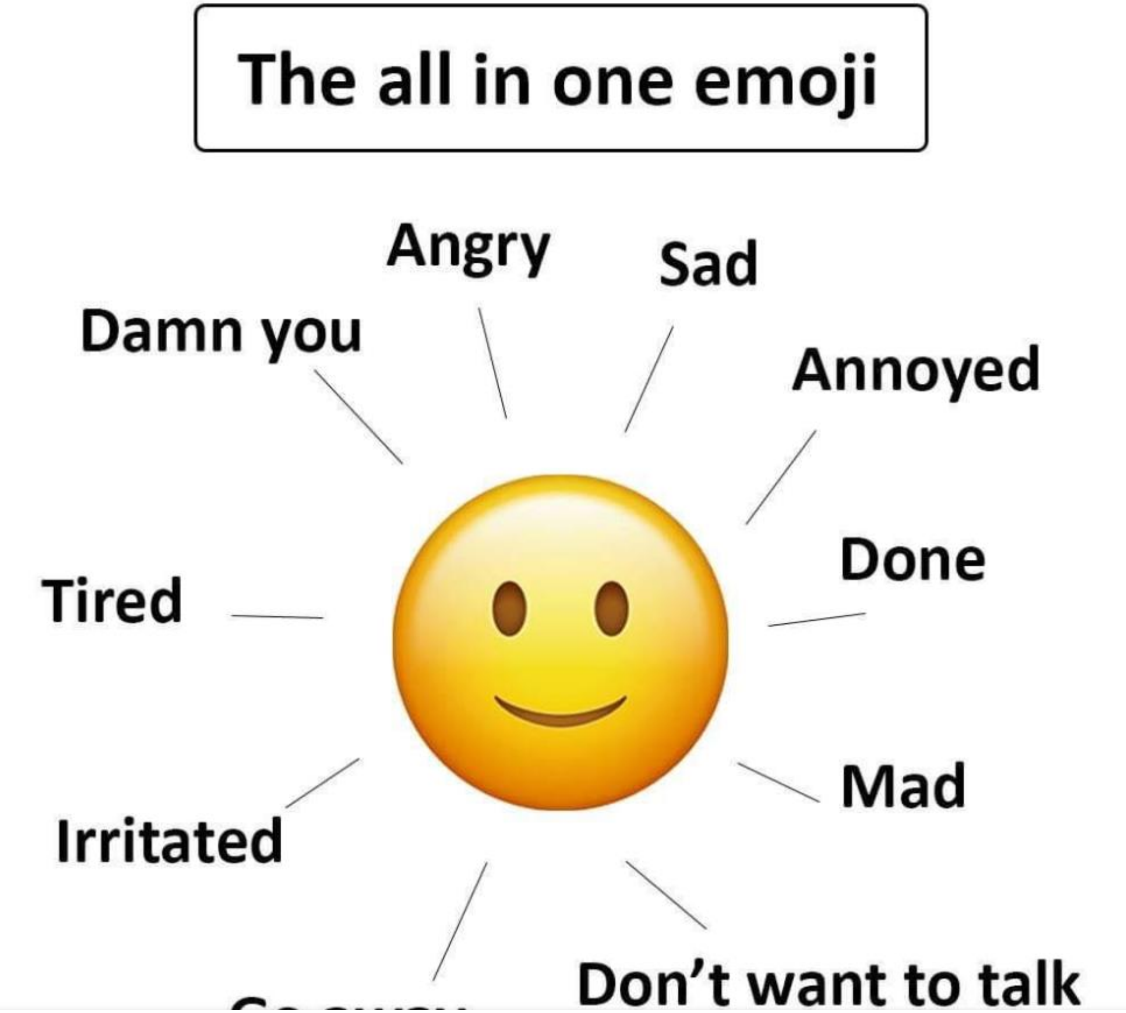
(2) Yeah, because that's such a brilliant idea.

(3) Ah, of course you're right.

(4) Yeah, I totally believe that happened.

(5) Ah, the wonders of bureaucracy.

Sarcasm detection 5: emoticons



https://www.facebook.com/SarcasmLol/photos/a.1533243200337792/5686346485027422/?type=3

Sarcasm detection 5: emoticons

(1) Thanks for your help :)

(2) Oh, you're a real genius. :|

(3) Oh, you're sooooo funny! ;P

(4) Sure, I believe you. -_-

(5) Thanks for being so helpful! ;-)

(6) Oh, I'm sure you're right. :/

Sarcasm detection 6: ML negation

Sarcasm detection 6: ML negation

(2) *Emphatic flavor of MN*

a. Around here we *don't* LIKE coffee—we LOVE it.
b. She *doesn't* sell INsurance—she sells inSURance.
c. I'm *not* HIS brother—he's MY brother!
d. Mozart's sonatas were for piano and violin, *not* for violin and piano.
e. I'm *not* happy with the plan, I'm ecstatic!
f. You *didn't* eat some of the cookies, you ate them all!

(3) *Hallmarks of MN*

a. Metalinguistic negations are standardly used as a rejoinder to a previous utterance.

b. There is a certain prosodic pattern commonly associated with metalinguistic negation.

Six semantic sources of metalinguistic negation markers (Yoon, ms)

1. Objection due to inappropriateness

(16) I'm *not* happy with the plan, I'm ecstatic! [English]

(17) a. *No way* some men are chauvinists – all men are chauvinists.

b. *Nonsense* I managed to trap two mongeese. I managed to trap two mongooses. (Drozd 2001: 56-57)

(19) a.*Swuni-nun yekan *an* yeypputa. [Korean]

Swuni-TOP	ordinarily	not pretty		
b. Swuni-nun	yekan	yeypp-*ci*	*an*-h-ta.	(Taytanhi yeyppu-ta.)
Swuni-TOP	ordinarily	pretty	not-be-DC	extremely pretty-DC

'Swuni is not ordinarily pretty.' ('She is extremely pretty.')

Six semantic sources of metalinguistic negation markers

2. Objection due to unusuality or abnormality

(21)	A: Estás	um pouco	preocupado?	[European Portuguese]
	are-2SG	a little	worried	
	'Are you a little worried?'			
	B: Estou	*lá/agora*	um pouco	preocupado.
	am	MN-marker	a little	worried
	'I'm not a little worried, I am worried sick.'			

(20)	A: Ne	cokum	kekcengtoy-ni?	[Korean]
	are-2SG	a little	worried-Q	
	'Are you a little worried?'			
	B: Cokum	kekceengtoy-*nyani!*	Kekceengtoye-cwu-keyss-e!	
	a little	worried-exclamatory.Q	worried-die-will-Decl	
	'I'm not a little worried, I am worried sick!'			

Six semantic sources of metalinguistic negation markers

3. Objection due to unidentifiability-driven non-referentiality

(22) JOHN KNOW ANSWER *!part:indef!* [ASL]
'John knows the answer! (How could you have thought he wouldn't?)'

(25) A: O governo vai baixar os impostos. [European Portuguese]
the government goes lower the taxes
'The government is going to lower the taxes.'
B: a. *Qual quê*!
which what

Six semantic sources of metalinguistic negation markers

3. Objection due to unidentifiability-driven non-referentiality

(23) A: Ce kaswu cham mesiss-ta! [Korean]
that singer really cool-Decl
'That singer is really cool!'
B: Mesiss-ki-nun *mwusun!*
cool-NMLZ-Top what/which
'Lit. What/which, he's cool!' (Intended: 'He's cool, my eye!)

(24) A: Ce kaswu cham mesiss-ta!
that singer really cool-Decl
'That singer is really cool!'
B: Mesiss-ki-nun *etey!*
cool-NMLZ-Top where
'Lit. Where, he's cool!' (Intended: 'He's cool, my eye!)

Six semantic sources of metalinguistic negation markers

4. Objection due to non-existence-driven non-referentiality

(33) A: He found proofs that clinched the argument. [English]
B: He found proofs that clinched the argument *nothing*. (Bolinger 1977: 45)

(34) A: O governo vai baixar os impostos. [European Portuguese]
the government goes lower the taxes
'The government is going to lower taxes.'
B: a. Vai *nada* (baixar os impostos).
goes nothing lower the taxes
b. Vai baixar os impostos *nada*.
goes lower the taxes nothing
'Like hell (it is going to lower taxes)!

Six semantic sources of metalinguistic negation markers

4. Objection due to non-existence-driven non-referentiality

(28) A: Ce kaswu cham mesiss-ta! [Korean]
that singer really cool-Decl
'That singer is really cool!'
B: Mesiss-ki-nun *kayppwul!*
cool-NMLZ-Top dog.horn
'Dog's horn, he's cool!'

5. Objection due to unworthiness or dispreference

(35) A: You still love me. [English]
B: *Like hell* I still love you. (Horn 1989: 402)

(36) "Relax, pal," Mark said. "Some days are better than others."
Todd gulped his beer and said, "Relax *my ass*. We've been doing this crap for a month and it feels like I'm carrying the load here." (John Grisham, *The Rooster Bar*, 203) (Martins 2019:(8))

(37) Deberías desculparte por tu comportamiento. [Spanish]
should.1SG apologize.REFL.1SG for your behaviour
'You should apologize for your behavior.'
B: ¡*Una mierda* voy (yo) a disculparme!
a shit go I to apologize.REFL
'Like hell I will apologize!' (Olza Moreno 2017: 47)

Six semantic sources of metalinguistic negation markers

5. Objection due to unworthiness or dispreference

(42) A: Ce kaswu cengmal mesiss-ta! [Korean]
that singer really cool-Decl
'That singer is really cool!'
B: Ce kaswu cengmal mesiss-tani, *wuski-ney!*
that singer really cool-Excl laughable-Excl
'That's laughable, that singer is really cool!'

(43) B': Mesiss-ki-nun *(mwusun)* *{elecwuk-ul/mangh-al/yempyeng/wulacil/nimilel}!*
cool-NMLZ-Top what/which freezing.to.death/going.bust/epidemic/damn/damn
'Lit. Freezing to death/going bust/epidemic/damn/damn, he's cool!'

6. Objection due to insincerity

(44) *Yeah right* you ate some of the cookies. You ate all of them! (Drozd 2001: 56-57)

(45) [Korean]

A: Ce kaswu	cengmal	mesiss-ta!	
that singer	really	cool-Decl	

'That singer is really cool!'

B: Ce kaswu	cengmal	mesiss-nun-ke,	*coaha-ney!*
that singer	really	cool-Top-NMLZ	like.it-Excl

'That singer is really cool?! Oh, you'd like that, do ya?'

B': Ce kaswu	cengmal	mesiss-tani,	*nolkoiss-ney!*
that singer	really	cool-Excl	enjoy.it-Excl

'That singer is really cool?! Oh, you're enjoying that, aren't ya?'

(46) [French]

A: En réalieté, J'ai un super pouvoir.
'Actually, I have a superpower.'
B: Tu parles!
'Go on (Lit. you speak)!'

Sarcasm detection 5: ML negation

Table 1. The genesis of Metalinguistic Negation markers (Yoon, under revision)

Six classes of MN markers	Semantic sources for MN markers	MN markers
Class 1	inappropriateness	**Regular negation markers:** not, no way, nonsense in English; dhen 'not,' oxi 'no' in Greek; ci anh 'not' (external negation) in Korean
Class 2	unusuality or abnormality	**Emphatics markers:** nyani 'exclamative' in Korean; lá 'lit. there', agora 'lit. now' in European Portuguese
Class 3	unidentifiability-driven non-referentiality	**Anti-specificity markers:** part:indef 'someone or other' in ASL; mwusun 'which,'etey 'where,' mwusun + depreciatives in Korean dialects; qual 'which,' qual quê 'which what,' o quê 'the what' in European Portuguese; qué … ni qué + depreciatives 'what … nor what' in Peninsular Spanish
Class 4	non-existence-driven non-referentiality	**Non-existence markers**: kayppwul 'dog's horn,' nonexistent event-describing expressions (e.g. 'sounds like a ghost's peeling and eating grains', 'sounds like a dog's nibbling on grass,' 'sounds like a maki roll's side popping,' 'sounds like an earthwarm's yawning,' 'sounds like a frog's side-kicking') in Korean; nothing, nothing of the sorts in English; nada 'nothing' in European Portuguese; minga 'no/nothing' in Rioplatense Spanish
Class 5	unworthiness or dispreference	**Depreciative markers:** the hell, like hell, my ass, my eye, bullshit, poppycock, fiddlesticks, your old man, like fun, like fudge, yo' mama, my foot, X shma/schma-X in English; una leche 'a blow/hit', (unas/las) narices 'a/the noses', una mierda 'a shit', los cojones 'the balls' in Spanish; uma ova 'a fish roe', o tanas (obscure meaning), o caralho/o caraças ('penis' (slang)), uma merda 'a shit' in European Portuguese; qué … ni qué {narices/cojones/coño/mierda/leche} 'what … nor what noses/balls/cunt/shit/blow' in Peninsular Spanish (repeated from class 3); mon oeil! 'my eye' in French; wuskiney 'laughable,' {elecwuk-ul/mangh-al/yempyeng/wulacil/ nimilel}'freezing.to.death/going.bust/epidemic/damn/damn' in Korean dialects
Class 6	insincerity	**Irony markers:** coahaney 'oh, you'd like that, do ya?', nolkoissney 'oh, you're enjoying that, aren't ya? in Korean; yeah right, yeah yeah, oh yeah in English, tu parles! 'go on (lit. you speak)' in French

Sarcasm detection 5: ML negation

Table 2. Four types of MN and six classes of MN markers

Four types of MN	Six classes of MN markers	Semantic sources for MN markers	MN markers
Type I: Appropriateness assessment MN	Class 1	inappropriateness	Regular negation markers: not, no way, nonsense in English, ci an 'not' in Korean, etc.
Type II: Emphatic denial MN	Class 2	unusuality or abnormality	Emphatics markers: lá 'lit. there' in European Portuguese, nyani 'exclamative' in Korean, etc.
	Class 3	unidentifiability-driven non-referentiality	Anti-specificity markers: part:indef 'someone or other' in ASL, mwusun 'which' in Korean, etc.
Type III: Negative emphatic denial MN	Class 4	non-existence-driven non-referentiality	Non-existence markers: nothing in English, nada in Spanish, kayppwul 'dog's horn' in Korean, etc.
	Class 5	unworthiness or dispreference	Depreciative markers: the hell, like hell, X shma/schma-X in English, wuskiney 'laughable,' elecwuk-ul 'freezing to death' in Korean, etc.
Type IV: Inory MN	Class 6	insincerity	Irony markers: yeah right in English, coahaney 'oh, you'd like that, do ya?', nolkoissney 'oh, you're enjoying that, aren't ya?' in Korean, tu parles! 'go on' in French, etc.

Q: How to detect sarcasm?

A1: Linguistic cues

A2: Contextual cues: Amazon rating

Research example: Oh & Yoon (in progress)

DETECTING SARCASM
IN MOVIE AND TV SHOW REVIEWS

Research example: Oh & Yoon (in progress)

Introduction : Detecting Sarcasm in Amazon Movies and TV Reviews

- Using a dataset of Amazon Movies and TV Reviews
- Python's *Transformers* library will be used to conduct sentiment analysis on the reviews.
- If reviews with a rating score under 3 are labelled as positive, they will be categorized as potentially sarcastic.
- Using tokenization to analyze the word frequency in these reviews.
- The goal is
    - to identify which words or phrases are most commonly associated with sarcasm in Movies and TV reviews
    - to provide insights into the expression of sarcasm in online reviews more broadly.

Research example: Oh & Yoon (in progress)

Overview of the dataset

```
json_data
```


```
 {'overall': 5.0,
  'vote': '3',
  'verified': True,
  'reviewTime': '02 18, 2013',
  'reviewerID': 'A2VHSG6TZHU1OB',
  'asin': '0001527665',
  'style': {'Format:': ' Amazon Video'},
  'reviewerName': 'Ken P',
  'reviewText': 'Having lived in West New Guinea (Papua) during the ti
me period covered in this video, it is realistic, accurate, and convey
s well the entrance of light and truth into a culture that was for cen
turies dead to and alienated from God.',
  'summary': 'Realistic and Accurate',
  'unixReviewTime': 1361145600},
```

Research example: Oh & Yoon (in progress)

Methods and Techniques

```
reviews_chunked = list_chunk(reviews, 100)
ratings_chunked = list_chunk(ratings, 100)
```

- Systematic Sampling Method

  in order to sample the data evenly

```
classified = []

for i in reviews_chunked:
    text = str(i[0])
    if len(text) < 512:
        classified=classified+classifier(text)
    else:
        text = " "
        classified=classified+classifier(text)
```

- Delete reviews with more than 512 letters,

  because the classifier cannot handle them

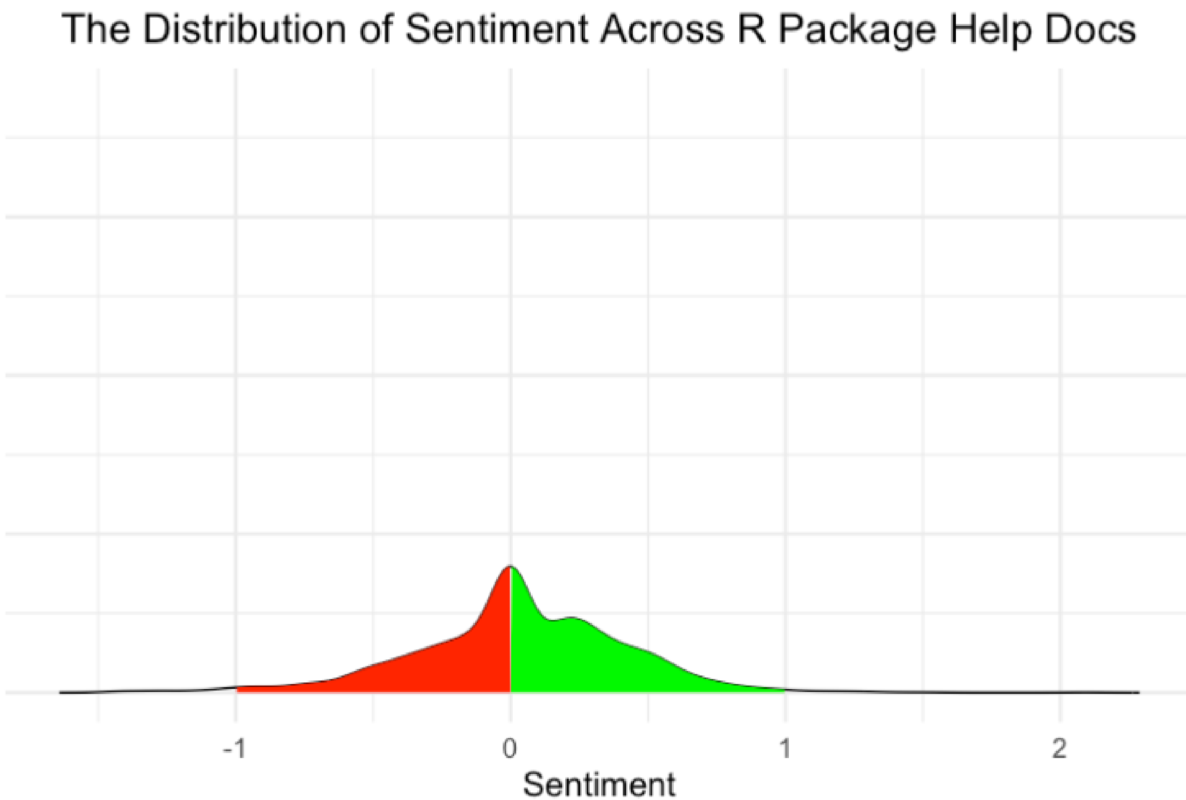
Obtained Results

```
results_df
```

	label	score	rating	review
0	Positive	0.891905	1.0	It is a shame that a kids movie has God d*** u...
1	Positive	0.999997	1.0	iT'S AMAZING THAT THEY WOULD RELEASE A MOVIE A...
2	Positive	0.991348	1.0	Who pays $11,000 for a silly cartoon? Great w...
3	Positive	0.999964	1.0	It was horrible. Hard target 2 is way better
4	Positive	0.986253	2.0	I bought it since critics rated it as the best...
...	...	...	...	...
635	Positive	0.999999	2.0	I watched this based on the other reviews - am...
636	Positive	1.000000	1.0	The fact this movie won picture of the year sh...
637	Positive	0.572173	1.0	Beautiful scenery and a good actor wasted. I ...
638	Positive	0.999550	2.0	I cannot believe ice cube agreed to play this ...
639	Positive	0.844736	2.0	The movie "Outsourced" was wonderful! This si...

Research example: Oh & Yoon (in progress)

Data Preparing : RStudio



- Global Environment
- package:sentimentr
- package:lubridate
- package:forcats
- package:purrr
- package:readr
- package:tidyr
- package:tibble
- package:tidyverse
- package:dplyr
- package:stringr
- package:tidytext
- package:ggplot2
- package:syuzhet
- package:wordcloud
- package:RColorBrewer
- package:SnowballC
- package:tm
- package:NLP
- package:stats
- package:graphics
- package:grDevices
- package:utils
- package:datasets
- package:methods
- package:base

Research example: Oh & Yoon (in progress)

Data Preprocessing

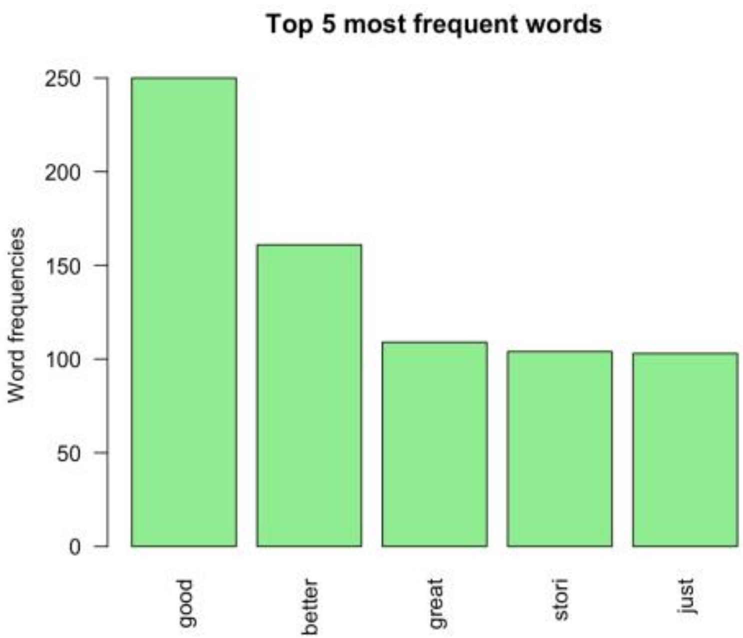
```
> bounded_sentences %>% filter(between(sentiment,0,1)) ->  pos_result
>

> head(pos_result)
                                                                                                                   text
1: If you are a Christian family and want to keep the eyes and ears of your little ones pure, you will want to skip this movie.
2:                         The Sound of music, Fiddler on the roof, The music man, would all be better alternatives to this one.
3:         iT'S AMAZING THAT THEY WOULD RELEASE A MOVIE AS BAD AS THIS AND IT'S TRUELY AMAZING THAT DENZEL WOULD STAR IN IT.
4:                                                                                      Great way to get people to look at it.
5:                                                                                                                 Nice Joke.
6:                                                                                               Hard target 2 is way better
   element_id sentence_id word_count  sentiment characters
1:          1           4         26 0.03922323        124
2:          1           5         19 0.45883147        101
3:          2           1         22 0.18122061        113
4:          3           2          9 0.16666667         38
5:          3           3          2 0.77781746         10
6:          4           2          5 0.24596748         27
```

Research example: Oh & Yoon (in progress)

Results

```
> head(dtm_d, 5)
         word freq
good     good  250
better better  161
great   great  109
stori   stori  104
just     just  103
```



Word Bar Chart

Word cloud plot

Research example: Oh & Yoon (in progress)

Extraction of Sentences with 'good'

```
> filtered_sentences <- pos_result$text[sapply(pos_result$text, function(sentence) grepl("good", sentence))]
> print(filtered_sentences)
 [1] "It was so good and insightful."
 [2] "thru his machine for a good clean up."
 [3] "Well, it was good for what it was."
 [4] "The car chase and the cops are about as good as a good ole Keystone Cops comedy."
 [5] "The fights are good and  exciting, but the whole film itself is forgettable."
 [6] "kind of good in a cultish sort of way"
 [7] "How this got so many good reviews is beyond us."
 [8] "Delivery and quality of the picture was very good."
 [9] "The Blu-Ray looks pretty good and"
[10] "Perhaps it was a good film in it's day."
[11] "Very good movie, watched it with our daughters and granddaughters!"
[12] "Paltrow did a good acting job, and looked stunning."
[13] "The NWO coming back was pretty good, but predictable, and the main event was just half an our full of
chest slaps, with below average wresling."
[14] "Soooo bad and sooo good."
[15] "this film isn't one of Eddie's best or even that good or even pretty good."
[16] "nobody i know talks about this film and with good reason it's not  something you choose to remember."
[17] "If you liked the movie, good for you."
[18] "I thought that Tom and his Karate were very good."
[19] "A laugh now and than is good."
[20] "Danial Day Lewis's version was really good ."
```

Research example: Oh & Yoon (in progress)

Extraction of Sentences with 'great'

```
> filtered_sentences1
 [1] "happy with all orders -  packaging was great  no complaints here"
 [2] "Get the Secret instead - it is great!"
 [3] "The only positive thing about this movie is that it serves as a great reminder that anyone who came of
age in the 80's and yearns for that simple era needs to be ignored as the simpleton they are."
 [4] "The scenery and sets were great and Ms Hepburns costumes were awesome."
 [5] "A classic movie setting a new genre by Mike Nichols and introducing a new talent Dustin Hoffman,Still g
reat to see  Anne Bancroft"
 [6] "The talent is great, no doubt."
 [7] "It's a great looking film...."
 [8] "RICHARD MATHESON great author."
 [9] "I would recommend these other great foreign martial arts movies with a better storyline such as:"
[10] "Otherwise, it's a great cure for insomnia."
[11] "The special effects are nice, and Jude Law gives a great performance, but they weren't enough to overco
me the mind numbing effects of the story itself."
[12] "ok\" than a great revelation."
[13] "He is a great actor!"
[14] "The cast was great."
[15] "i started the pregnancy with great cardiovascular health due to running, so i wanted to gain some stren
gth..."
[16] "My Review is on the Dawn of the Dead (Unrated Director's Cut) [Blu-ray] (2004), great movie, looks good
in HD."
[17] "is just as said to be great"
[18] "The stars are great as expected."
[19] "Story was great  - loved it."
[20] "Otherwise great story to reinforce family values"
```

Research example: Oh & Yoon (in progress)

Extraction of Sentences with 'better'

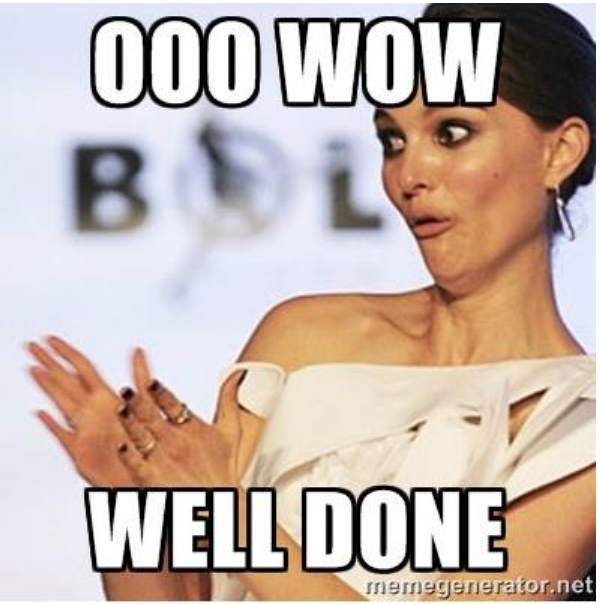
```
> filtered_sentences2 <- pos_result$text[sapply(pos_result$text, function(sentence) grepl("better", sentenc
e))]
>
> filtered_sentences2
  [1] "The Sound of music, Fiddler on the roof, The music man, would all be better alternatives to this one."
  [2] "Hard target 2 is way better"
  [3] "a very bad print the vhs tape looks much better"
  [4] "The description seemed better than the movie."
  [5] "I remember seeing this as a kid and it was a lot better."
  [6] "Some people love this one and I expected better than it delivered."
  [7] "Fine script and far better than usual Hollywood garbage."
  [8] "At least it was better than this!"
  [9] "Part 1&2 were better than this."
 [10] "Was much better when I was 11 years old."
 [11] "I watch all the series and they were much better than this long movie."
 [12] "The franchise got better with each sequel."
 [13] "Naomi Watts looked better with shorter hair in Peter Jackson's KING KONG."
 [14] "Clive Owens looked better in KING ARTHUR."
 [15] "This movie looks better sitting in it's DVD case than playing in your DVD player."
 [16] "My recollections of this movie were far better than the current realty."
 [17] "I was real dissapointed with it and thought it could've been much much better than it was."
 [18] "I enjoy Ben Affleck's acting but he can do so much better than this drivel."
 [19] "the whole found footage thang was done much better in, \"cannibal holocaust.\""
 [20] "I liked the original digi mon better."
```

Research example: Oh & Yoon (in progress)

Findings:

- Words like "good," "great," and "better" exhibited varying degrees of sarcasm in reviews.
  - The word "good" had both positive and sarcastic connotations, often used in a past-tense context.
  - "Great" was generally positive but could be used sarcastically with specific expressions including negation.
  - "Better" frequently appeared in sarcastic and negatively oriented evaluations.

https://tenor.com/search/great-sarcasm-gifs
https://tenor.com/search/sarcastic-good-job-gifs
https://oladino.com/product/never-better-skeleton-funny-dead-inside-sarcastic-svg-cricut-file/



https://tenor.com/search/great-sarcasm-gifs
https://tenor.com/view/fantastic-thats-great-sarcastic-sarcasm-shrugging-it-off-gif-13761565

Q: How to detect sarcasm?

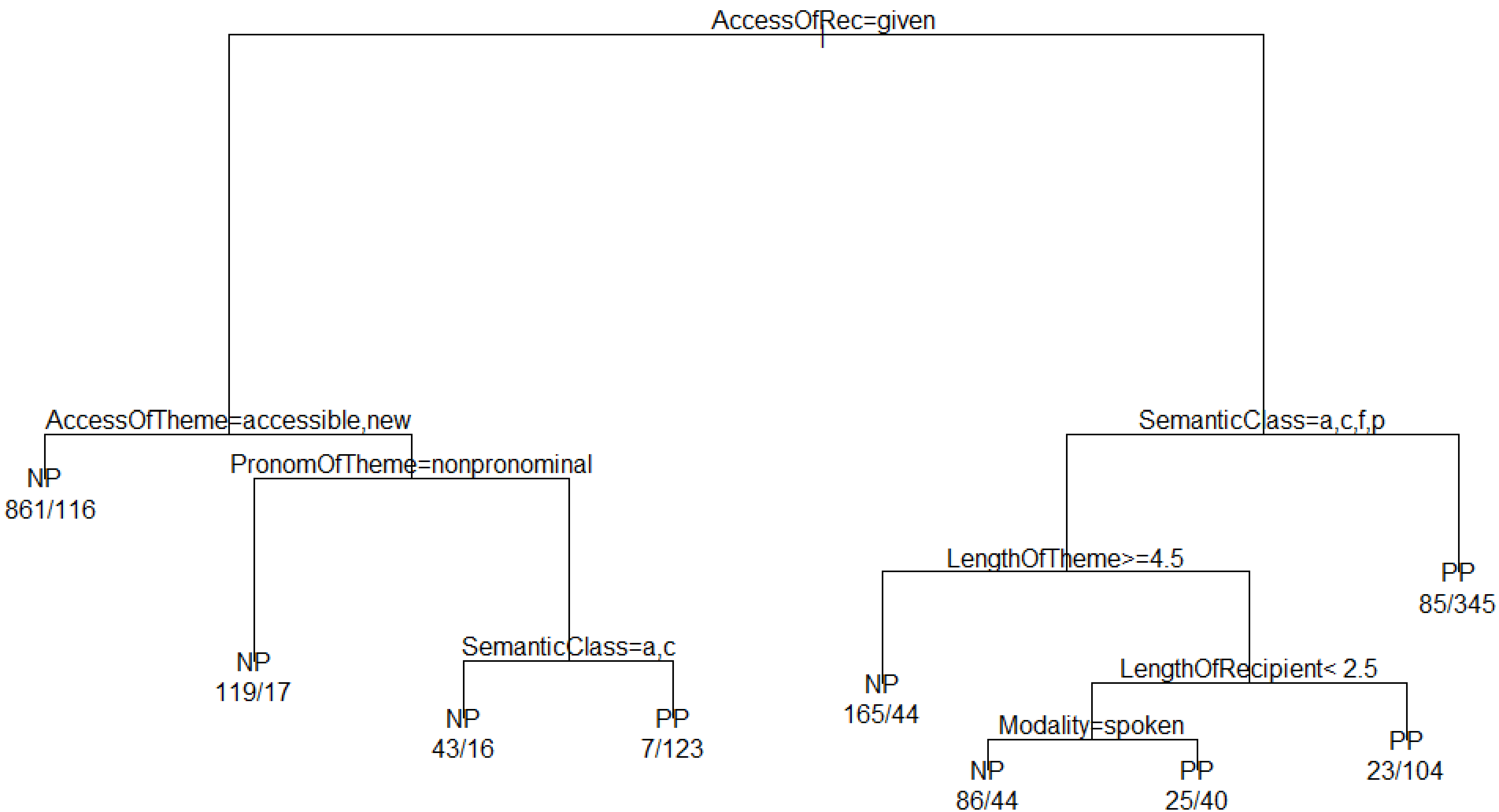
A1: Linguistic cues

A2: Contextual cues: Amazon rating

A3: Machine Learning approaches

Machine Learning

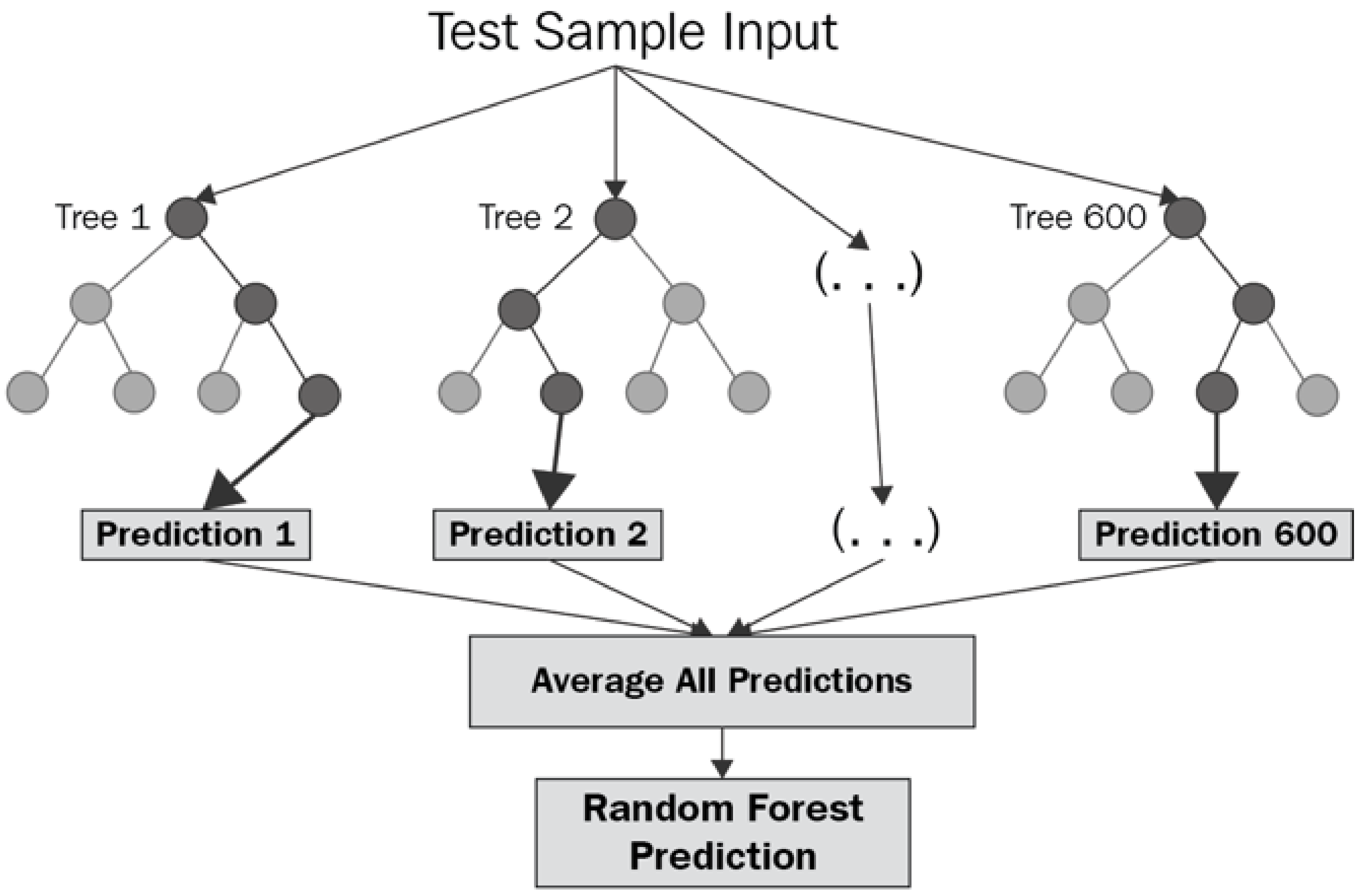
Decision trees for dative data



Machine Learning

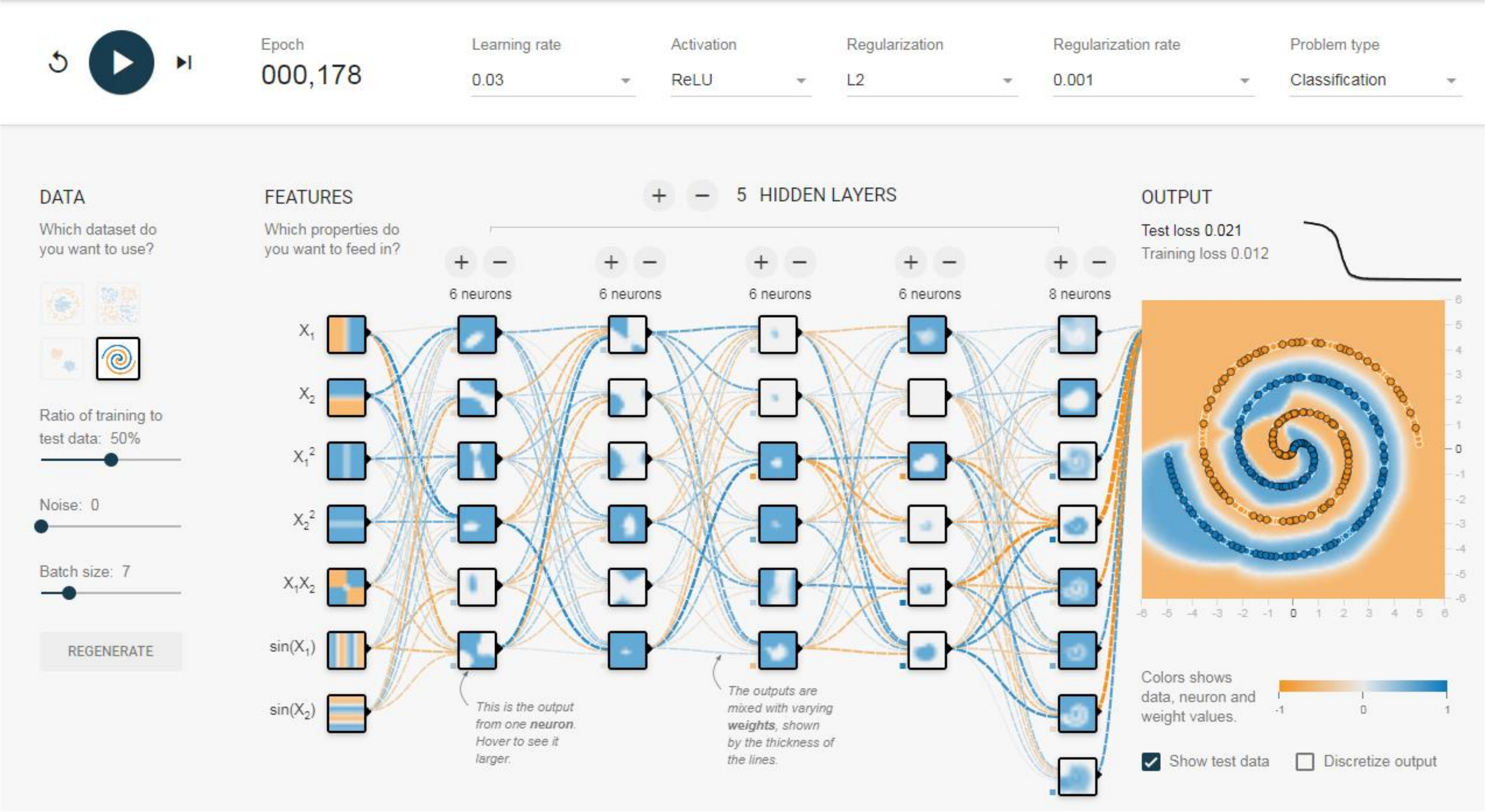
Random Forest

- Generate multiple (smaller) decision trees and keep the average
- Ensemble machine learning based on wisdom of crowds
- Avoid overfitting for better prediction performance



Machine Learning

Deep learning: Artificial Neural Network with multiple inner layers.
https://playground.tensorflow.org



Machine Learning approaches to Sarcasm detection

Table 3. Common feature-related approaches that were applied for sarcasm detecting in Twitter.

Algorithm	Author(s)	Sarcasm Features								
		Lexical	Stem	Pragmatic	Frequency	TF-IDF	POS	Ambiguity	Synonyms	Personality
SVM	González-Ibánez et al. (2011)	X		X	X		X			
	Tungthamthiti et al. (2014)	X		X						
	Bouazizi and Ohtsuki (2015)	X		X						
	Ghosh et al. (2015)	X		X	X		X			
	Barbieri et al. (2015)	X			X		X	X	X	
	Signhaniya et al. (2015)	X	X				X			
	Tungthamthiti et al. (2016)	X			X		X			
Logistic regression	Kovaz et al. (2013)	X			X		X			
	Jain and Hsu (2015)	X			X					
	Abercrombie and Hovy (2016)	X					X			
	Bali and Singh (2016)	X		X			X			
Naïve Bayes	Saha et al. (2017)	X	X				X			
	Das et al. (2018)	X			X					
	Parde and Nielsen (2018)	X		X	X					
Random Forest	Bouazizi and Ohtsuki (2018)	X		X			X			
	Bouazizi and Ohtsuki (2016)	X		X	X		X			
SASI	Davidov et al. (2010)	X			X					
Bootstrapping	Riloff et al. (2013)	X					X			
MaxEnt	Ptáček et al. (2014)	X		X	X		X			
SCUBA	Rajadesingan et al. (2015)	X					X			
LibSVM	Joshi et al. (2015)	X		X						
CUE-CNN	Amir et al. (2016)	X			X		X			
GRNN	Zhang et al. (2016)	X				X				
CNN-SVM	Poria et al. (2016)	X								X
CNN + LSTM + DNN	Ghosh and Veale (2016)	X		X	X					
Gradient Boost	Prasad et al. (2017)	X	X				X			
MIARN	Tay et al. (2018)	X								
MODEL-KEY	Ren et al. (2018)	X				X				
FBCA	Parmar et al. (2018)	X			X		X			

TF-IDF: Term Frequency-Inverse Document Frequency; SVM: Support Vector Machine; SASI: Semi-supervised Algorithm for Sarcasm Identification; MaxEnt: Maximum Entropy; SCUBA: Sarcasm Classification Using a Behavioral modeling Approach; CUE-CNN: Content and User Embedding Convolutional Neural Network; GRNN: Gated Recurrent Neural Network; CNN-SVM: Convolutional Neural Network–Support Vector Machine; LSTM: Long Short-Term Memory; DNN: Deep Neural Network; MIARN: Multi-dimensional Intra-Attention Recurrent Network; FBCA: Feature-based Composite Approach.

(Sarsam et al. 2020)

Machine Learning approaches to Sarcasm detection

Common features of sarcasm detection

1. Lexical features: interjections and punctuation
2. Stemmed features
3. Pragmatic features: positive/negative emoticons, ToUser
4. Frequency-related features
5. TF-IDF
6. Part-Of-Speech (POS) taggers
7. Ambiguity
8. Synonyms
9. Personality

(Sarsam et al. 2020)

Machine Learning approaches to Sarcasm detection

Common features of sarcasm detection

1. **Lexical features: interjections and punctuation**
2. Stemmed features
3. **Pragmatic features:** positive/negative emoticons, ToUser
4. Frequency-related features
5. **TF-IDF**
6. **Part-Of-Speech (POS) taggers**
7. **Ambiguity**
8. **Synonyms**
9. Personality

(Sarsam et al. 2020)

https://www.reddit.com/r/tomodachilife/comments/14omhgh/when_youve_literally_helped_your_mii_successfully/?rdt=52205

https://www.reddit.com/r/tomodachilife/comments/14omhgh/when_youve_literally_helped_your_mii_successfully/?rdt=52205

http://www.quickmeme.com/meme/35g4i4

https://www.pinterest.com/pin/341289096609996666/
https://www.reddit.com/r/tomodachilife/comments/14omhgh/when_youve_literally_helped_your_mii_successfully/?rdt=52205
https://cz.pinterest.com/pin/342836590385909405/?amp_client_id=CLIENT_ID%28_%29&mweb_unauth_id=&simplified=true
https://makeameme.org/meme/wow-thank-you-vxy0cd
https://www.teepublic.com/pin/17352828-sarcastic-thank-you


Thank You
For Listening!
ICANHASCHEEZBURGER.COM

https://makeameme.org/meme/any-questions-wed

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

Subset #1: 1^{st} -1734^{th} sentences
Subset #2: 1735^{th}-3646^{th} sentences

A: Two-sample Binomial tests

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275 ↑	0.134

ex) For negative sentences,

p_1 = (Prob. of neg. sentence in subset #1)
p_2 = (Prob. of neg. sentence in subset #2)
p = (Prob. of neg. sentence in the whole)
$\hat{p}_1, \hat{p}_2, \hat{p}$ are sample estimates for these

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

Subset #1: 1st -1734th sentences
Subset #2: 1735th-3646th sentences

A: Two-sample Binomial tests

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134 ↓

ex) For negative sentences,

p_1 = (Prob. of neg. sentence in subset #1)
p_2 = (Prob. of neg. sentence in subset #2)
p = (Prob. of neg. sentence in the whole)
$\hat{p}_1, \hat{p}_2, \hat{p}$ are sample estimates for these

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

Subset #1: 1^{st} -1734^{th} sentences
Subset #2: 1735^{th}-3646^{th} sentences

A: Two-sample Binomial tests

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085↑	-0.036↓

ex) For negative sentences,

p_1 = (Prob. of neg. sentence in subset #1)
p_2 = (Prob. of neg. sentence in subset #2)
p = (Prob. of neg. sentence in the whole)
$\hat{p}_1, \hat{p}_2, \hat{p}$ are sample estimates for these

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

Subset #1: 1st -1734th sentences
Subset #2: 1735th-3646th sentences

A: Two-sample Binomial tests

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085↑	-0.036↓

ex) For negative sentences,

p_1 = (Prob. of neg. sentence in subset #1)
p_2 = (Prob. of neg. sentence in subset #2)
p = (Prob. of neg. sentence in the whole)
$\hat{p}_1, \hat{p}_2, \hat{p}$ are sample estimates for these

Does this change happen just by chance?

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

Subset #1: 1st -1734th sentences
Subset #2: 1735th-3646th sentences

A: Two-sample Binomial tests (normal approximation, two-sides)

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085 ↑	-0.036 ↓
Z-statistic	6.085	-3.051
P-value	1.17×10^{-9}	0.0023

Reject the null since z-stat is too extreme

p_1 = (Prob. of neg. sentence in subset #1)
p_2 = (Prob. of neg. sentence in subset #2)
p = (Prob. of neg. sentence in the whole)
$\hat{p}_1, \hat{p}_2, \hat{p}$ are sample estimates for these

H_o: $p_2 - p_1 = 0$ (No change)
H_A: $p_2 - p_1 \neq 0$ (Real change)

Under H_o

$$\text{Z-statistic} = \frac{\hat{p}_2 - \hat{p}_1}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{N_1}+\frac{1}{N_2}\right)}} \sim N(0,1)$$

Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

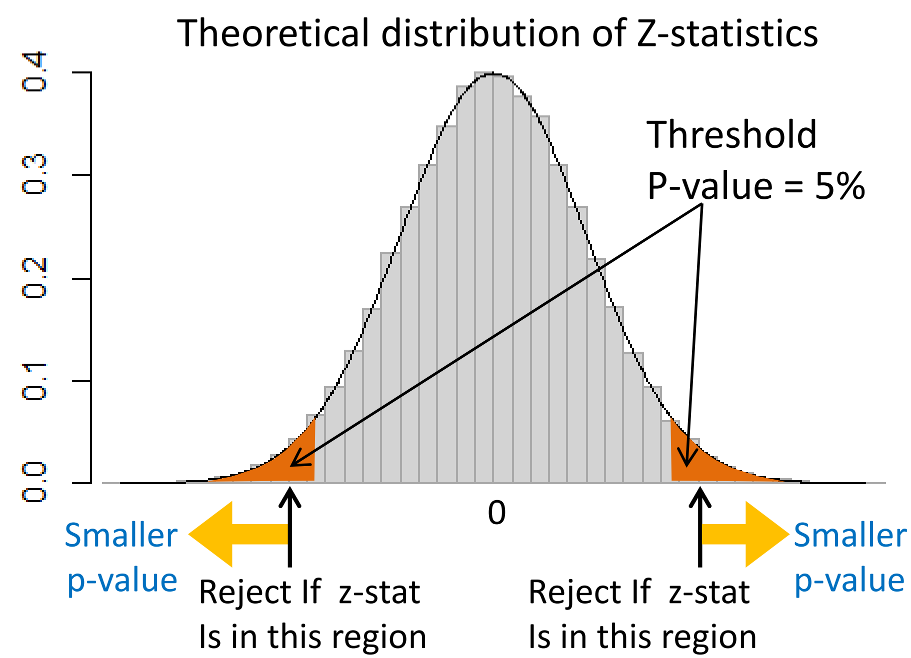
Subset #1: 1st -1734th sentences
Subset #2: 1735th-3646th sentences

Negative sentiment increases and positive sentiment decreases, significantly.

A: Two-sample Binomial tests (normal approximation, two-sides)

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085 ↑	-0.036 ↓
Z-statistic	6.085	-3.051
P-value	1.17×10^{-9}	0.0023

Reject the null since z-stat is too extreme



Example #1: Sentiment Analysis

Q: Changes in sentiment? How much? How significant?

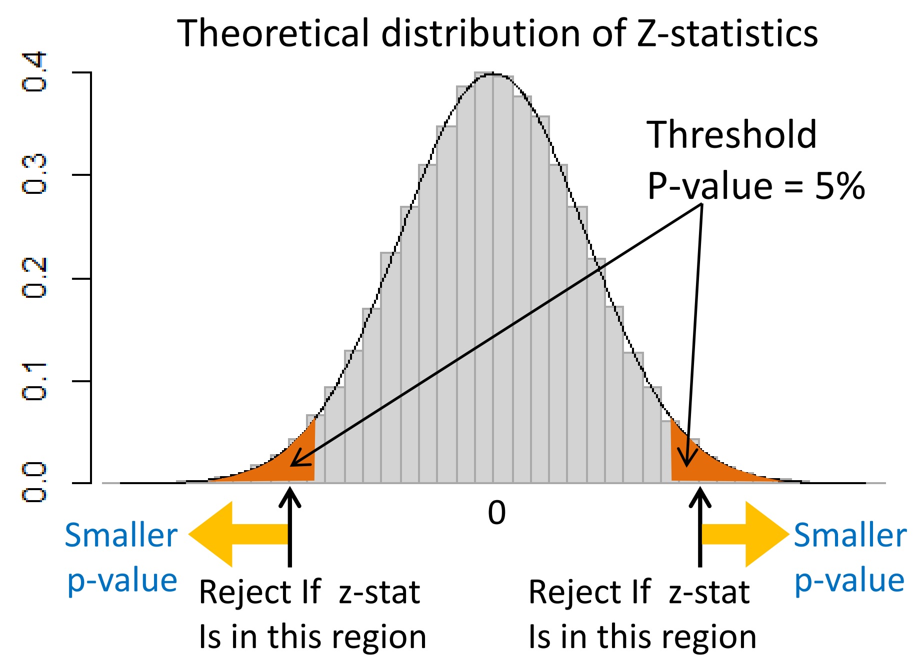
Subset #1: 1st -1734th sentences
Subset #2: 1735th-3646th sentences

Negative sentiment increases and positive sentiment decreases, significantly.

A: Two-sample Binomial tests (normal approximation, two-sides)

Statistic	Prob. of Negative	Prob. of Positive
$\hat{p}_1$	0.189	0.170
$\hat{p}_2$	0.275	0.134
$\hat{p}_2 - \hat{p}_1$	0.085 ↑	-0.036 ↓
Z-statistic	6.085	-3.051
P-value	1.17×10^{-9}	0.0023

Reject the null since z-stat is too extreme



Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg true.pos true.neg
   1:          1     0     0 FALSE FALSE    FALSE    FALSE
   2:          2     0     0 FALSE FALSE    FALSE    FALSE
   3:          3     0     0 FALSE FALSE    FALSE    FALSE
   4:          4     0     0 FALSE FALSE    FALSE    FALSE
   5:          5     0     0 FALSE FALSE    FALSE    FALSE
  ---
3642:       3642     1     0  TRUE FALSE     TRUE    FALSE
3643:       3643     0     1 FALSE  TRUE     TRUE    FALSE
3644:       3644     0     1 FALSE  TRUE    FALSE     TRUE
3645:       3645     0     0 FALSE FALSE    FALSE    FALSE
3646:       3646     0     1 FALSE  TRUE     TRUE    FALSE
```

True values

Assume these two columns are true sentiments of sentences manually labeled by human experts.

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg true.pos true.neg
   1:          1     0     0 FALSE FALSE    FALSE    FALSE
   2:          2     0     0 FALSE FALSE    FALSE    FALSE
   3:          3     0     0 FALSE FALSE    FALSE    FALSE
   4:          4     0     0 FALSE FALSE    FALSE    FALSE
   5:          5     0     0 FALSE FALSE    FALSE    FALSE
  ---
3642:       3642     1     0  TRUE FALSE     TRUE    FALSE
3643:       3643     0     1 FALSE  TRUE     TRUE    FALSE
3644:       3644     0     1 FALSE  TRUE    FALSE     TRUE
3645:       3645     0     0 FALSE FALSE    FALSE    FALSE
3646:       3646     0     1 FALSE  TRUE     TRUE    FALSE
```

Prediction error

Predicted values

True (reference) values

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Sensitivity = Recall = Power = True Positive Rate = 762/(762+50) = **0.9384**
= 1 - (Type-II error) = 1 – (False Negative Rate) = 1 – 0.0616

This "Negative" means "Non-negative sentiment", not "Negative sentiment"

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Uninformative classifier:
Sensitivity + Specificity = 1

Confusion matrix

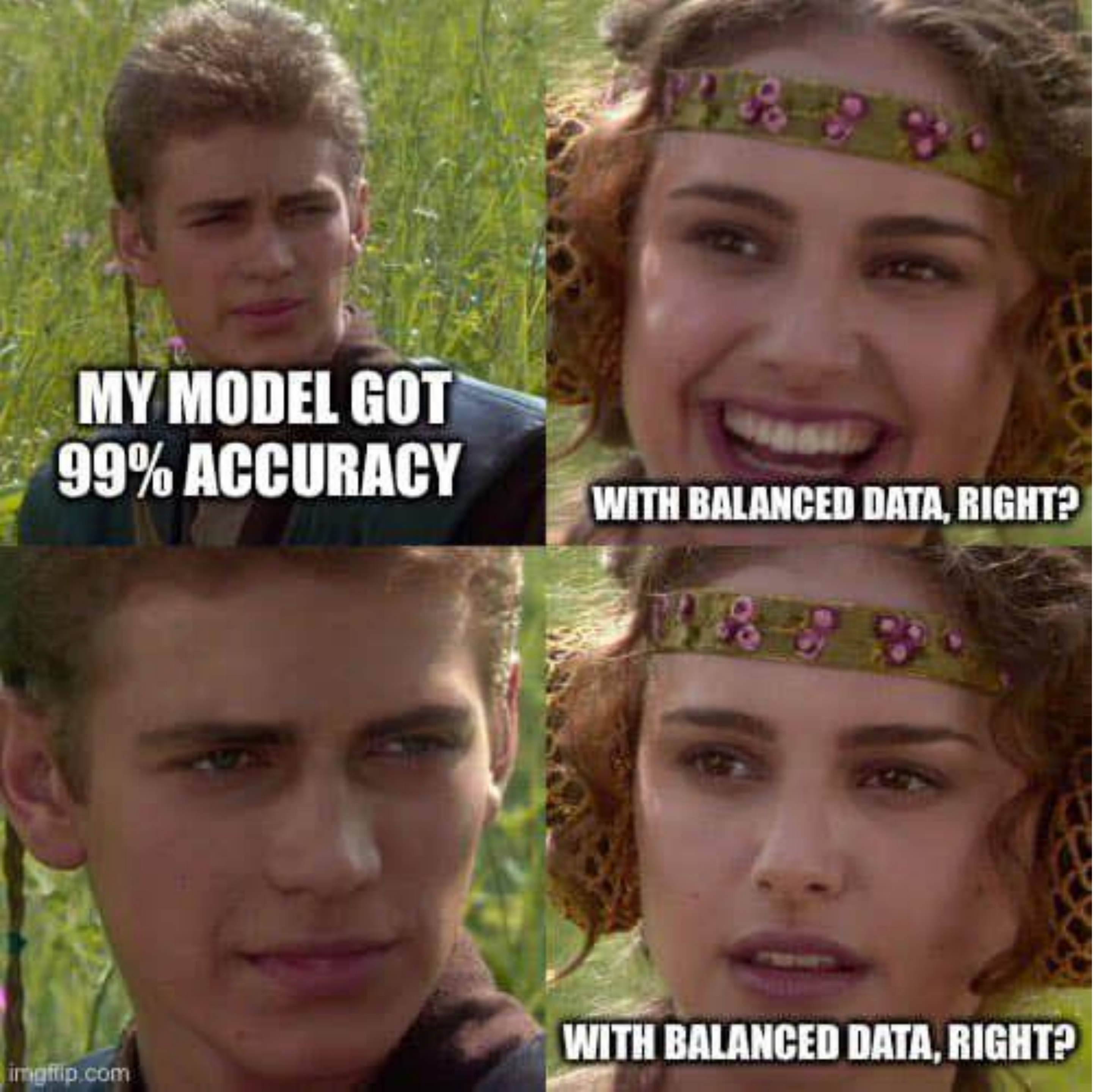
True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

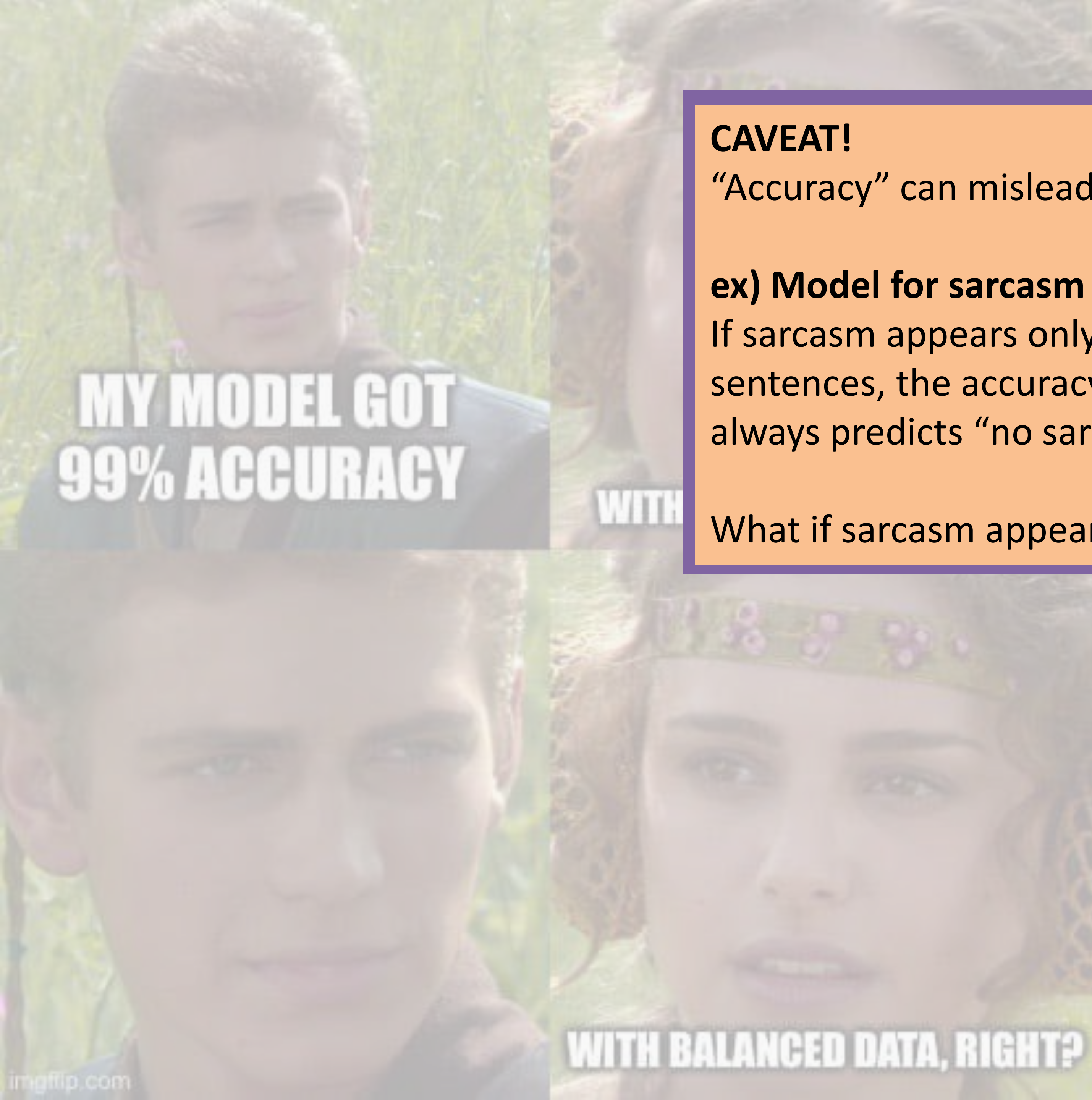
Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Sensitivity = Recall = Power = True Positive Rate = 762/(762+50) = **0.9384**
= 1 - (Type-II error) = 1 – (False Negative Rate) = 1 – 0.0616

Specificity = True Negative Rate = 2743/(2743+91) = **0.9679**
= 1 - (Type-I error) = 1 – (False $\underbrace{\text{Positive}}$ Rate) = 1 - 0.0321

This "Positive" means "Negative sentiment", not "Non-negative sentiment"


MY MODEL GOT 99% ACCURACY
WITH BALANCED DATA, RIGHT?
WITH BALANCED DATA, RIGHT?
imgflip.com



CAVEAT!
"Accuracy" can mislead model performance.

ex) Model for sarcasm detection
If sarcasm appears only once out of 100 sentences, the accuracy of a model that always predicts "no sarcasm" is also 99%.

What if sarcasm appears 0.5% of the time?

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg true.pos true.neg
   1:          1     0     0 FALSE FALSE    FALSE    FALSE
   2:          2     0     0 FALSE FALSE    FALSE    FALSE
   3:          3     0     0 FALSE FALSE    FALSE    FALSE
   4:          4     0     0 FALSE FALSE    FALSE    FALSE
   5:          5     0     0 FALSE FALSE    FALSE    FALSE
  ---
3642:       3642     1     0  TRUE FALSE     TRUE    FALSE
3643:       3643     0     1 FALSE  TRUE     TRUE    FALSE
3644:       3644     0     1 FALSE  TRUE    FALSE     TRUE
3645:       3645     0     0 FALSE FALSE    FALSE    FALSE
3646:       3646     0     1 FALSE  TRUE     TRUE    FALSE
```

True values

Assume these two columns are true sentiments of sentences manually labeled by human experts.

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

```
> rnj_sentiment_count
      sentenceID n.pos n.neg   pos   neg true.pos true.neg
   1:          1     0     0 FALSE FALSE    FALSE    FALSE
   2:          2     0     0 FALSE FALSE    FALSE    FALSE
   3:          3     0     0 FALSE FALSE    FALSE    FALSE
   4:          4     0     0 FALSE FALSE    FALSE    FALSE
   5:          5     0     0 FALSE FALSE    FALSE    FALSE
  ---
3642:       3642     1     0  TRUE FALSE     TRUE    FALSE
3643:       3643     0     1 FALSE  TRUE     TRUE    FALSE
3644:       3644     0     1 FALSE  TRUE    FALSE     TRUE
3645:       3645     0     0 FALSE FALSE    FALSE    FALSE
3646:       3646     0     1 FALSE  TRUE     TRUE    FALSE
```

Prediction error

Predicted values

True (reference) values

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Confusion matrix

True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Sensitivity = Recall = Power = True Positive Rate = 762/(762+50) = **0.9384**
= 1 - (Type-II error) = 1 – (False Negative Rate) = 1 – 0.0616

This "Negative" means "Non-negative sentiment", not "Negative sentiment"

Example #1: Sentiment Analysis

Q: Is this sentiment classification model good?

Uninformative classifier:
Sensitivity + Specificity = 1

Confusion matrix

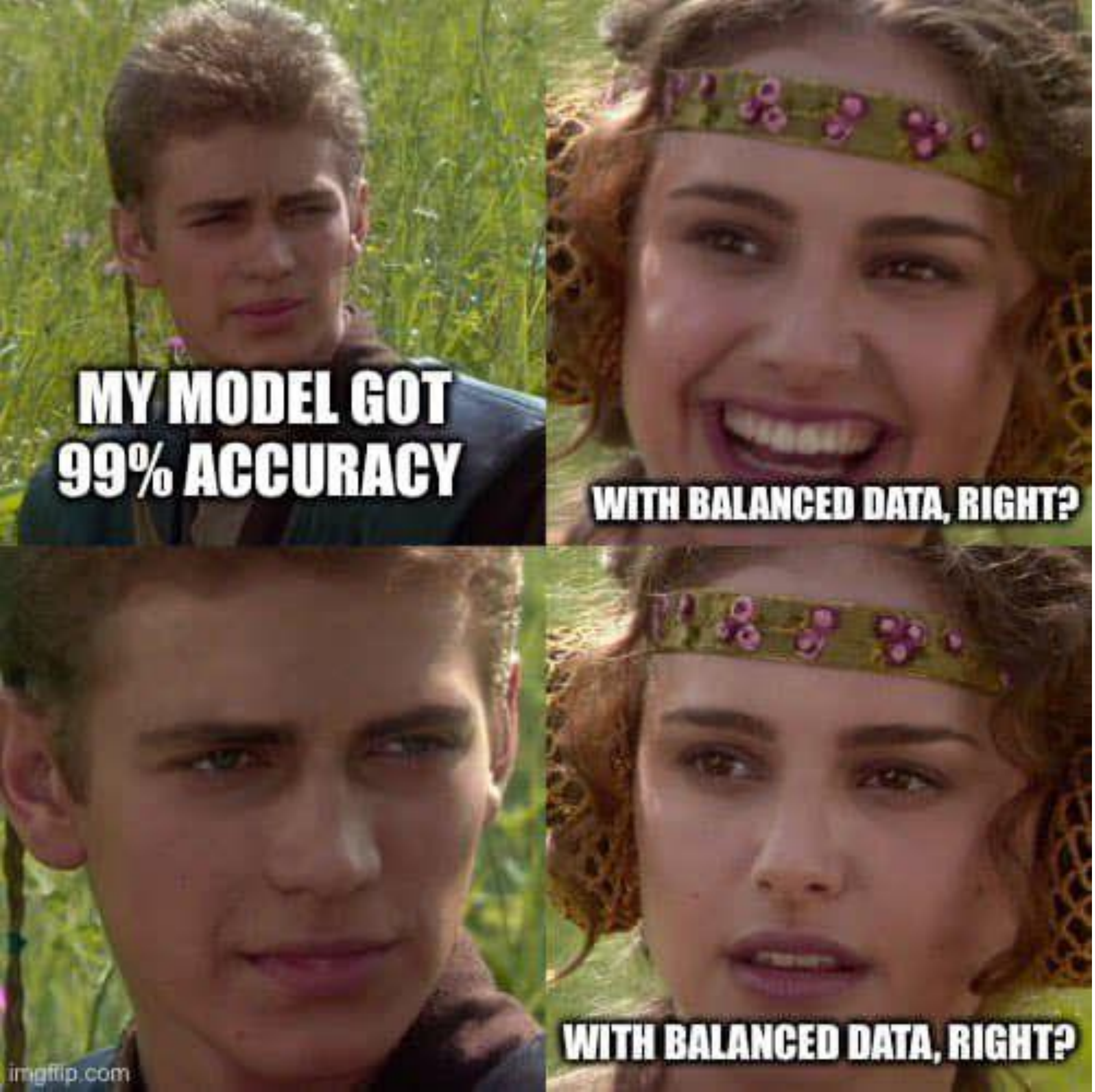
True / **Predicted**	Negative sentiment (**event**), 23%	Non-negative sentiment (**no event**), 77%
Negative	762	91
non-Negative	50	2743

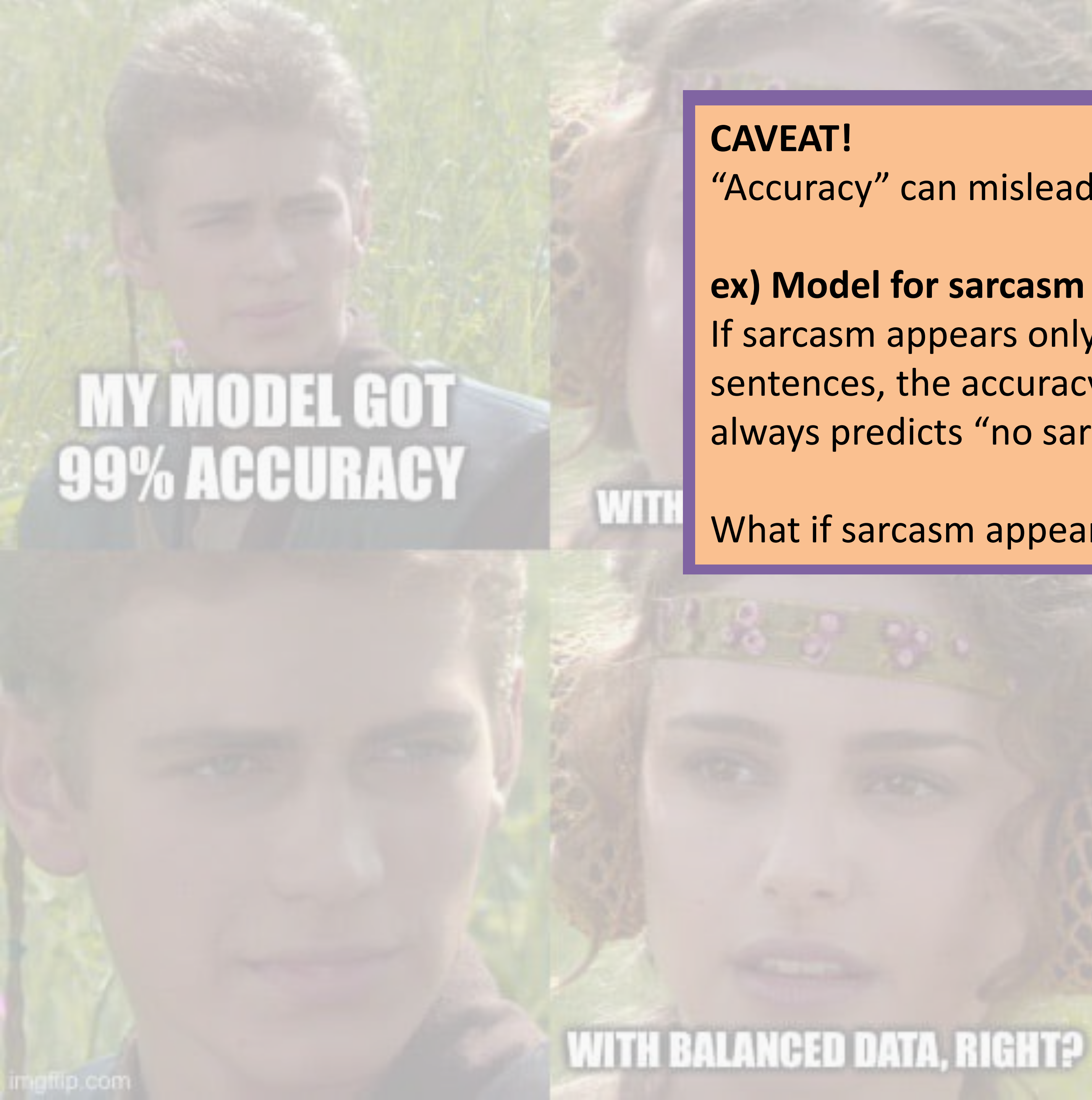
Accuracy = (2743+762)/(2743+762+50+91) = **0.9613**

Sensitivity = Recall = Power = True Positive Rate = 762/(762+50) = **0.9384**
= 1 - (Type-II error) = 1 – (False Negative Rate) = 1 – 0.0616

Specificity = True Negative Rate = 2743/(2743+91) = **0.9679**
= 1 - (Type-I error) = 1 – (False Positive Rate) = 1 - 0.0321

This "Positive" means "Negative sentiment", not "Non-negative sentiment"


MY MODEL GOT 99% ACCURACY
WITH BALANCED DATA, RIGHT?
WITH BALANCED DATA, RIGHT?
imgflip.com



CAVEAT!
"Accuracy" can mislead model performance.

ex) Model for sarcasm detection
If sarcasm appears only once out of 100 sentences, the accuracy of a model that always predicts "no sarcasm" is also 99%.

What if sarcasm appears 0.5% of the time?