IITB-Monash Research Academy

An Indian-Australian research partnership

Indian Institute of Technology Bombay

MONASH University

Investigations in Computational Sarcasm

Aditya Joshi
Postdoctoral Research Fellow, Data61, CSIRO

This work was done from 2014-2017 during my PhD.
PhD Supervisors: Pushpak Bhattacharyya, IITB
Mark J Carman, Monash

03.12.2018

Outline

Outline

Introduction

Sarcasm,
Motivation,
Incongruity,
Work Overview

Sarcasm Detection

Sarcasm Detection Using Incongruity Within Target Text

Sarcasm Detection Using Incongruity Outside Target Text

Beyond detection

Understanding the phenomenon,
Sarcasm generation

Conclusion

Summary, pointers to future work

Outline

Introduction

Sarcasm, Motivation, Incongruity, Work Overview

Sarcasm Detection

Sarcasm Detection Using Incongruity Within Target Text

Sarcasm Detection Using Incongruity Outside Target Text

Beyond detection

Understanding the phenomenon, Sarcasm generation

Conclusion

Summary, pointers to future work

Sarcasm & Computational Sarcasm

Sarcasm is defined as ‘the use of irony to mock or convey contempt’

(Source: Oxford Dictionary)

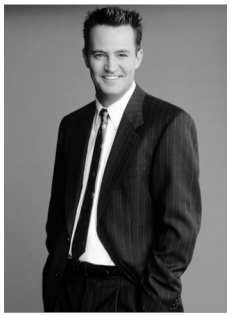
I love this perfume so much that I suggest you wear it with your windows shut. *(Pang et al, 2008)*

Etymology: Greek *‘sarkasmós’* (‘to tear flesh with teeth’)

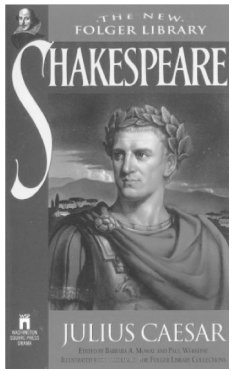
Computational sarcasm is defined as the set of computational approaches to process sarcasm (in text)

Sarcasm in Popular Culture

TV



Theater

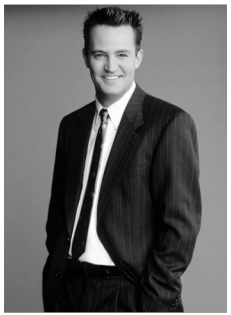


Science Fiction

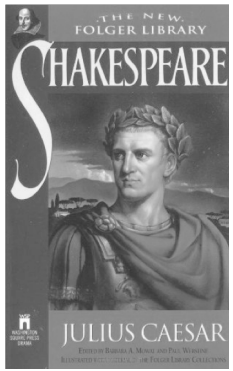
Images from wikimedia commons and other internet sources

Sarcasm in Popular Culture

TV



Theater



Science Fiction

Why
computational sarcasm?

Images from wikimedia commons and other internet sources

Motivation for Computational Sarcasm

For improved sentiment analysis

	Precision (Sarc)	Precision (Non-sarc)
Conversation Transcripts		
MeaningCloud[1]	20.14	49.41
NLTK (Bird, 2006)	38.86	81
Tweets		
MeaningCloud[1]	17.58	50.13
NLTK (Bird, 2006)	35.17	69

Two sentiment analysis systems perform poorly for sarcastic text as compared to non-sarcastic text.

For a successful Turing test

Human: You are fast like a snail

ALICE (Wallace, 2009): Thank you for telling me I am fast like a snail.

Assistant.ai: A good assistant is whatever their boss needs them to be.

For 'human-like' interactions, chatbots must understand sarcasm and respond appropriately.

1 www.meaningcloud.com

A linguistically grounded approach

Incongruity: ‘Incompatibility’

A situation where components of a text are incompatible either with each other or with some background knowledge.

> Verbal irony uses **incongruity** to suggest a distinction between reality and expectation Gibbs (1994)
>
> Sarcasm/irony is understood because of **incongruity** Ivanko and Pexman (2003)

Incongruity provides a useful framework to devise solutions for different problems in computational sarcasm.

Computational Sarcasm through the lens of incongruity

Incongruity provides a useful framework to understand and fit different problems of computational sarcasm

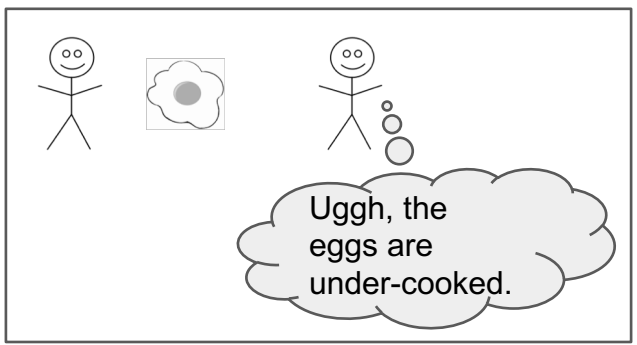
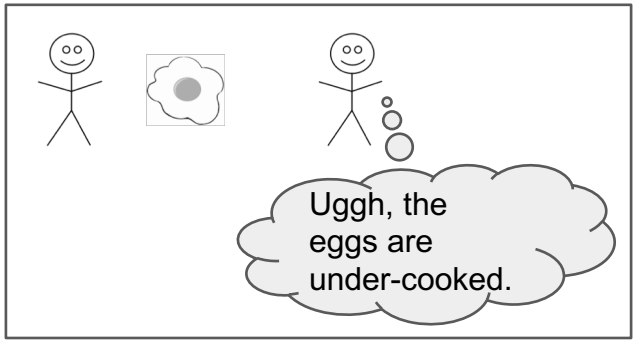


Image of egg from wikimedia commons.

Computational Sarcasm through the lens of incongruity

Incongruity provides a useful framework to understand and fit different problems of computational sarcasm



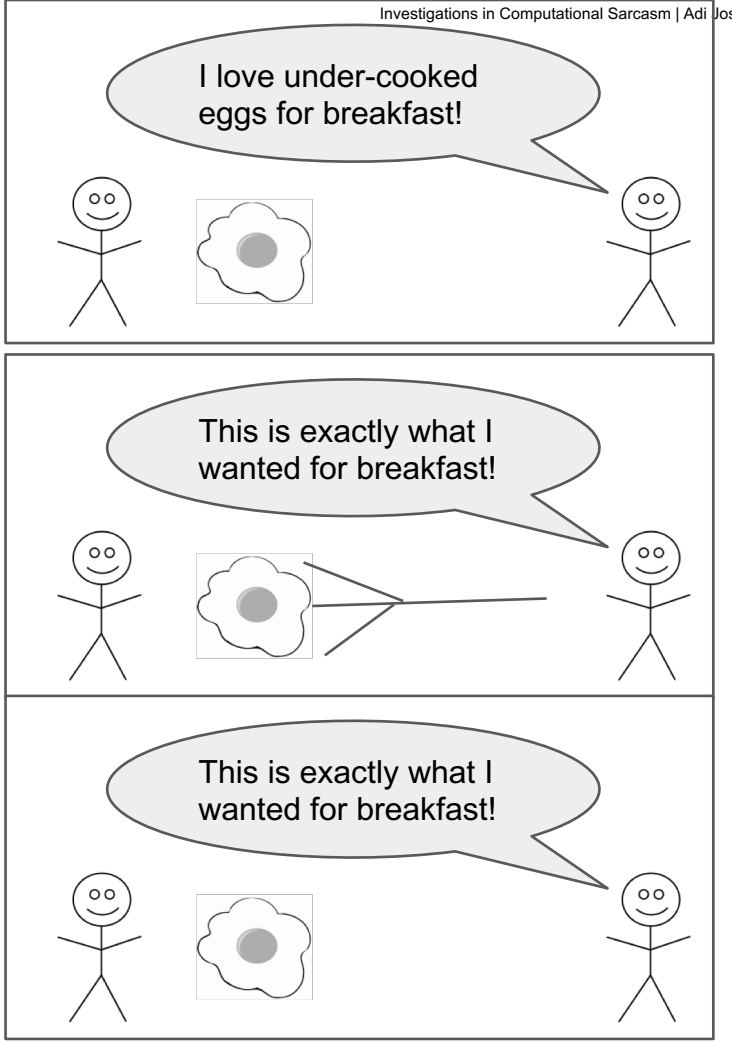
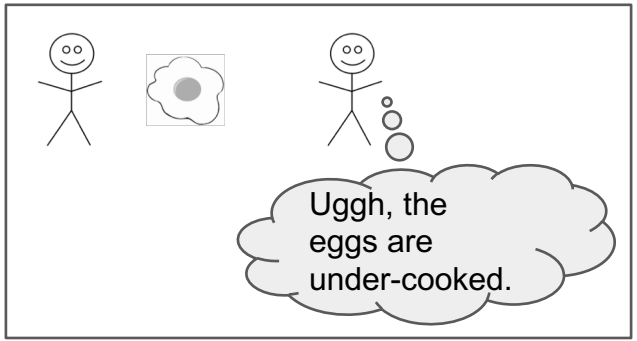


Image of egg from wikimedia commons.

Computational Sarcasm through the lens of incongruity

Incongruity provides a useful framework to understand and fit different problems of computational sarcasm



Increasing difficulty

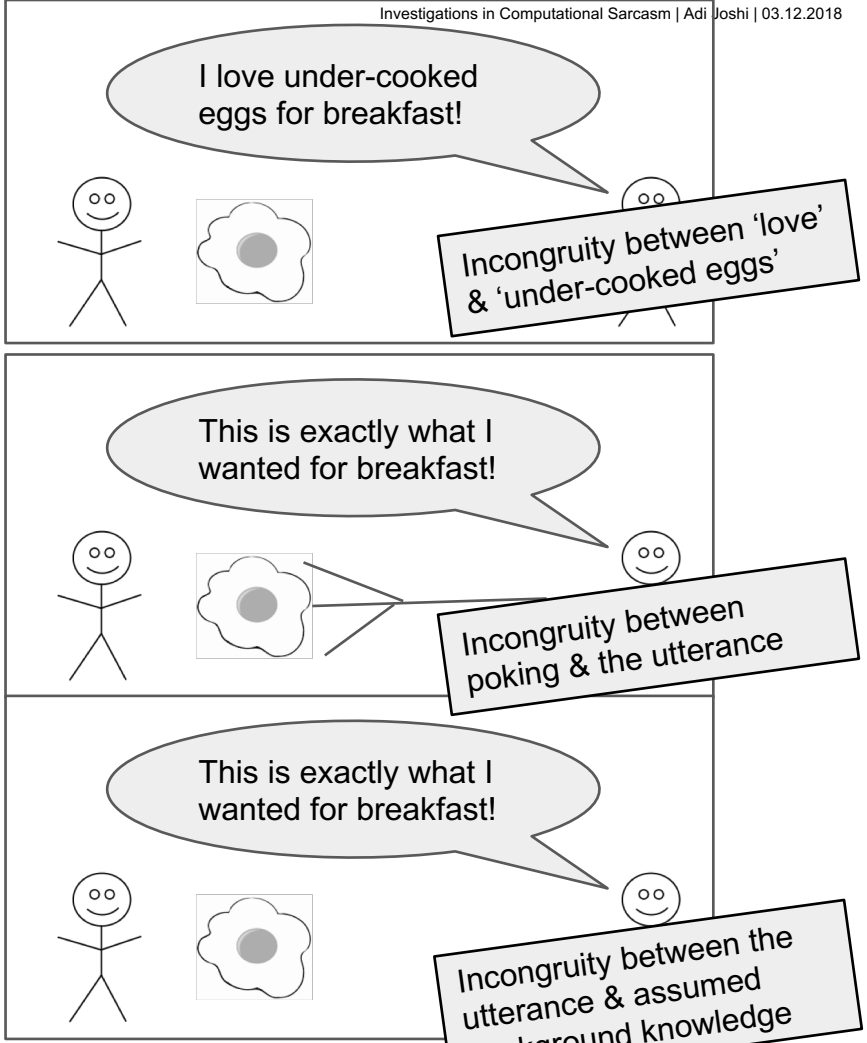
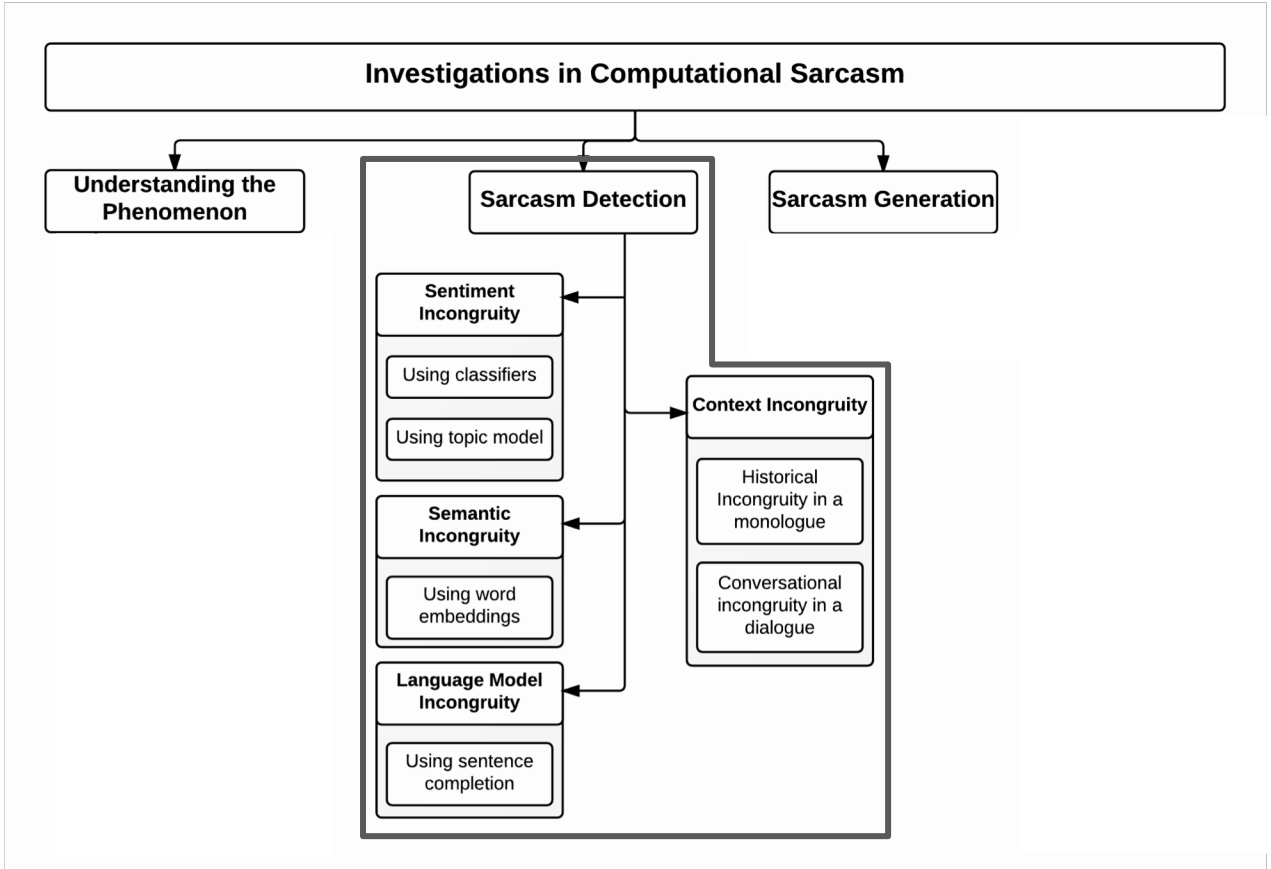


Image of egg from wikimedia commons.

Work Overview



Outline

Introduction

Sarcasm,
Motivation,
Incongruity,
Work Overview

Sarcasm Detection

Sarcasm Detection Using Incongruity Within Target Text

Sarcasm Detection Using Incongruity Outside Target Text

Beyond detection

Understanding the phenomenon,
Sarcasm generation

Conclusion

Summary, pointers to future work

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- Language model incongruity
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

Outline: Approaches for Sarcasm Detection

- **Sentiment incongruity as features***
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- Language model incongruity
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

* Aditya Joshi, Vinita Sharma, Pushpak Bhattacharyya, Harnessing context incongruity for sarcasm detection, ACL-IJCNLP 2015.

Sentiment incongruity using features: Motivation

Sentiment incongruity is incongruity expressed through the use of sentiment words:

- **Explicit Incongruity**: Words of both polarity are present
  - *Being stranded in traffic is the best way to start a week!*

- **Implicit Incongruity:** Words of one polarity are present, with a phrase of implied polarity
  - *I love this paper so much that I made a doggy bag out of it.*

Hypothesis: Augmenting features capturing sentiment incongruity can be useful for sarcasm detection

Sentiment Incongruity Features

Lexical	
Unigrams	Unigrams in the training corpus
Pragmatic	
Capitalization	Numeric feature indicating presence of capital letters
Emoticons & laughter expressions	Numeric feature indicating presence of emoticons and 'lol's
Punctuation marks	Numeric feature indicating presence of punctuation marks
Implicit Incongruity *	
Implicit Sentiment Phrases	Boolean feature indicating phrases extracted from the implicit phrase extraction step
Explicit Incongruity +	
#Explicit incongruity	Number of times a word is followed by a word of opposite polarity
Largest positive /negative subsequence	Length of largest series of words with polarity unchanged
#Positive words	Number of positive words
#Negative words	Number of negative words
Lexical Polarity	Polarity of a tweet based on words present

* Based on a Bootstrapping algorithm by Riloff et al (2013)
+ Based on features by Ramteke et al (2013)

Experiment Setup

Three datasets:

Tweet-A (5208 total, 4172 sarcastic)

Tweet-B (2278 total, 506 sarcastic) Riloff et al. (2013)

Discussion-A (1502 total, 752 sarcastic) Walker et al. (2012)

LibSVM[1], five-fold cross-validation

1 https://www.csie.ntu.edu.tw/ cjlin/libsvm/

Results

Features	P	R	F
Original Algorithm by Riloff et al. (2013)			
Ordered	0.774	0.098	0.173
Unordered	0.799	0.337	0.474
Our system			
Lexical (**Baseline**)	0.820	0.867	0.842
Lexical+Implicit	0.822	0.887	0.853
Lexical+Explicit	0.807	0.985	0.8871
All features	0.814	0.976	**0.8876**

Tweet-A

Features	P	R	F
Lexical (**Baseline**)	0.645	0.508	0.568
Lexical+Explicit	0.698	0.391	0.488
Lexical+Implicit	0.513	0.762	0.581
All features	0.489	0.924	**0.640**

Discussion-A

Approach	P	R	F
Riloff et al. (2013) (**best reported**)	0.62	0.44	0.51
Maynard and Greenwood (2014)	0.46	0.38	0.41
Our system (all features)	**0.77**	**0.51**	**0.61**

Tweet-B

Our sentiment incongruity features outperform two past works and also for two text forms (tweets and discussion forum posts)

Error Analysis

Subjective polarity: *'Yay for 3 hour Chem labs'*

No incongruity within text: About 10% misclassified examples that we analyzed, contained no sentiment incongruity within the text.

Incongruity due to numbers: *'Going in to work for 2 hours was totally worth the 35 minute drive.'*

Dataset granularity: *'How special, now all you have to do is prove that a glob of cells has rights. I happen to believe that a person's life and the right to life begins at conception'.*

Politeness: *'Post all your inside jokes on facebook, I really want to hear about them'.*

Error Analysis

Subjective polarity: *'Yay for 3 hour Chem labs'*

Incongruity detection using author's historical context

No incongruity within text: About 10% misclassified examples that we analyzed, contained no sentiment incongruity within the text.

Semantic Incongruity detection

Incongruity due to numbers: *'Going in to work for 2 hours was totally worth the 35 minute drive.'*

Dataset granularity: *'How special, now all you have to do is prove that a glob of cells has rights. I happen to believe that a person's life and the right to life begins at conception'.*

Politeness: *'Post all your inside jokes on facebook, I really want to hear about them'.*

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- **Semantic incongruity as word embedding-based features***
- Sentiment incongruity as topic model
- Language model incongruity
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

* Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya and Mark J Carman, 'Are Word Embedding-based Features Useful for Sarcasm Detection?', EMNLP 2016.

Semantic Incongruity: Motivation

Semantic incongruity may occur due to semantic discordance, without the use of sentiment words.

Hypothesis: Incongruity can be captured using word embedding-based features, **in addition to other features**

“A woman needs a man like a fish needs a bicycle.”

Word2Vec similarity(man,woman) = 0.766
Word2Vec similarity(fish, bicycle) = 0.131

Features for semantic incongruity

Unweighted similarity features (S):

Over every word and then word pair, compute:

1) Maximum score of most similar word pair
2) Minimum score of most similar word pair
3) Maximum score of most dissimilar word pair
4) Minimum score of most dissimilar word pair

	man	woman	fish	needs	bicycle
man	-	**0.766**	**0.151**	**0.078**	**0.229**
woman	0.766	-	0.084	0.060	0.229
fish	0.151	0.084	-	0.022	0.130
needs	0.078	0.060	0.022	-	0.060
bicycle	0.229	0.229	0.130	0.060	-

Distance-weighted similarity features (WS): 4 S features weighted by linear distance between the two words

Both (S+WS): 8 features

Experiment Setup

- Dataset: 3629 Book snippets (759 sarcastic) downloaded from GoodReads website, labeled by users with tags. We download the ones with ‘sarcasm’ as sarcastic, ones with ‘philosophy’ as non-sarcastic
- Five-fold cross-validation
- Classifier: SVM-Perf (Joachims, 2006a) optimised for F-score
- Configurations:
  - Four prior works (augmented with our sets of features)
  - Four kinds of pre-trained word embeddings (Word2Vec[1], LSA[2], GloVe[3], Dependency weights-based[4])

1 https://code.google.com/archive/p/word2vec/
2 http://www.lingexp.uni-tuebingen.de/z2/LSAspaces/
3 http://nlp.stanford.edu/projects/glove/
4 https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/

Results (1/2)

The semantic incongruity features alone do not perform well.

Our semantic incongruity features result in improved performance when augmented with features from four past works for four kinds of word embeddings, with two exceptions.

The past works include our sentiment incongruity features (J).

	LSA			GloVe			Dependency Weights			Word2Vec		
	P	R	F	P	R	F	P	R	F	P	R	F
L	73	79	75.8	73	79	75.8	73	79	75.8	73	79	75.8
+S	81.8	78.2	**79.95**	81.8	79.2	**80.47**	81.8	78.8	80.27	80.4	80	**80.2**
+WS	76.2	79.8	77.9	76.2	79.6	77.86	81.4	80.8	81.09	80.8	78.6	79.68
+S+WS	77.6	79.8	78.68	74	79.4	76.60	82	80.4	**81.19**	81.6	78.2	79.86
G	84.8	73.8	78.91	84.8	73.8	78.91	84.8	73.8	**78.91**	84.8	73.8	**78.91**
+S	84.2	74.4	**79**	84	72.6	77.8	84.4	72	77.7	84	72.8	78
+WS	84.4	73.6	78.63	84	75.2	**79.35**	84.4	72.6	78.05	83.8	70.2	76.4
+S+WS	84.2	73.6	78.54	84	74	78.68	84.2	72.2	77.73	84	72.8	78
B	81.6	72.2	76.61	81.6	72.2	76.61	81.6	72.2	76.61	81.6	72.2	76.61
+S	78.2	75.6	**76.87**	80.4	76.2	**78.24**	81.2	74.6	**77.76**	81.4	72.6	76.74
+WS	75.8	77.2	76.49	76.6	77	76.79	76.2	76.4	76.29	81.6	73.4	77.28
+S+WS	74.8	77.4	76.07	76.2	78.2	77.18	75.6	78.8	77.16	81	75.4	**78.09**
J	85.2	74.4	79.43	85.2	74.4	79.43	85.2	74.4	79.43	85.2	74.4	79.43
+S	84.8	73.8	78.91	85.6	74.8	79.83	85.4	74.4	79.52	85.4	74.6	**79.63**
+WS	85.6	75.2	**80.06**	85.4	72.6	78.48	85.4	73.4	78.94	85.6	73.4	79.03
+S+WS	84.8	73.6	78.8	85.8	75.4	**80.26**	85.6	74.4	**79.6**	85.2	73.2	78.74

Performance obtained on augmenting word embedding features to features from four prior works, for four word embeddings; L: Liebrecht et al. (2013), G: González-Ibánez et al. (2011a), B: Buschmeier et al. (2014) , J: Joshi et al. (2015)

Results (2/2)

Word Embedding	Average F-score Gain
LSA	0.452
Glove	0.651
Dependency	1.048
Word2Vec	1.143

Average gain in F-scores for the four types of word embeddings; These values are computed for a subset of these embeddings consisting of words common to all four

Word2Vec and Dependency weight-based word embeddings give the highest F-score gain.

Error Analysis

Embedding issues due to incorrect senses: *'Great. Relationship advice from one of America's most wanted.'*

Contextual sarcasm: *'Oh, and I suppose the apple ate the cheese'.*

Metaphors in non-sarcastic text: *'Oh my love, I like to vanish in you like a ripple vanishes in an ocean - slowly, silently and endlessly'.*

Error Analysis

Embedding issues due to incorrect senses: *'Great. Relationship advice from one of America's most wanted.'*

Contextual sarcasm: *'Oh, and I suppose the apple ate the cheese'.*

Incongruity detection using conversational context

Metaphors in non-sarcastic text: *'Oh my love, I like to vanish in you like a ripple vanishes in an ocean - slowly, silently and endlessly'.*

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- **Sentiment incongruity as topic model***
- Language model incongruity
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

* Aditya Joshi, Prayas Jain, Pushpak Bhattacharyya, Mark J Carman, 'Who would have thought of that!': A Novel Hierarchical Topic Model for Extraction of Sarcasm-prevalent Topics and Sarcasm Detection', ExPROM-COLING 2016.

Sentiment incongruity using topic model: Motivation

Sarcastic tweets are likely to have a mixture of words of both sentiments as against tweets with literal sentiment (either positive or negative)

Hypothesis: Our topic model discovers sarcasm-prevalent topics, in order to aid the task of sarcasm detection

1. A document-level topic variable that models sarcasm prevalence
2. A word-level sentiment variable that models sentiment mixture

The first topic model for sarcasm detection, to the best of our knowledge.

Input/Output

Input:

Hashtag-based supervised dataset of tweets

Three labels: Literal positive, literal negative and sarcastic

Word-sentiment distribution

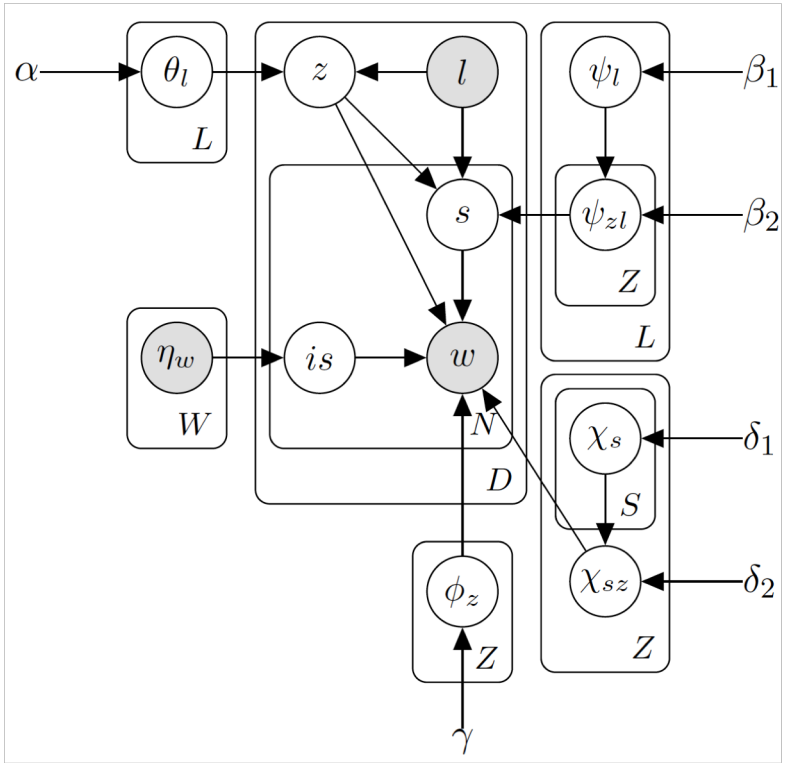
Output:

Sarcasm-prevalent topics

Sentiment-label distributions

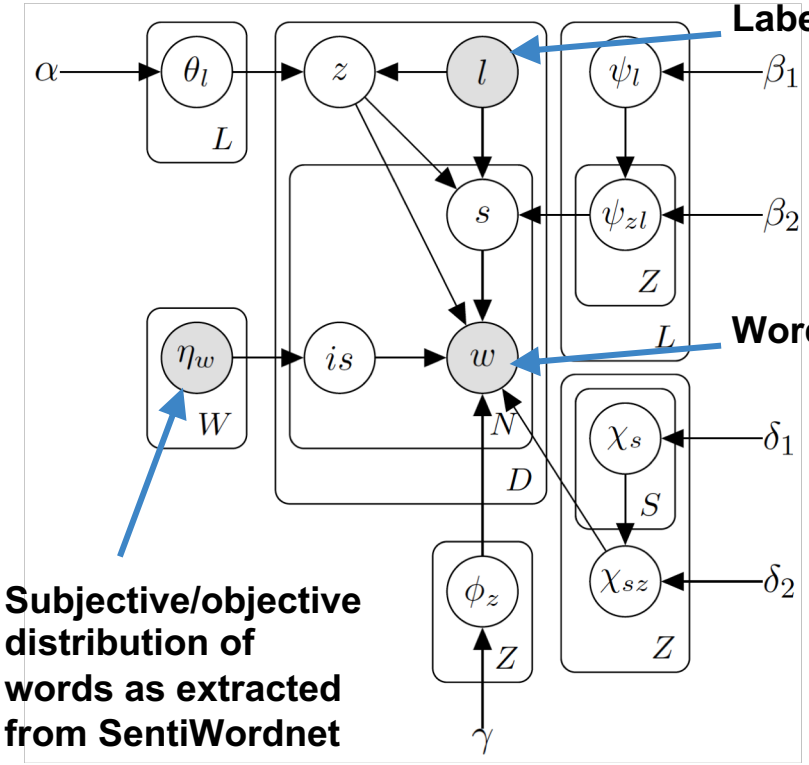
Sentiment clusters corresponding to topics

Plate Diagram



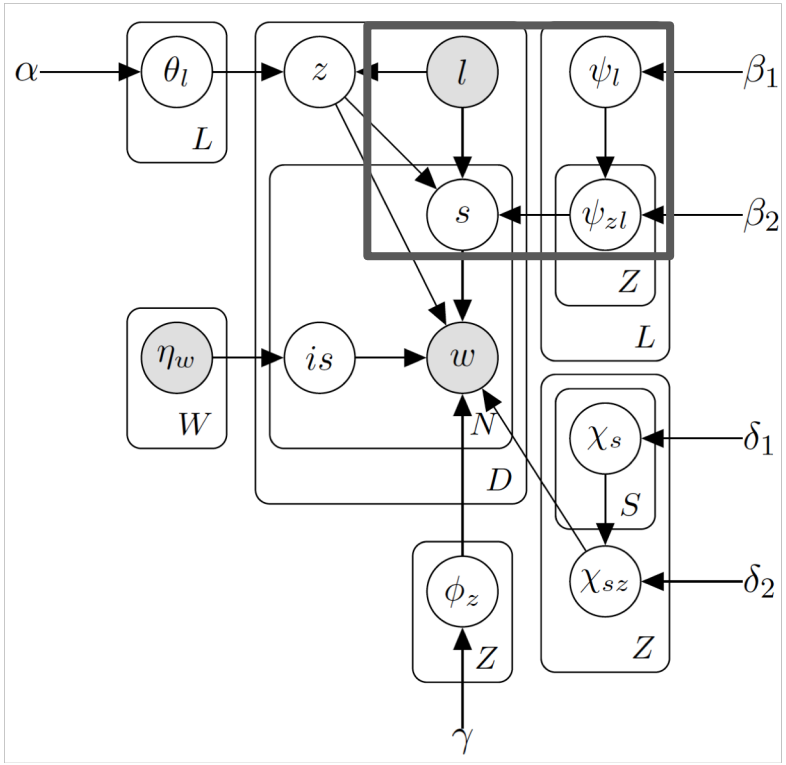
Observed Variables and Distributions	
w	Word in a tweet
l	Label of a tweet; takes values: positive, negative, sarcastic)
Distributions	
η_w	Distribution over switch values given a word w
Latent Variables and Distributions	
z	Topic of a tweet
s	Sentiment of a word in a tweet; takes values: positive, negative
is	Switch variable indicating whether a word is a topic word or a sentiment word; takes values: 0, 1
Distributions	
θ_l	Distribution over topics given a label l, with prior α
ϕ_z	Distribution over words given a topic z and switch =0 (topic word), with prior γ
χ_s	Distribution over words given sentiment s and switch=1 (sentiment word), with prior δ_1
χ_{sz}	Distribution over words given a sentiment s and topic z and switch=1 (sentiment word), with prior δ_2
ψ_l	Distribution over sentiment given a label l and switch =1 (sentiment word), with prior β_1
ψ_{zl}	Distribution over sentiment given a label l and topic z and switch =1 (sentiment word), with prior β_2

Plate Diagram



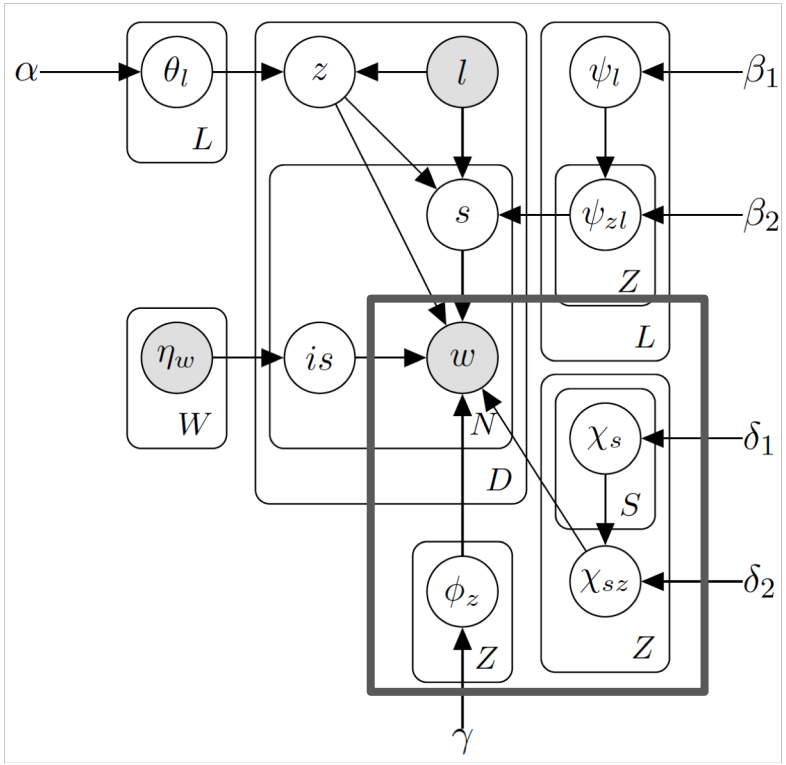
Observed Variables and Distributions	
w	Word in a tweet
l	Label of a tweet; takes values: positive, negative, sarcastic)
Distributions	
η_w	Distribution over switch values given a word w
Latent Variables and Distributions	
z	Topic of a tweet
s	Sentiment of a word in a tweet; takes values: positive, negative
is	Switch variable indicating whether a word is a topic word or a sentiment word; takes values: 0, 1
Distributions	
θ_l	Distribution over topics given a label l, with prior α
ϕ_z	Distribution over words given a topic z and switch =0 (topic word), with prior γ
χ_s	Distribution over words given sentiment s and switch=1 (sentiment word), with prior δ_1
χ_{sz}	Distribution over words given a sentiment s and topic z and switch=1 (sentiment word), with prior δ_2
ψ_l	Distribution over sentiment given a label l and switch =1 (sentiment word), with prior β_1
ψ_{zl}	Distribution over sentiment given a label l and topic z and switch =1 (sentiment word), with prior β_2

Plate Diagram



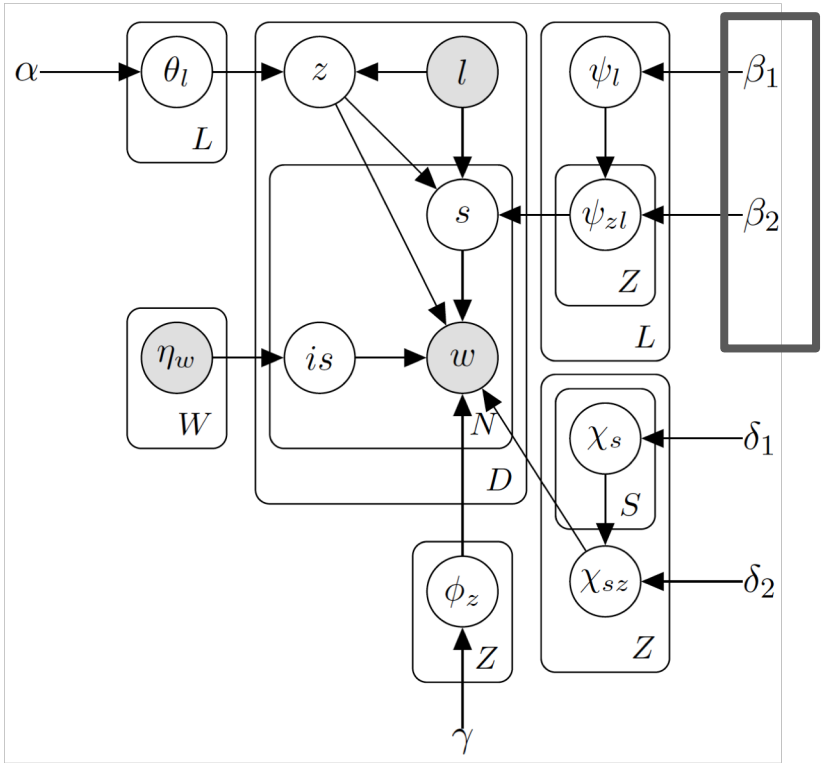
Observed Variables and Distributions	
w	Word in a tweet
l	Label of a tweet; takes values: positive, negative, sarcastic)
Distributions	
η_w	Distribution over switch values given a word w
Latent Variables and Distributions	
z	Topic of a tweet
s	Sentiment of a word in a tweet; takes values: positive, negative
is	Switch variable indicating whether a word is a topic word or a sentiment word; takes values: 0, 1
Distributions	
θ_l	Distribution over topics given a label l, with prior α
ϕ_z	Distribution over words given a topic z and switch =0 (topic word), with prior γ
χ_s	Distribution over words given sentiment s and switch=1 (sentiment word), with prior δ_1
χ_{sz}	Distribution over words given a sentiment s and topic z and switch=1 (sentiment word), with prior δ_2
ψ_l	Distribution over sentiment given a label l and switch =1 (sentiment word), with prior β_1
ψ_{zl}	Distribution over sentiment given a label l and topic z and switch =1 (sentiment word), with prior β_2

Plate Diagram



Observed Variables and Distributions	
w	Word in a tweet
l	Label of a tweet; takes values: positive, negative, sarcastic)
Distributions	
η_w	Distribution over switch values given a word w
Latent Variables and Distributions	
z	Topic of a tweet
s	Sentiment of a word in a tweet; takes values: positive, negative
is	Switch variable indicating whether a word is a topic word or a sentiment word; takes values: 0, 1
Distributions	
θ_l	Distribution over topics given a label l, with prior α
ϕ_z	Distribution over words given a topic z and switch =0 (topic word), with prior γ
χ_s	Distribution over words given sentiment s and switch=1 (sentiment word), with prior δ_1
χ_{sz}	Distribution over words given a sentiment s and topic z and switch=1 (sentiment word), with prior δ_2
ψ_l	Distribution over sentiment given a label l and switch =1 (sentiment word), with prior β_1
ψ_{zl}	Distribution over sentiment given a label l and topic z and switch =1 (sentiment word), with prior β_2

Plate Diagram



Observed Variables and Distributions	
w	Word in a tweet
l	Label of a tweet; takes values: positive, negative, sarcastic)
Distributions	
η_w	Distribution over switch values given a word w
Latent Variables and Distributions	
z	Topic of a tweet
s	Sentiment of a word in a tweet; takes values: positive, negative
is	Switch variable indicating whether a word is a topic word or a sentiment word; takes values: 0, 1
Distributions	
θ_l	Distribution over topics given a label l, with prior α
ϕ_z	Distribution over words given a topic z and switch =0 (topic word), with prior γ
χ_s	Distribution over words given sentiment s and switch=1 (sentiment word), with prior δ_1
χ_{sz}	Distribution over words given a sentiment s and topic z and switch=1 (sentiment word), with prior δ_2
ψ_l	Distribution over sentiment given a label l and switch =1 (sentiment word), with prior β_1
ψ_{zl}	Distribution over sentiment given a label l and topic z and switch =1 (sentiment word), with prior β_2

Experiment Setup

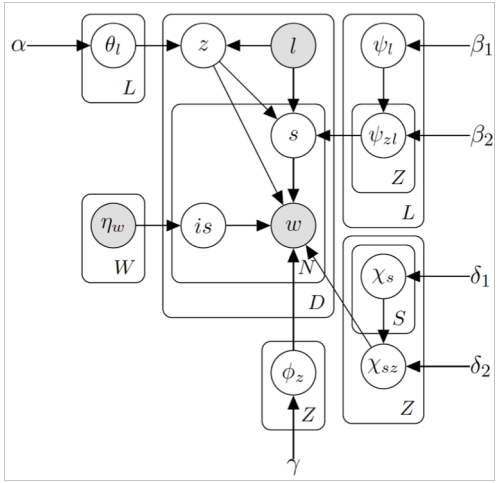
166,955 tweets, out of which nearly 75000 are sarcastic. Created using hashtag-based supervision

L=3

S=2

Z=50

Block-based Gibbs sampling



Results (1/4)

Music	work/school	Orlando Incident	Holiday	Quotes	Food
pop	work	orlando	summer	quote(s)	food
country	sleep	shooting	wekend	morning	lunch
rock	night	prayers	holiday	inspiration	vegan
bluegrass	morning	families	friends	motivation	breakfast
beatles	school	victims	sun,beach	mind	cake

Stock(s)/ Commodities	Father	Gun	Pets	Health
silver	father(s)	gun(s)	dog	fitness
gold	dad	orlando	cat	gym
index	daddy	trump	baby	run
price	family	shooting	puppy	morning
consumer	work	muslim	pets	health

Top topic words for a set of topics

Topics P(l/z)	Positive	Negative	Sarcastic
Holiday	**0.9538**	0.0140	0.0317
Father	**0.9224**	0.0188	0.0584
Quote	**0.8782**	0.0363	0.0852
Food	**0.8100**	0.0331	0.1566
Music	**0.7895**	0.0743	0.1363
Fitness	**0.7622**	0.0431	0.1948
Orlando Incident	0.0130	**0.9500**	0.0379
Gun	0.1688	0.3074	**0.5230**
Work	0.1089	0.0354	**0.8554**
Humor	0.0753	0.1397	**0.7841**

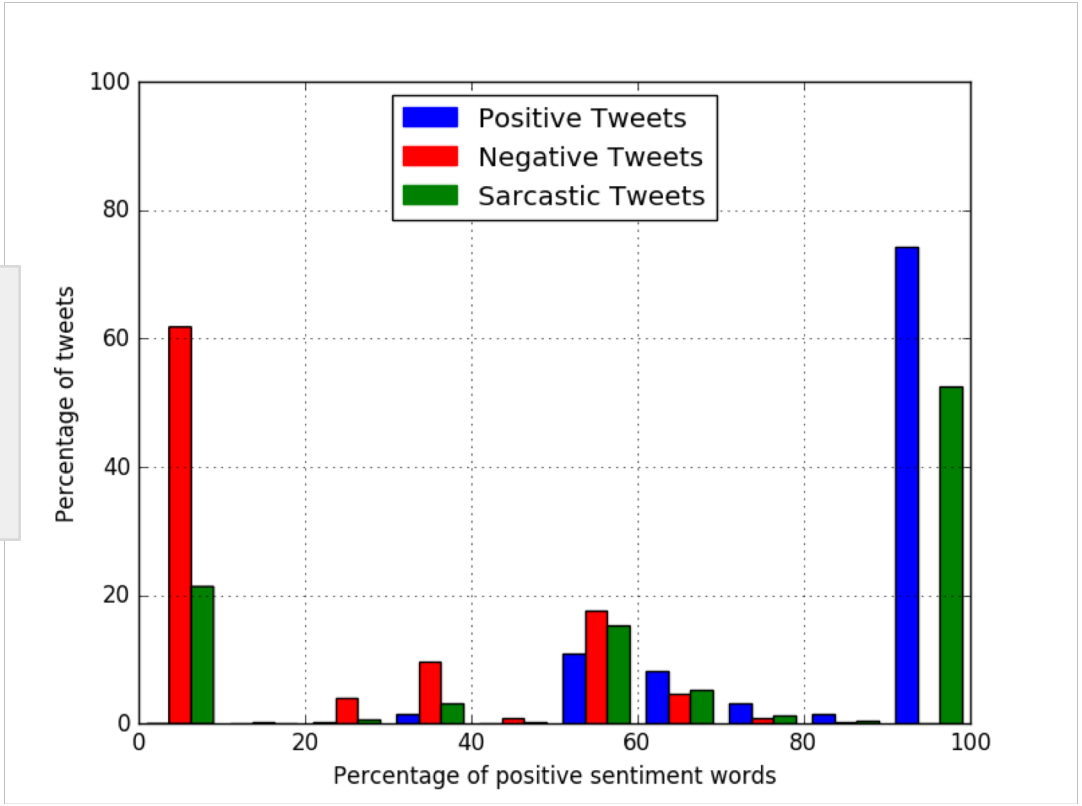
Label distribution of topics

Compared to LDA and supervised LDA, the extracted topics are qualitatively better.

Results (2/4)

Nearly 50% sarcastic tweets have 100% positive words

Nearly 20% sarcastic tweets have 0% positive words



Distribution of tweets for different sentiment mixtures

Results (3/4)

Application to Sarcasm Detection

1. Log-likelihood-based, for each label
2. Sampling-based

Compared with two prior works

Test set: 35398 total, 26210 positive, 5535 negative, **3653 sarcastic**

Results (4/4)

Approach	P (%)	R (%)	F (%)
Buschmeier et al. [2014]	10.41	100.00	18.85
Liebrecht et al. [2013]	11.03	99.88	19.86
LDA	100.0	23.55	38.12
Supervised LDA as features	100.0	10.26	18.61
Sarcasm Topic Model: Log Likelihood	55.70	32.87	**41.34**
Sarcasm Topic Model: Sampling	58.58	13.11	21.42

The sarcasm topic model performs better than four baselines for sarcasm detection.

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- **Language model incongruity***
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

* Aditya Joshi, Samarth Agrawal, Pushpak Bhattacharyya, Mark J Carman, 'Expect the unexpected: Harnessing Sentence Completion for Sarcasm Detection', PACLING 2017.

Language Model Incongruity: Motivation

Incongruity in sarcastic sentences goes against the expected language model as given by context2vec

I love being _____

Top words predicted by sentence completion

Rank	Word
0	star-struck
1	honest
5	overprotective
8	super-fit
12	open-minded
1102	ignored

Our Approach

Input: Sentence
Parameter: Threshold
For every content word *cw* at position i:

Get the most likely word *lw* for position i, given rest of the sentence

Calculate similarity between *cw* and *lw*

If minimum similarity over all content words < threshold:

Return sarcastic

Else:

Return non-sarcastic

Our Approach: Example

Input: *'I love being ignored'*
Parameter: 0.4
For every content word *cw* at position i: *{love, ignored}*

Get the most likely word *lw* for position i, given rest of the sentence

Calculate similarity between *cw* and *lw*

If minimum similarity over all content words < threshold:

Return sarcastic

Else:

Return non-sarcastic

Our Approach: Example

Input: *'I love being ignored'*
Parameter: 0.4
For every content word *cw* at position i: *{love, ignored}*

Get the most likely word *lw* for position i, given rest of the sentence

Calculate similarity between *cw* and *lw*

I [] being ignored. → Expected word: hate
I love being [] → Expected word: happy

If minimum similarity over all content words < threshold:

Return sarcastic

similarity(love, hate) = 0
similarity(ignored, happy) = 0.0204

Else:

Return non-sarcastic

Our Approach: Example

Input: *'I love being ignored'*
Parameter: 0.4
For every content word *cw* at position i: *{love, ignored}*

Get the most likely word *lw* for position i, given rest of the sentence

Calculate similarity between *cw* and *lw*

I [] being ignored. → Expected word: hate
I love being [] → Expected word: happy

If minimum similarity over all content words < threshold:

Return **sarcastic**

similarity(love, hate) = 0
similarity(ignored, happy) = 0.0204

Else:

Return non-sarcastic

Experiment Setup

Datasets: Tweets Riloff et al (2013), Discussion forum posts Walker et al (2014)

Threshold: Experimentally determined over a range of values; two-fold cross-validation

Similarity metrics: Word2Vec similarity, Wordnet similarity

Approach:

(1) Approach 1: Iterate over all words
(2) Approach 2: Iterate over top 50% most incongruous words (based on pairwise word2vec similarity)

Results

		P	R	F
Riloff et al. (2013)		62	44	51
Joshi et al. (2015)		77	51	61
Similarity Metric	**Best-T**	**P**	**R**	**F**
All words				
Word2Vec	(0.1, 0.1)	67.68	47.96	56.12
WordNet	(0.1, 0.1)	68.83	76.93	72.66
Incongruous words-only				
Word2Vec	(0.42,0.1)	63.92	77.64	70.09
WordNet	(0.14,0.12)	82.81	77.91	**80.28**

Tweets

		P	R	F
Joshi et al. (2015)		48.9	92.4	**64**
Similarity Metric	**Best-T**	**P**	**R**	**F**
All words				
Word2Vec	(0.48, 0.48)	56.20	52.17	54.10
WordNet	(0.37, 0.46)	43.13	48.04	45.45
Incongruous words-only				
Word2Vec	(0.19,0.25)	36.48	47.41	41.23
WordNet	(0.15,0.12)	28.34	48.04	35.33

Discussion forum posts

*Our language model incongruity algorithm outperforms past work for tweets but **not** for discussion forum posts.*

The candidate list of incongruous words appears to be crucial.

Discussion

Would sarcasm detection improve if we knew the exact out-of-place word?

We use dataset by Ghosh et al (2015) to validate this.

Approach	T	P	R	F
All-words	0.29	55.07	55.78	55.43
Oracle	0.014	59.13	68.37	**63.42**

Comparison with oracle case

If the exact incongruous word is known, there is a substantial improvement in the F-score. This 'oracle' is but a hypothetical case.

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- Language model incongruity
- **Historical context incongruity in a rule-based system***
- Conversational context incongruity using sequence labeling

* Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, Mark J Carman, 'Your sentiment precedes you: Using an author's historical tweets to predict sarcasm', WASSA at EMNLP 2015

Historical incongruity: Motivation

Some forms of sarcasm can be detected using incongruity with the historical context of the author

Today, X says: “P is the best politician in the history of this country!”

Case 1: Ten days ago, X said: “P took a good decision by supporting that policy.”

Case 2: Ten days ago, X said: “P took a bad decision by supporting that policy.”

Our Approach

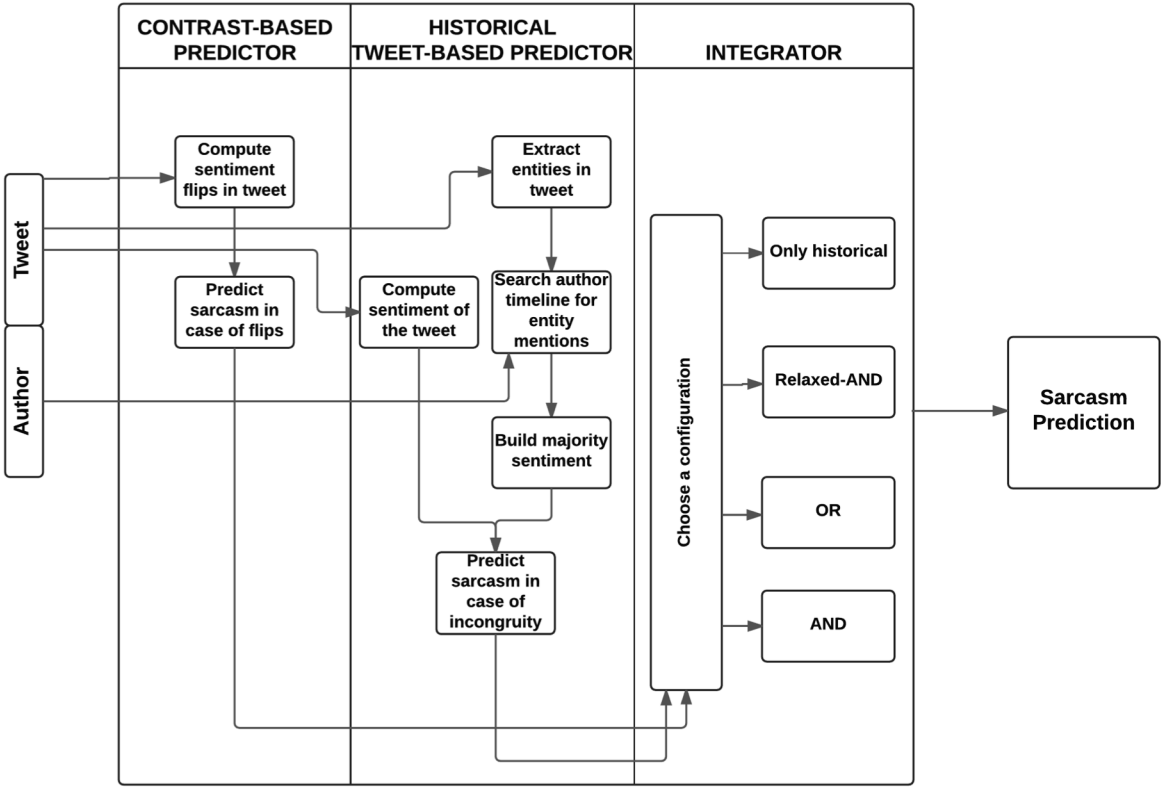
A rule-based system that combines simple sentiment incongruity with historical sentiment incongruity

Input: (Tweet, Twitter User/Author)
Output: Sarcastic/Non-sarcastic

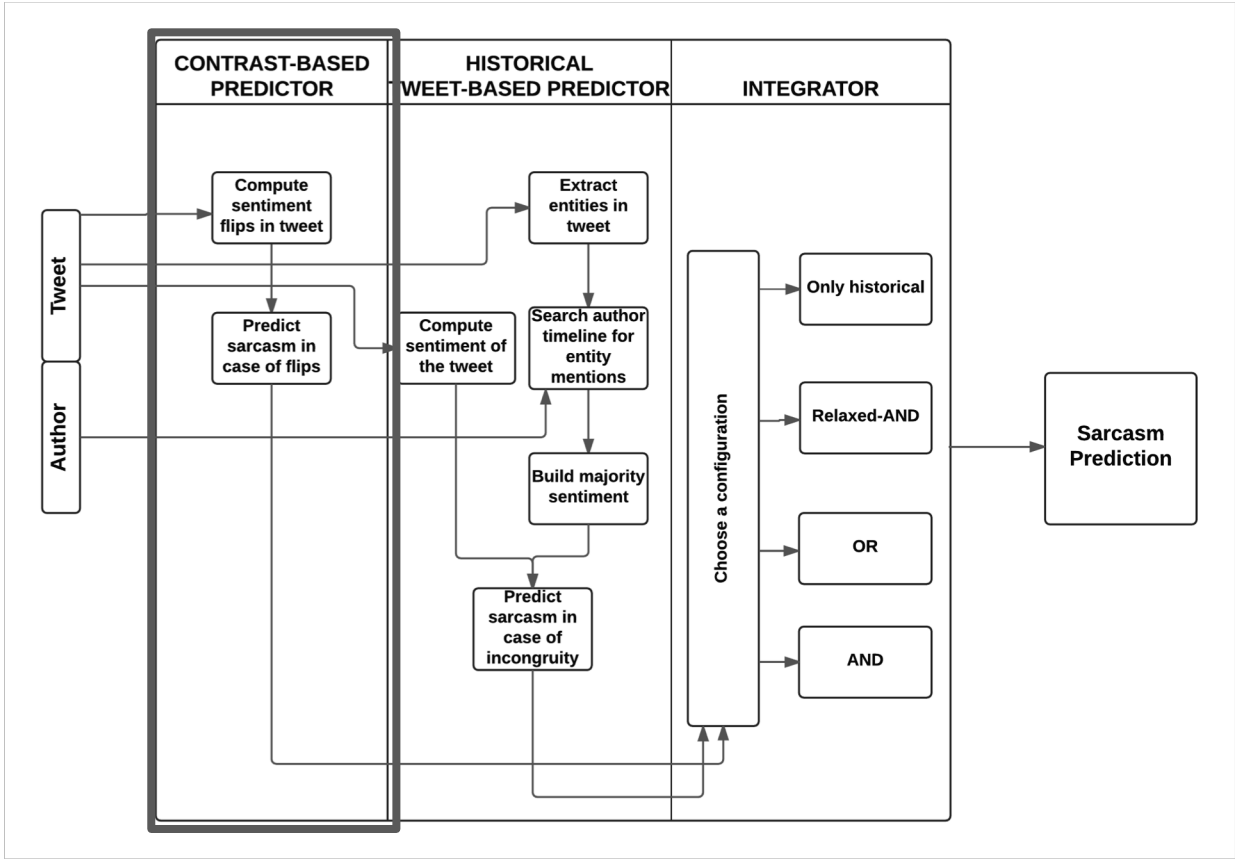
Assumption: The author has past tweets in order to capture her/his historical sentiment

To the best of our knowledge, the first reported approach for sarcasm detection using historical context incongruity

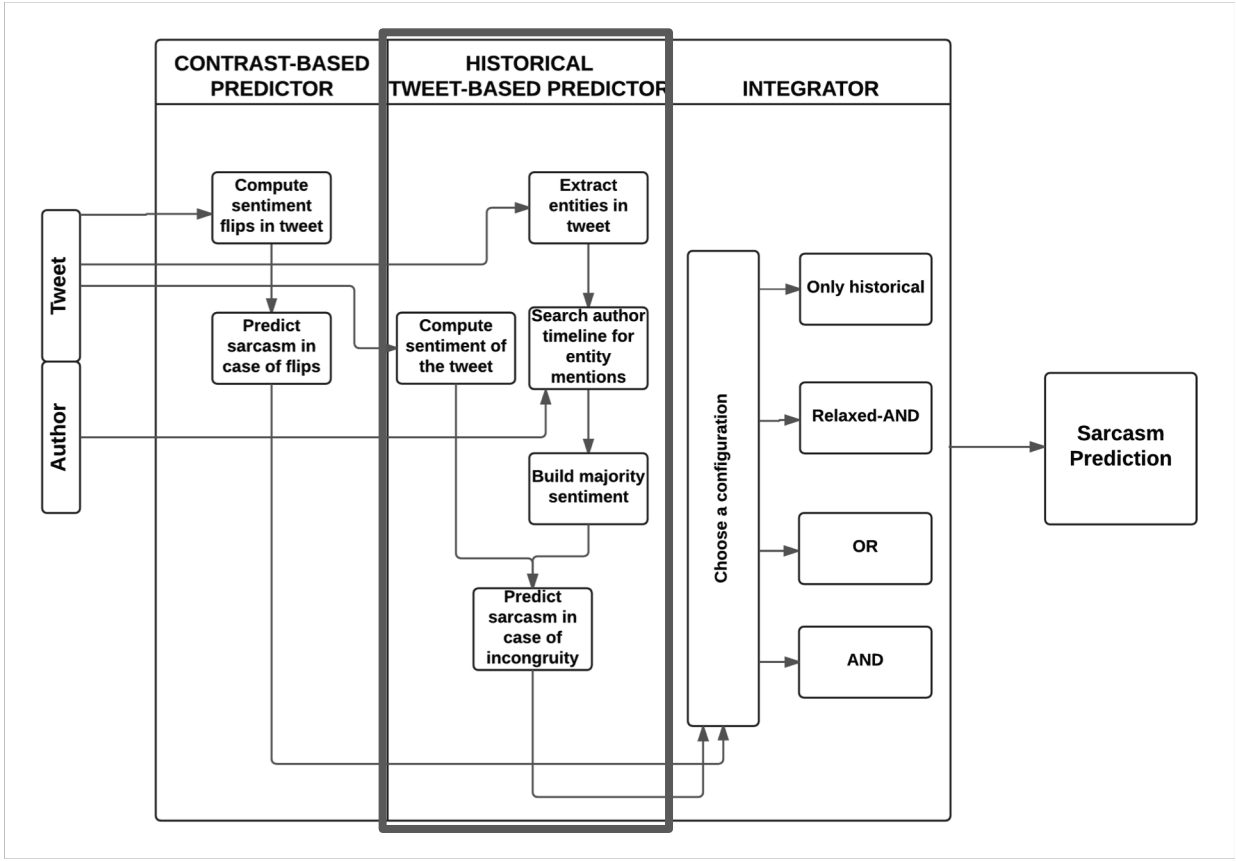
Architecture



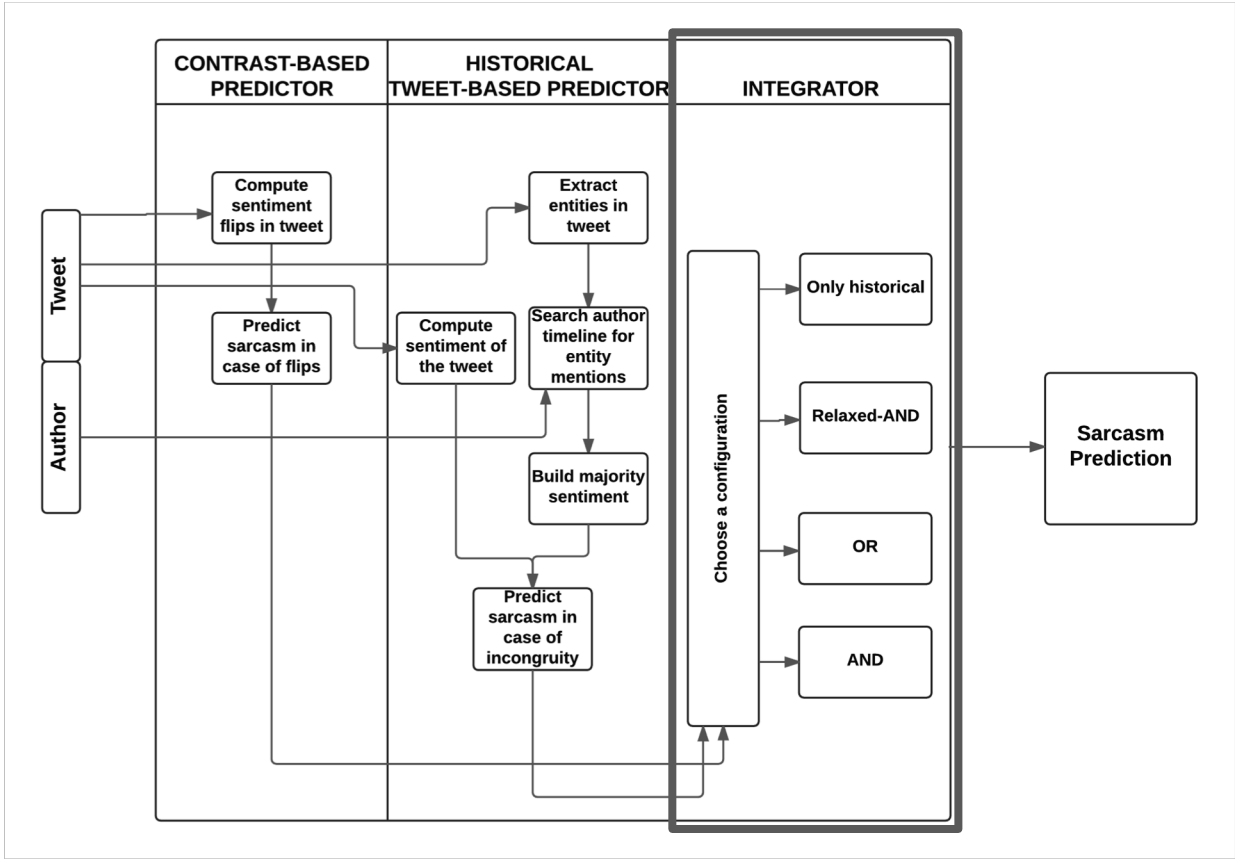
Architecture



Architecture



Architecture



Experiment Setup

Test tweets from Riloff et al. (2013): total tweets , 506 Sarcastic tweets, 1771 non-sarcastic tweets.

For contrast-based predictor: Extracted from 8000 '#sarcastic' tweets. 442 phrases with implicit negative sentiment (no implicit positive sentiment phrases because we extracted phrases from sarcastic tweets)

For historical sentiment-based predictor: Wordlists from Pang et al (2004) and Mohammad et al (2013)

Results

	Precision	Recall	F- Score
Best Reported value by Riloff et al(2013)	0.62	0.44	0.51
Only Historical Tweet-based	0.498	0.499	0.498
OR	0.791	**0.8315**	0.811
AND	0.756	0.521	0.617
Relaxed – AND	**0.8435**	0.81	**0.826**

Tweets

Relaxed-AND gives the best F-Score.

This approach assumes that the majority sentiment is expressed non-sarcastically.

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- Language model incongruity
- Historical context incongruity in a rule-based system
- **Conversational context incongruity using sequence labeling***

* Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya and Mark J Carman, 'Harnessing Sequence Labeling for Sarcasm Detection in Dialogue from TV Series Friends', CONLL 2016

Conversational Incongruity: Motivation

Target text: “I absolutely love this restaurant”.

Case 1:

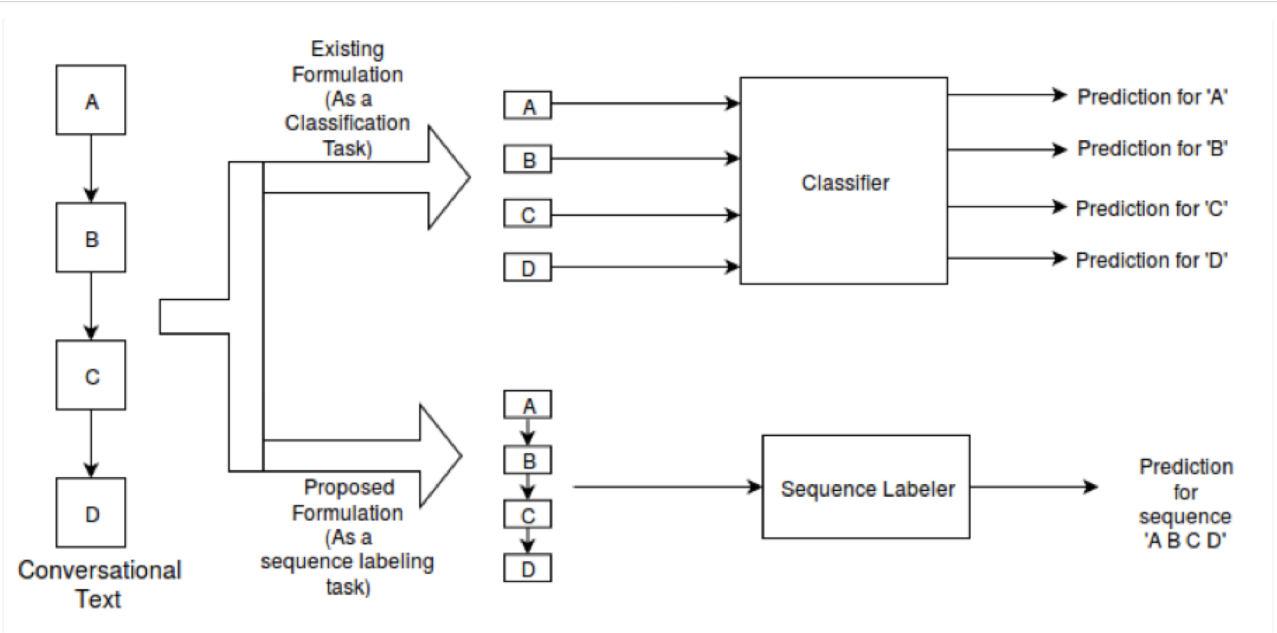
A: “Delicious food - but reasonably priced!”
B: “I absolutely love this restaurant.”

Case 2:

A: “There is a fly in my soup.”
B: “I absolutely love this restaurant.”

For conversational datasets, we propose the use of sequence labeling as compared to classification

Formulations for Sarcasm Detection of Conversational Text



Dataset (1/2)

- We create a sarcasm-labeled dataset that consists of transcripts of a comedy TV show ‘Friends’
- Our annotators are two linguists with an experience of more than 8K hours of annotation.
- Inter-annotator agreement for a subset of size 105 scenes is 0.44. Comparable with other manually annotated datasets for sarcasm detection (Tsur et al., 2010)
- 913 scenes, 17338 utterances, 1888 sarcastic utterances with an average of 18.6 utterances per scene

Dataset (2/2)

Character	% sarcastic
Phoebe	9.70
Joey	11.05
Rachel	9.74
Monica	8.87
Chandler	**22.24**
Ross	8.42

(1)

Label	% containing actions
Sarcastic	**28.23**
Non-sarcastic	23.95
All	24.43

(2)

Label	Positive Score	Negative Score
Sarcastic	**1.55**	1.20
Non-sarcastic	0.97	0.75
All	1.03	0.79

(3)

1. Percentage of sarcastic utterances
2. Percentage of utterances with actions
3. Average positive/negative scores

Experiment Setup

Sets of learning algorithms:

Classifiers: SVM (undersampled and oversampled), Naive Bayes

Sequence labelers: SVM-HMM (Yasemin et al, 2003), SEARN (Daume' et al, 2006)

Three sets of features:

Our dataset-derived features

Features from prior work: Buschmeier et al (2014), Gonzalez-Ibanez et al (2011)

Dataset-derived features

Feature	Description
Lexical features	
Spoken words	Unigrams of spoken words
Conversational context features	
Action words	Unigrams of action words
Sentiment score	Lexical sentiment score of utterance
Previous sentiment score	Lexical sentiment score of previous utterance
Author context features	
Speaker	Character who spoke this utterance
Speaker-Listener	Pair comprising of speaker of this utterance and speaker of previous utterance

Results

Algorithm	Precision (%)	Recall (%)	F-Score (%)
	Formulation as classification		
SVM (U)	83.6	48.6	57.2
SVM (O)	84.4	76.8	**79.8**
Naive Bayes	77.2	33.8	42
	Formulation as sequence labelling		
SVM-HMM	83.8	88.2	**84.2**
SEARN	82.6	83.4	82.8

Performance of our dataset-derived features

To capture conversational context incongruity, sequence labeling is a better formulation than classification for our dataset-derived features.

This holds for different combinations of dataset-derived features.

This holds for two other feature sets from past works as well.

Error Analysis

Long-range connection: In beginning of an episode, Ross says that he has never grabbed a spoon before - and at the end of the episode, he says with a sarcastic tone *“I grabbed a spoon”*.

Short expressions: *“Oh God, is it?”* ,*“Me too”, etc.*

Pretense (Camp, 2012) **:** *“Are you aware that you’re still talking?”*

Topic Drift (Eisterhold et al., 2006): When Phoebe gets irritated with another character talking for a long time, she says, “See? Vegetarianism benefits everyone”.

Discussion

Where does sequence labeling gain, as compared to classification?

We identify sentences that were correctly labeled by sequence labeling, but incorrectly labeled by classification, and classify them into types of sarcasm.

Type	%
Do not require context to be understood	18.36
Require context to be understood	**72.44**
Other	9.20

Sarcastic instances where classification fails, but sequence labeling works

Majority of sarcastic examples that the sequence labeling algorithm got right but the classification algorithm did not, require context to be understood as sarcastic.

Outline: Approaches for Sarcasm Detection

- Sentiment incongruity as features
- Semantic incongruity as word embedding-based features
- Sentiment incongruity as topic model
- Language model incongruity
- Historical context incongruity in a rule-based system
- Conversational context incongruity using sequence labeling

Summary: Sarcasm Detection

Type of Incongruity	Example	Technique/Approach	Key Findings
Sentiment Incongruity: Incongruity expressed through sentiment words	*'I love being ignored'*	Sentiment Flip-based **features**	Our features outperform two rule-based past work for two datasets
Semantic Incongruity: Incongruity through distant concepts	*'A woman needs a man like a fish needs a bicycle'*	**Word embedding**-based features	Word2Vec and Dep. Weight-based embeddings prove to be beneficial
Sentiment Incongruity: Incongruity expressed through sentiment words	*'Funerals' v/s 'Mondays'*	**Topic model** based on sentiment mixture	Sampling-based model outperforms two past works and LDA-based models
Language model incongruity: Using sentence completion	*'I love being []' : ignored or happy?*	**Sentence completion** to measure distance between expected and in-place words	As we restrict the set of words to be considered, the approach does better
Historical Incongruity: Incongruity with an author's past sentiment	*'X is the best politician ever'*	**Rule-based** technique with Historical sentiment-based predictor	Relaxed-AND configuration performs the best
Conversational Incongruity: Incongruity in a dialogue	*'There's a fly in my soup'* *'I love this restaurant'*	**Sequence labeling** algorithms	Sequence labeling works better than classification, for three sets of features

Putting it together

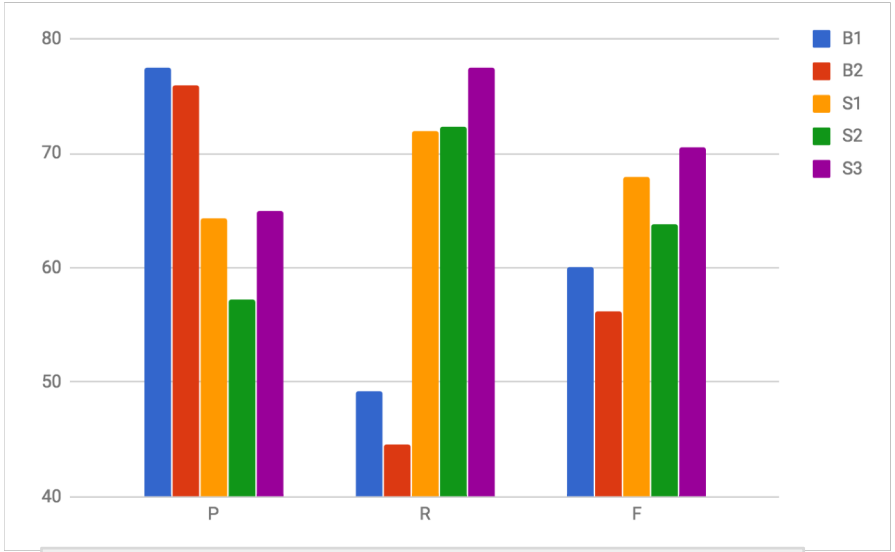
Comparison of each of the approaches against each other

Comparison of the approaches in combination

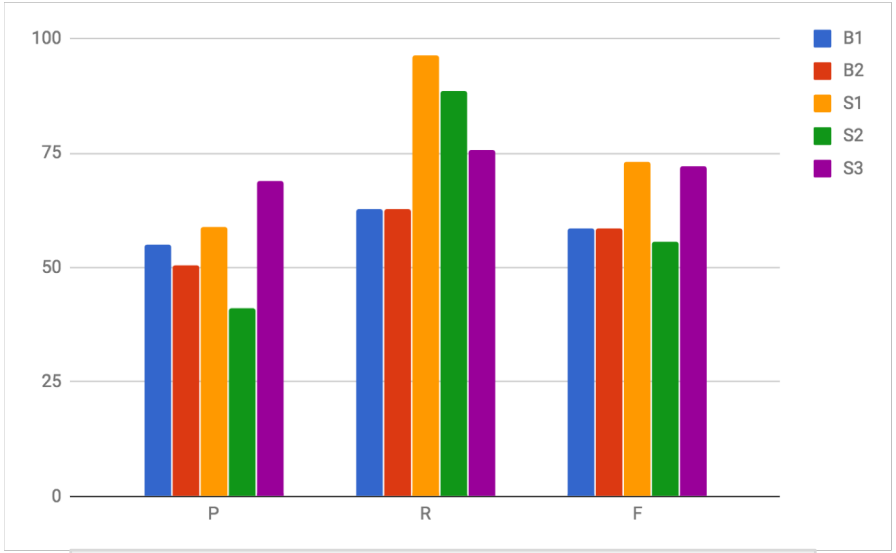
Two fold cross-validation

Three datasets: (a) Noisy text (tweets), (b) Well-formed text (book snippets), and (c) Conversational text (TV transcripts)

Comparative Evaluation: Non-Conversational Text (1/2)



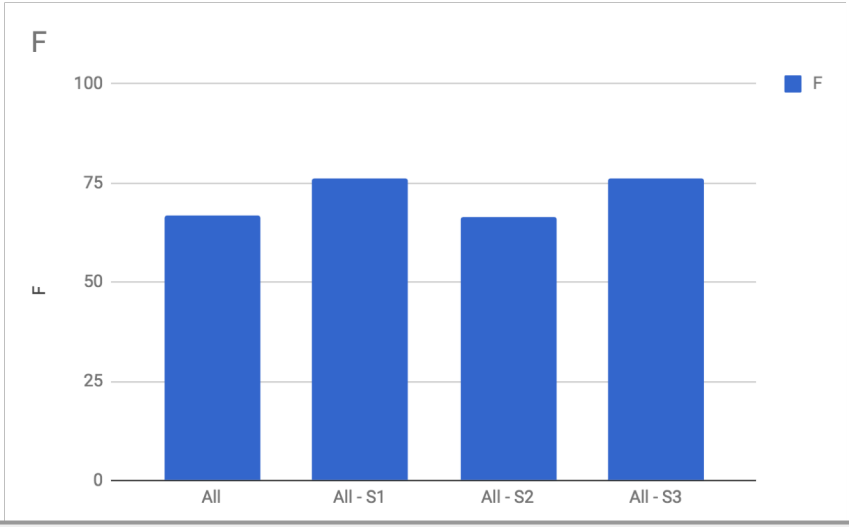
In case of book snippets, our approaches S1, S2, S3 outperform two baselines B1, B2.

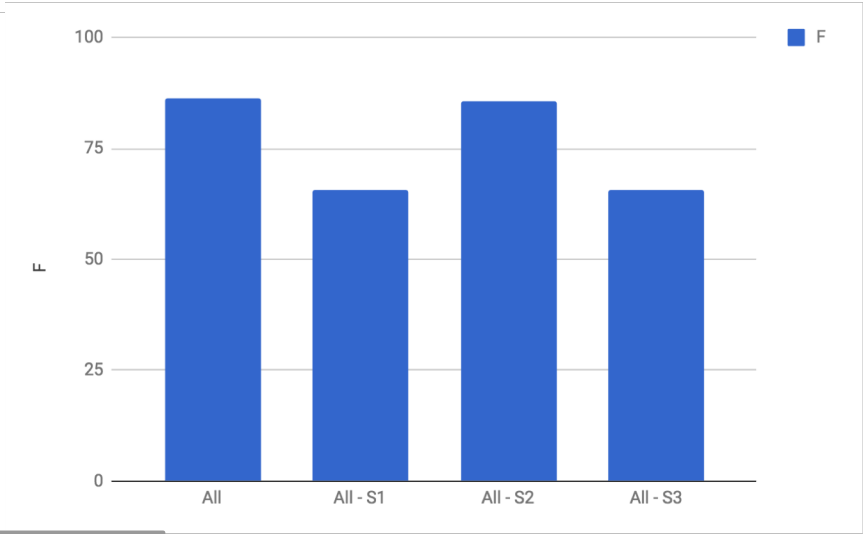


In case of tweets, our approaches S1, S3 outperform two baselines B1, B2.

S1: Sentiment Incongruity
S2: Semantic Incongruity
S3: Language Model Incongruity
B1, B2: Two past works by Buschmeier et al (2014) and Liebrecht et al (2013)

Comparative Evaluation: Non-Conversational Text (2/2)



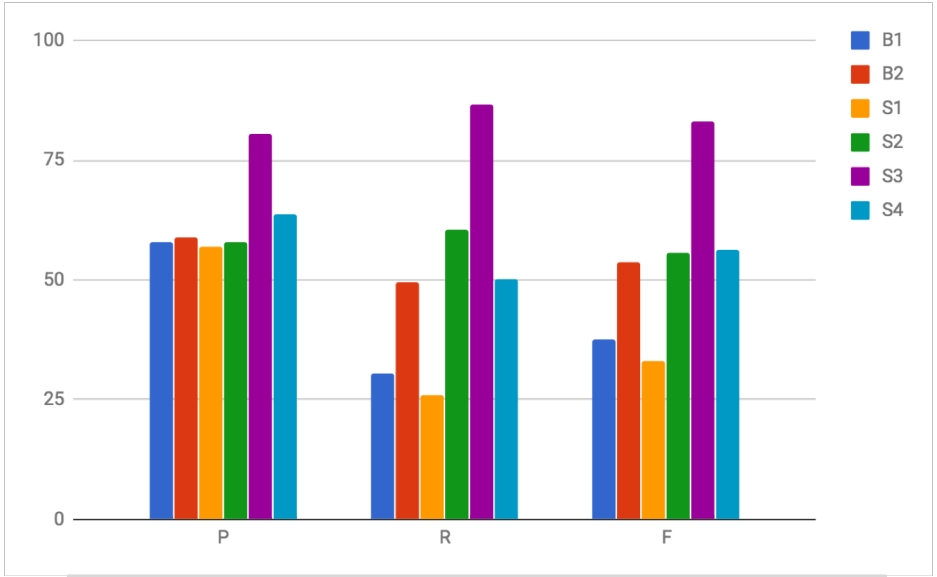


1. For **well-formed text (book snippets; left)**, a combination of semantic-based incongruity and language model-based incongruity results in the best performance.

2. For **noisy text (tweets; right)**, the ensemble works the best. The degradation is highest in case of sentiment-based incongruity.

S1: Sentiment Incongruity
S2: Semantic Incongruity
S3: Language Model Incongruity
S4: Conversational Context Incongruity
B1, B2: Two past works by Buschmeier et al (2014) and Liebrecht et al (2013)

Comparative Evaluation: Conversational Text (1/2)

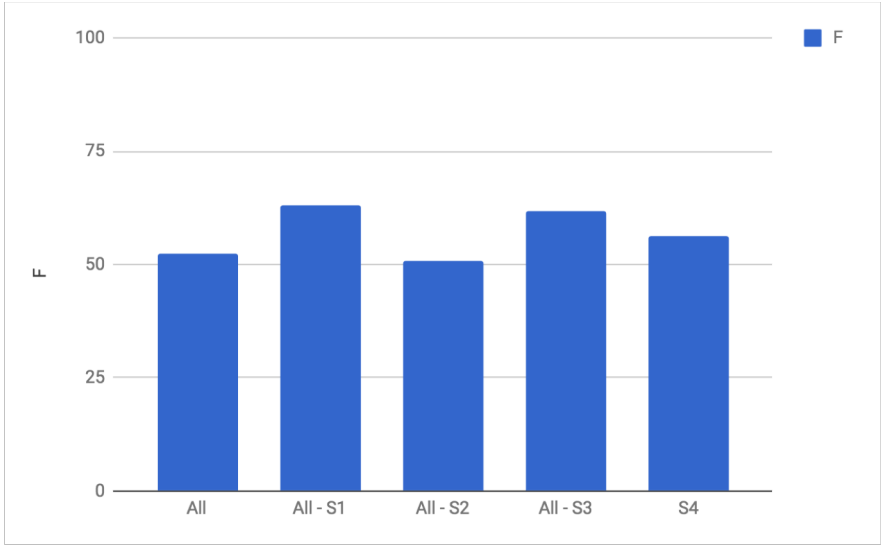


In case of book snippets, our approaches S2, S3, S4 outperform two baselines B1, B2.

S1: Sentiment Incongruity
S2: Semantic Incongruity
S3: Language Model Incongruity
S4: Conversational Context Incongruity
B1, B2: Two past works by Buschmeier et al (2014) and Liebrecht et al (2013)

Comparative Evaluation: Conversational Text (2/2)

For **short conversational text**, a combination of language model incongruity and semantic incongruity result in the best performance.



S1: Sentiment Incongruity
S2: Semantic Incongruity
S3: Language Model Incongruity
S4: Conversational Context Incongruity
B1, B2: Two past works by Buschmeier et al (2014) and Liebrecht et al (2013)

Outline

Introduction

Sarcasm, Motivation, Incongruity, Work Overview

Sarcasm Detection

Sarcasm Detection Using Incongruity Within Target Text

Sarcasm Detection Using Incongruity Outside Target Text

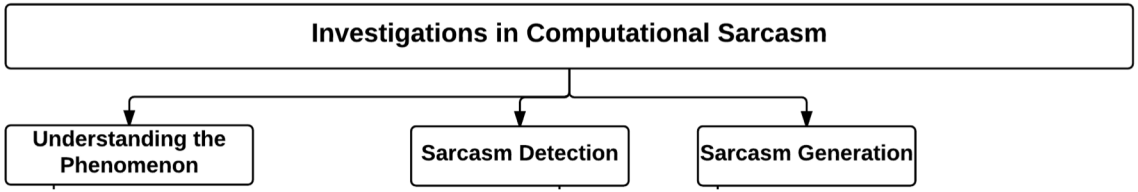
Beyond detection

Understanding the phenomenon, Sarcasm generation

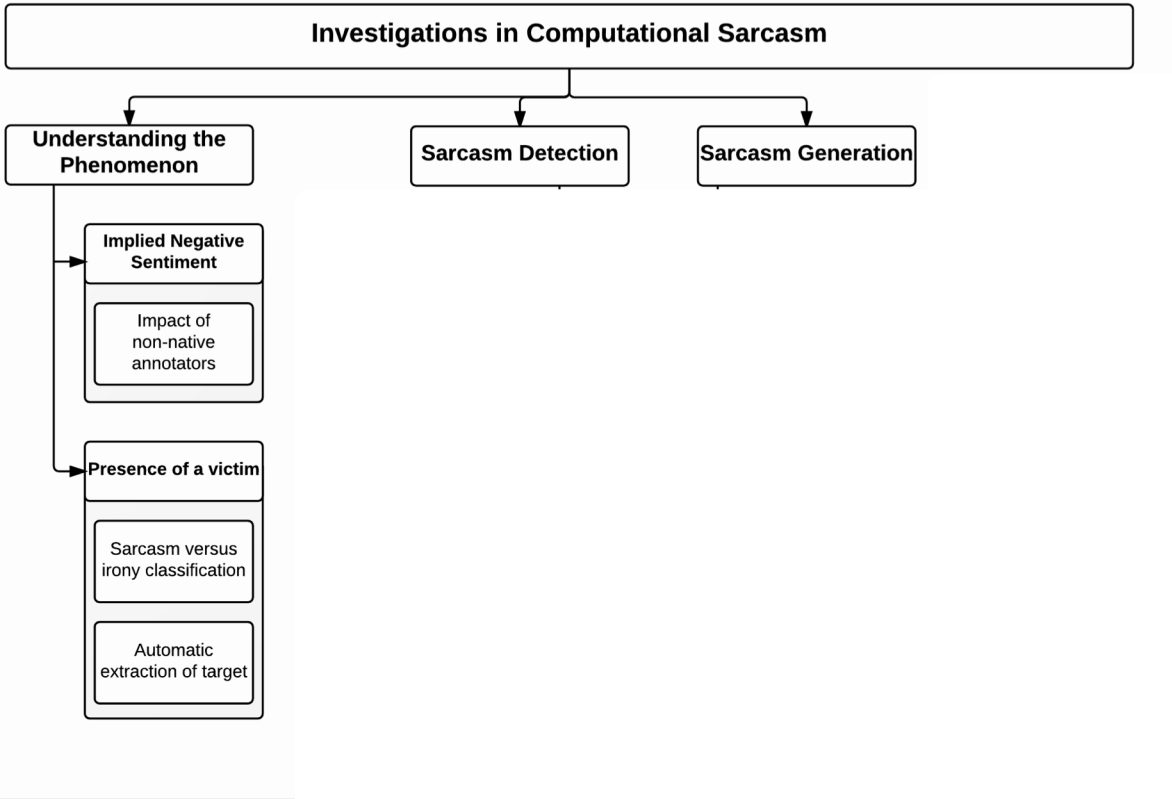
Conclusion

Summary, pointers to future work

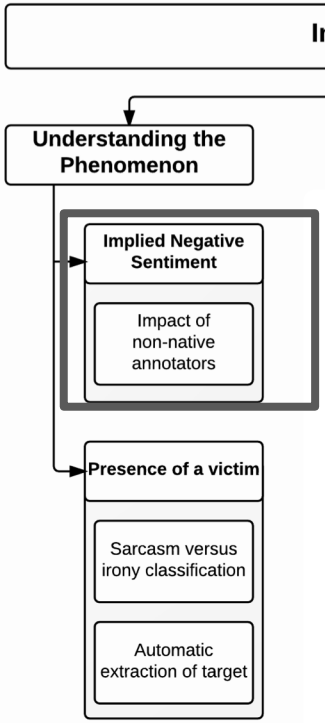
Beyond sarcasm detection



Beyond sarcasm detection



Beyond sarcasm detection



Premise: Indian annotators annotate two datasets annotated by American annotators

Experimentation:

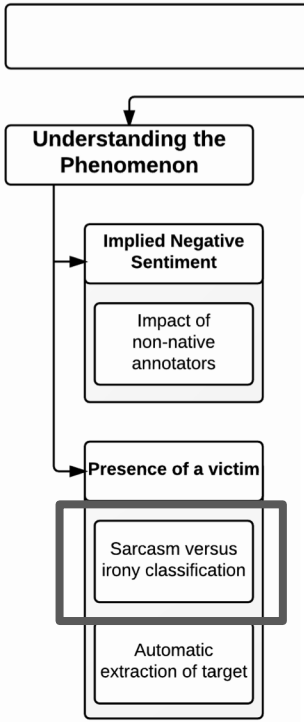
(a) Inter-annotator agreement,

(b) Relative performance of training on one set of annotations, testing on another

'It is sunny outside but I am at work. Yay'

Aditya Joshi, Pushpak Bhattacharyya, Mark J Carman, Jaya Saraswati and Rajita Shukla, 'How Do Cultural Differences Impact the Quality of Sarcasm Annotation?: A Case Study of Indian Annotators and American Text', 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities at ACL 2016

Beyond sarcasm detection

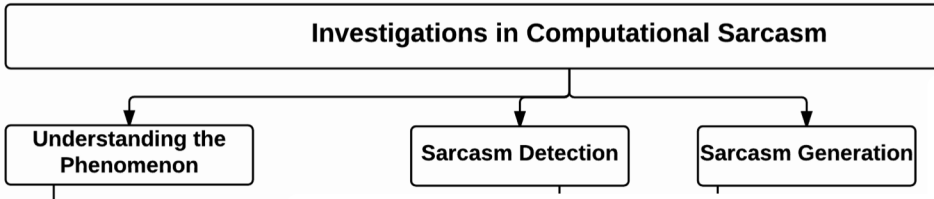


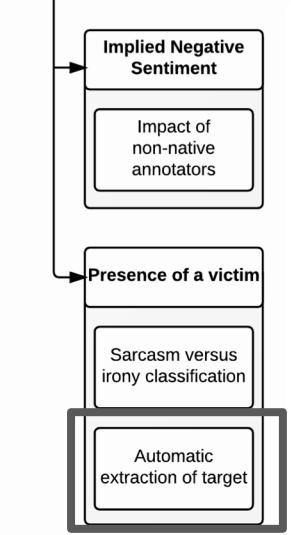
Premise: Annotators label book snippets as either sarcasm or irony
Experimentation:
(a) Inter-annotator agreement,
(b) Do features for sarcasm versus non-sarcasm work well for sarcasm versus irony?

'The scientist who discovered the cure for the disease died of the disease'

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark J Carman, Meghna Singh, Jaya Saraswati and Rajita Shukla, 'How Challenging is Sarcasm versus Irony Classification?: A Study With a Dataset from English Literature', Australasian Language Technology Association (ALTA) 2016

Beyond sarcasm detection

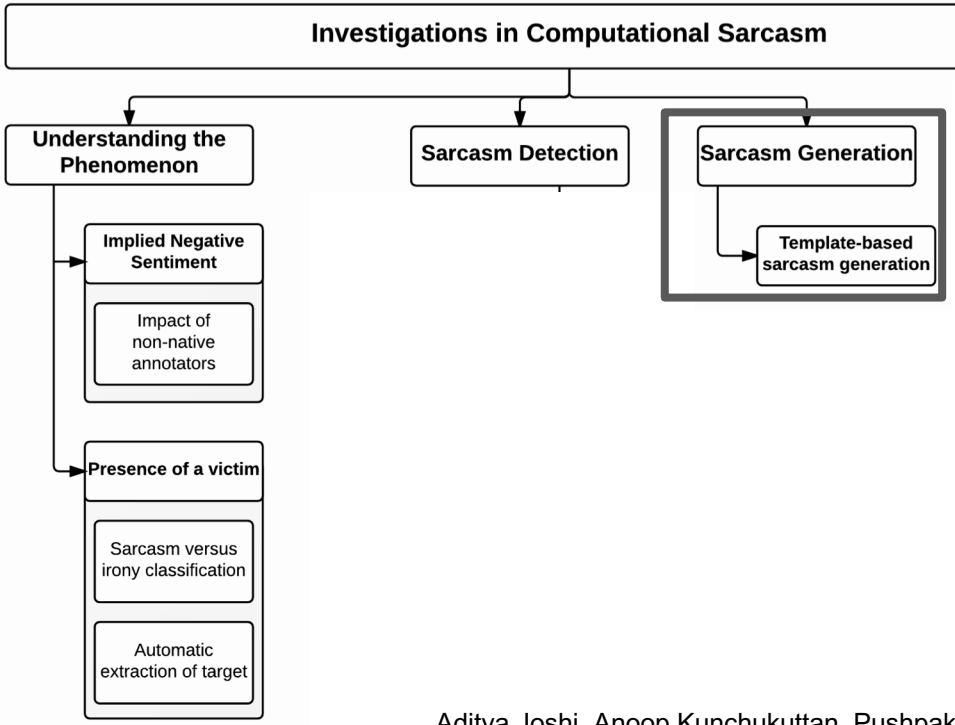




Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya, Mark J Carman, 'Automated Sarcasm Target Identification', LREC 2018.

Premise: Given a sarcastic text, extract a phrase indicating target of sarcasm
Experimentation:
(a) A hybrid (rule-based + statistical) approach for sarcasm target identification
(b) How does this compare against past approaches for sentiment target identification?
'I did not make it big in Hollywood because my writing was not bad enough': 'Hollywood'

Beyond sarcasm detection

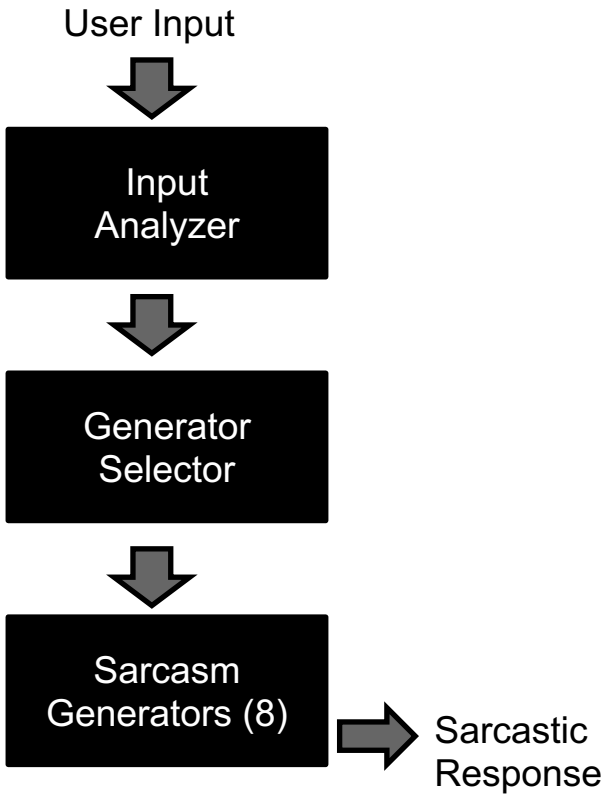


Premise: NLG system for sarcasm
Experimentation:
(a) Using templates to generate sarcasm,
(b) Qualitative evaluation by human evaluators

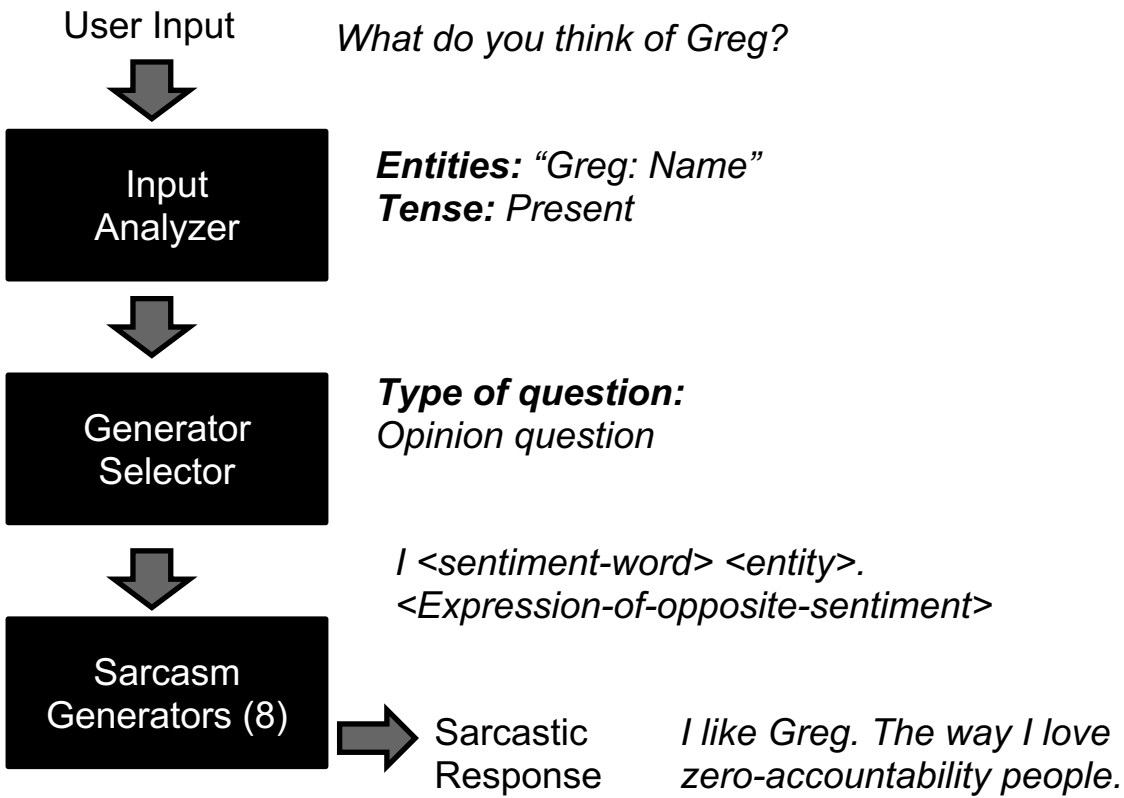
Input: What do you think of Greg?
Output: I like Greg. The way I love zero accountability people.

Aditya Joshi, Anoop Kunchukuttan, Pushpak Bhattacharyya, Mark J Carman, SarcasmBot: An open-source sarcasm-generation module for chatbots, WISDOM at KDD 2015

Template-based sarcasm generation: SarcasmBot



Template-based sarcasm generation: SarcasmBot



Sarcasm Generators

Sarcasm Generator	Description
(a) Offensive Word Response Generator	In case an offensive word is used in user input, select a placeholder from a set of responses.
(b) Opposite Polarity Verb-Situation Generator	Randomly select a verb. Compute its sentiment. Discover a situation which is opposite in sentiment.
(c) Opposite Polarity Person-Attribute Generator	Randomly select a named entity. Select incongruent pairs of famous people.
(d) Irrealis Sarcasm Generator	Create a hypothetical situation that is impossible by selecting from a set of undesirable situations.
(e) Hyperbole Generator	Select a noun phrase in the user input. Generate a hyperbole with a 'best ever' style regular expression.
(f) Incongruent Reason Generator	Select an unrelated reason as a response for a user input.
(g) Sentiment-based Sarcasm Generator	Compute sentiment of user input. Generate a response opposite in sentiment.
(h) Random Response Generator	Select one positive exclamation and one negative exclamation randomly from a set of exclamations. Place them together.

Evaluation

Expt 1: Average Scores on three parameters

Evaluation Parameter	Average
Coherence	0.698
Grammatical correctness	0.903
Sarcastic nature	0.806

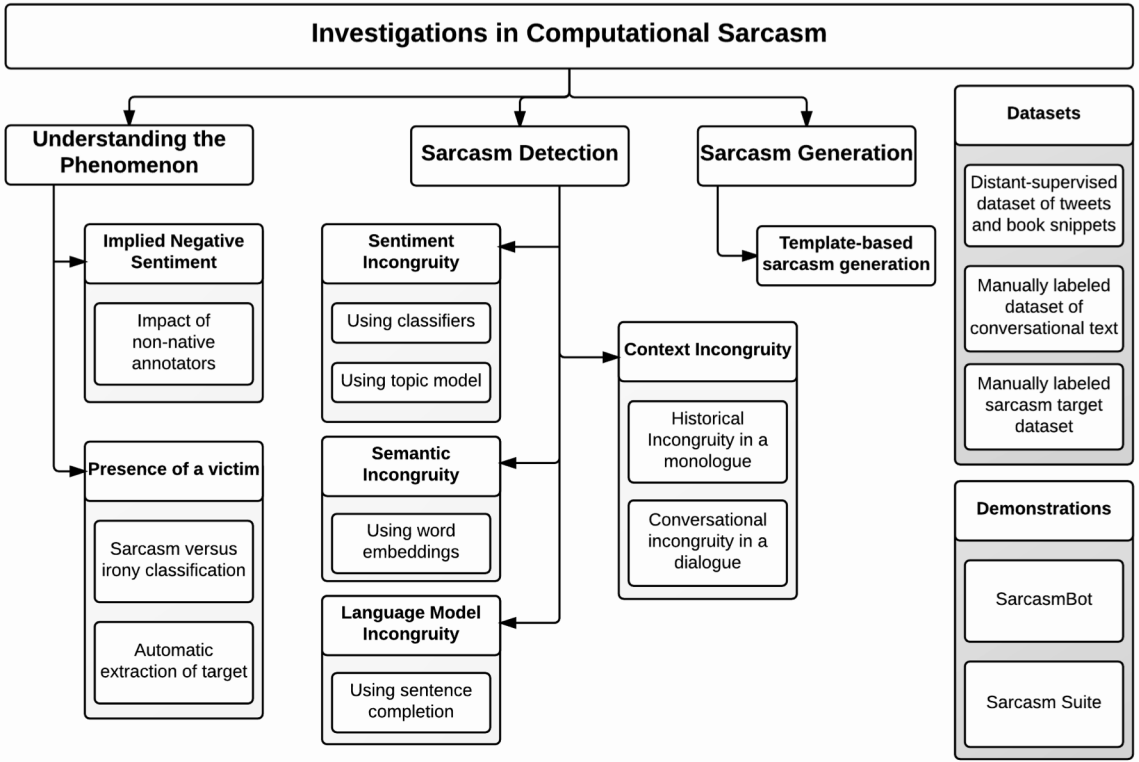
The chatbot is rated ~0.7 or above for the three quality parameters.

Expt 2: Identifying between ALICE and SarcasmBot

Strategy	Accuracy (%)
At least one evaluator is correct	87.09
Majority evaluators are correct	70.97
All evaluators are correct	61.29

In ~87% cases, at least one evaluator could correctly spot the output of SarcasmBot from two possible outputs.

Work Overview



Outline

Introduction

Sarcasm,
Motivation,
Incongruity,
Work Overview

Sarcasm Detection

Sarcasm Detection Using Incongruity Within Target Text

Sarcasm Detection Using Incongruity Outside Target Text

Beyond detection

Understanding the phenomenon,
Sarcasm generation

Conclusion

Summary, pointers to future work

Summary (1/2)

- We model computational sarcasm through the linguistic notion of incongruity
- To detect sarcasm, we employ:
  - Sentiment incongruity as sentiment change-based features
  - Semantic incongruity as word embedding-based features
  - Sentiment incongruity as sentiment distributions in a topic model
  - Language model incongruity as distance between expected and observed words
  - Historical incongruity as discordance with historical sentiment of the author
  - Conversational incongruity using sequence labeling
- We recommend which of the above/which combinations work well for three types of text

Summary (2/2)

- To generate sarcasm, we use templates to simulate incongruity
- To understand the phenomenon of sarcasm, we look at:
  - How well non-native annotators understand sarcasm
  - How the presence of a victim affects annotation
  - How sarcasm targets can be identified

- Our investigations in computational sarcasm result in a set of approaches for sarcasm detection, and two novel problems: sarcasm generation and sarcasm target identification

Publications

Conference Papers

1. Aditya Joshi, Vinita Sharma and Pushpak Bhattacharyya, ‘Harnessing context incongruity for sarcasm detection’, **ACL-IJCNLP 2015**.

2. Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya and Mark J Carman, 'Harnessing Sequence Labeling for Sarcasm Detection in Dialogue from TV Series Friends', **CONLL 2016**.

3. Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya and Mark J Carman, 'Are Word Embedding-based Features Useful for Sarcasm Detection?‘, **EMNLP 2016**.

4. Aditya Joshi, Diptesh Kanojia, Pushpak Bhattacharyya, Mark J Carman, 'Sarcasm Suite: A browser-based engine for sarcasm detection and generation', **AAAI-17**. (Demonstrations)

5. Aditya Joshi, Samarth Agrawal, Pushpak Bhattacharyya, Mark J Carman, 'Expect the unexpected: Harnessing Sentence Completion for Sarcasm Detection', **PACLING 2017.**

6. Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya and Mark Carman, Sarcasm Target Identification: Dataset and An Introductory Approach, **LREC 2018.**

Publications

Workshop & Symposium Papers

1. Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya and Mark J Carman, ‘Your sentiment precedes you: Using an author's historical tweets to predict sarcasm’, **WASSA workshop at EMNLP 2015.**
2. Aditya Joshi, Anoop Kunchukuttan, Pushpak Bhattacharyya and Mark J Carman, ‘SarcasmBot: An open-source sarcasm-generation module for chatbots’, **WISDOM workshop at SIGKDD 2015.**
3. Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark J Carman, Meghna Singh, Jaya Saraswati and Rajita Shukla, 'How Challenging is Sarcasm versus Irony Classification?: A Study With a Dataset from English Literature', **ALTA 2016**.
4. Aditya Joshi, Pushpak Bhattacharyya, Mark J Carman, Jaya Saraswati and Rajita Shukla, 'How Do Cultural Differences Impact the Quality of Sarcasm Annotation?: A Case Study of Indian Annotators and American Text', **LATECH workshop at ACL 2016**.
5. Aditya Joshi, Prayas Jain, Pushpak Bhattacharyya and Mark J Carman, ‘Who would have thought of that!’: A Novel Hierarchical Topic Model for Extraction of Sarcasm-prevalent Topics and Sarcasm Detection', **ExPROM workshop at COLING 2016.**

To know more…

Code (arranged by publications):

https://github.com/adityajo/ComputationalSarcasm

Survey Paper:

Aditya Joshi, Pushpak Bhattacharyya, Mark J Carman, 'Automatic Sarcasm Detection: A Survey', Vol. 50, Issue 5, ACM Computing Surveys, 2017.

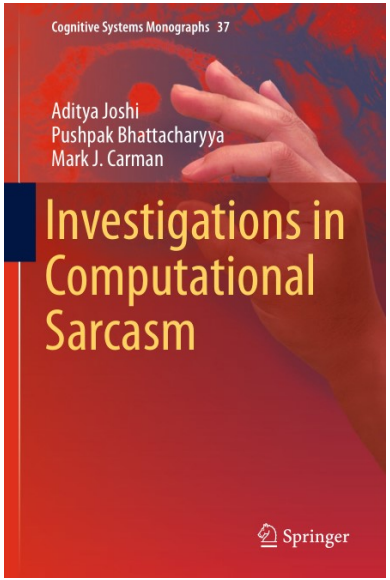
EMNLP 2017 Tutorial 'Computational Sarcasm':

http://www.cfilt.iitb.ac.in/tutorial-computational-sarcasm.pdf

TEDx Talk 'Detecting sarcasm, combating hate':

https://www.youtube.com/watch?v=1tJAoAYGzXs

Monograph:



References

S. Bird. NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions, pages 69–72. Association for Computational Linguistics, 2006.

R. S. Wallace. The anatomy of ALICE. Springer, 2009.

R. W. Gibbs. The poetics of mind: Figurative thought, language, and understanding. Cambridge University Press, 1994.

S. L. Ivanko and P. M. Pexman. Context incongruity and irony processing. Discourse Processes, 35(3):241–279, 2003.

Ankit Ramteke, Akshat Malu, Pushpak Bhattacharyya and Saketha Nath, *Detecting Turnarounds in Sentiment Analysis: Thwarting*, ACL 2013, Sofia, Bulgaria, 4-9 August, 2013

Riloff, Ellen, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. "Sarcasm as Contrast between a Positive Sentiment and Negative Situation." In *EMNLP*, vol. 13, pp. 704-714. 2013.

M. A. Walker, J. E. F. Tree, P. Anand, R. Abbott, and J. King. A corpus for research on deliberation and debate. In LREC, pages 812–817, 2012.

T. Joachims. Training linear svms in linear time. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 217–226. ACM, 2006a.

Melamud, O., Goldberger, J., & Dagan, I. (2016, August). context2vec: Learning generic context embedding with bidirectional lstm. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (pp. 51-61).

S. M. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. 29(3):436–465, 2013.

B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In ACL, page 271. Association for Computational Linguistics, 2004.

O. Tsur, D. Davidov, and A. Rappoport. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In ICWSM, 2010a.

K. Buschmeier, P. Cimiano, and R. Klinger. An impact analysis of features in a classification approach to irony detection in product reviews. ACL 2014, page 42, 2014.

S. Muresan, R. Gonzalez-Ibanez, D. Ghosh, and N. Wacholder. Identification of nonliteral language in social media: A case study on sarcasm. Journal of the Association for Information Science and Technology, 2016.

J. Eisterhold, S. Attardo, and D. Boxer. Reactions to irony in discourse: Evidence for the least disruption principle. Journal of Pragmatics, 38(8):1239–1256, 2006.

E. Camp. Sarcasm, pretense, and the semantics/pragmatics distinction*. Noˆus, 46(4):587–634, 2012.

"*Sarcasm is the lowest form of wit,*
but the highest form of intelligence."

Oscar Wilde (1854 - 1900)