

CS772: Deep Learning for Natural Language Processing

CNN, Eye Tracking, Sarcasm

Pushpak Bhattacharyya

Computer Science and Engineering
Department

IIT Bombay

Week 11 of 14th Mar, 2022

Detailing out CNN layers

Credit: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

CNN stages

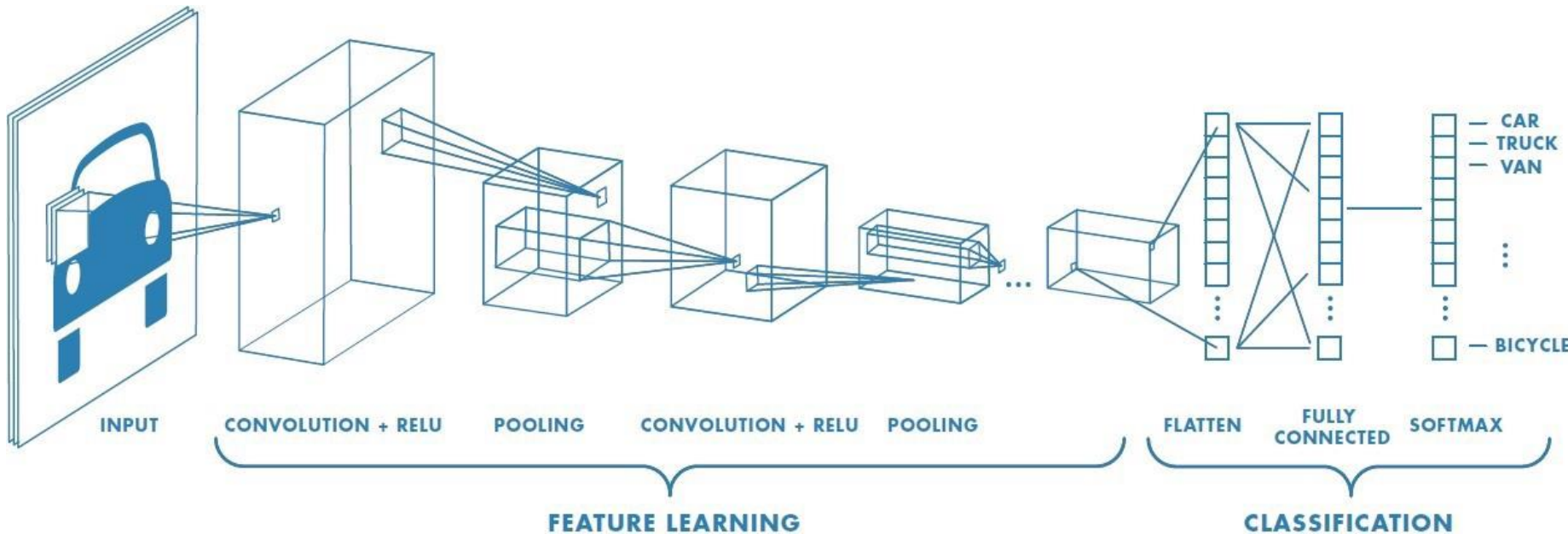


Image Credit: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Another depiction

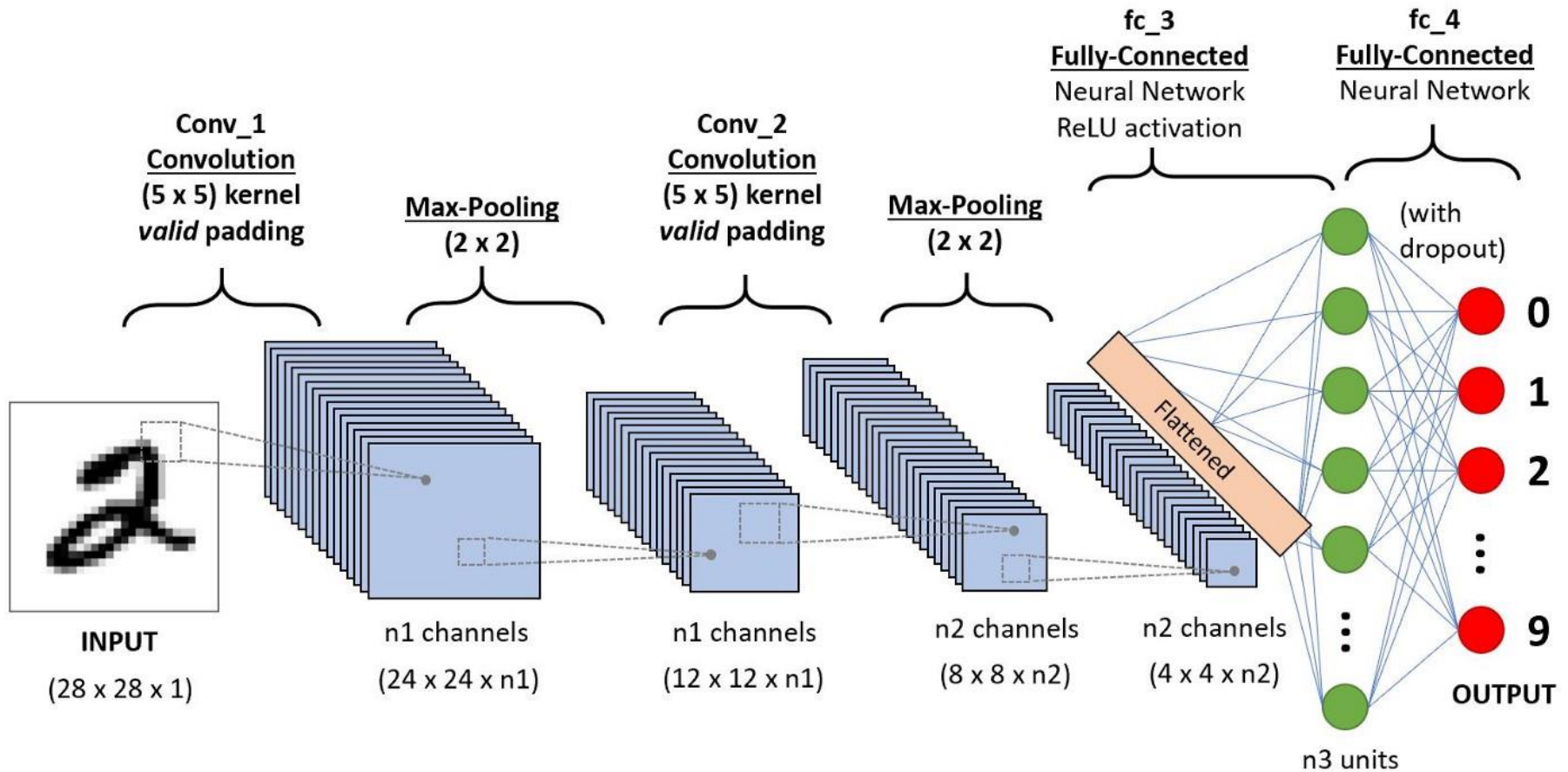
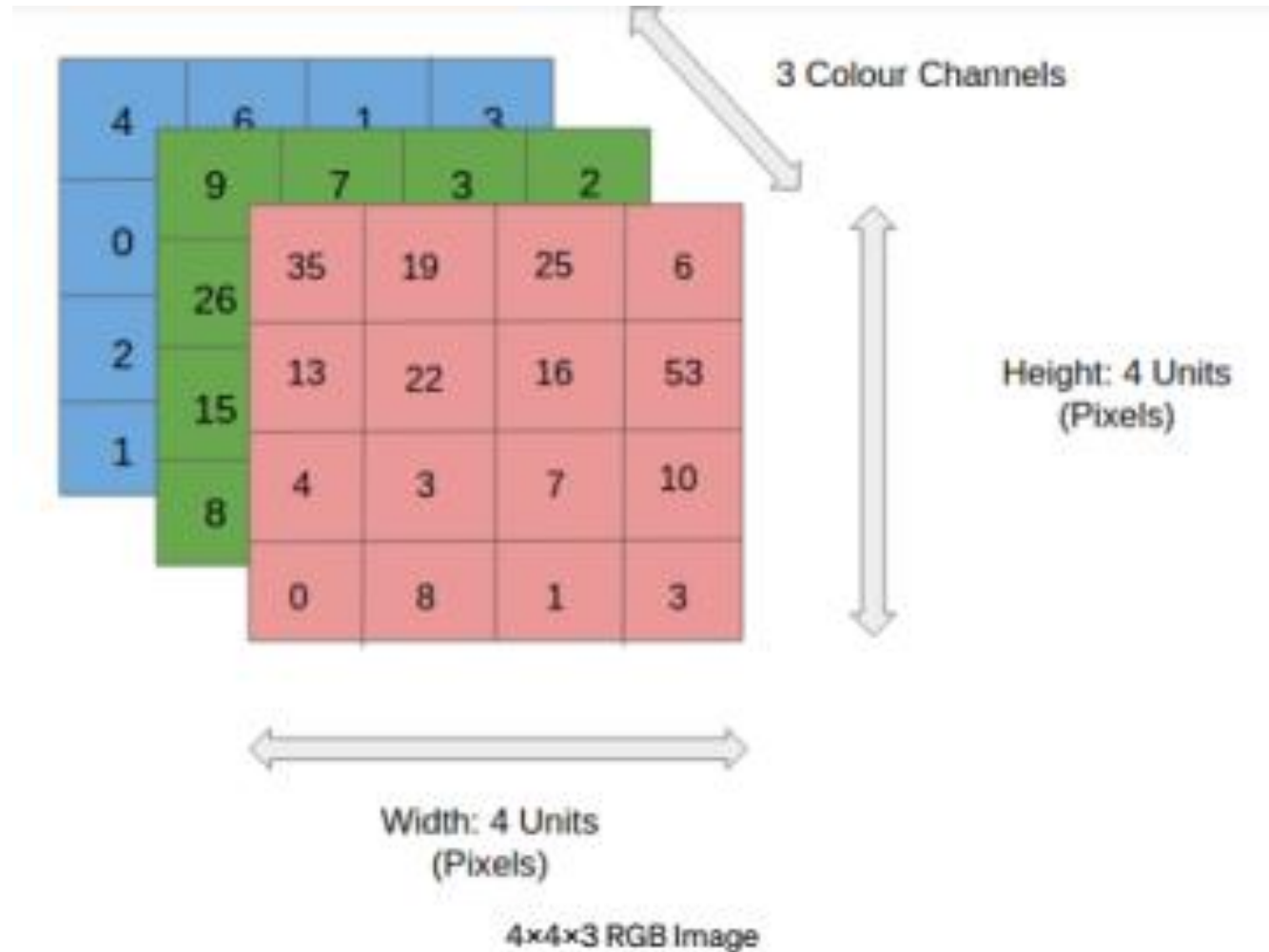
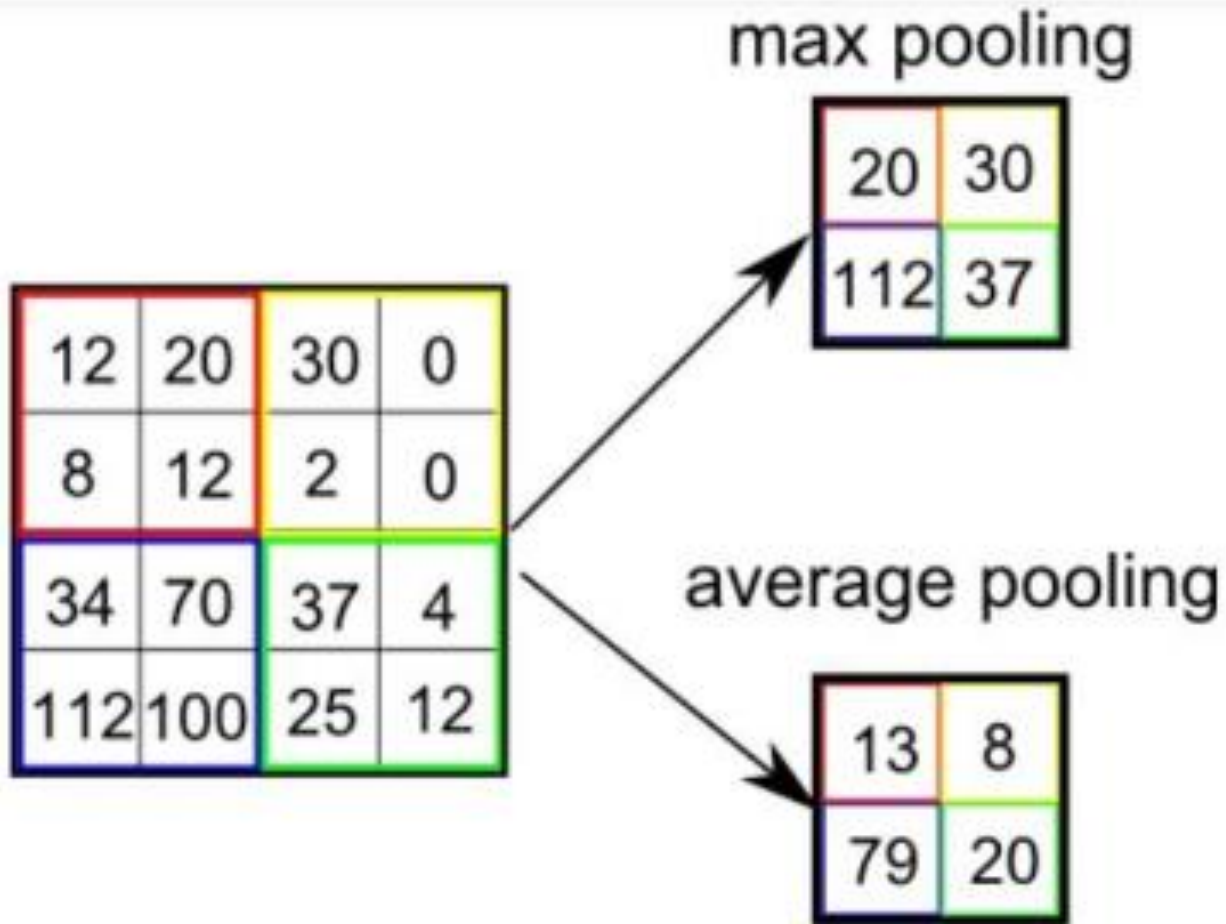


Image Credit: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Channelized Image

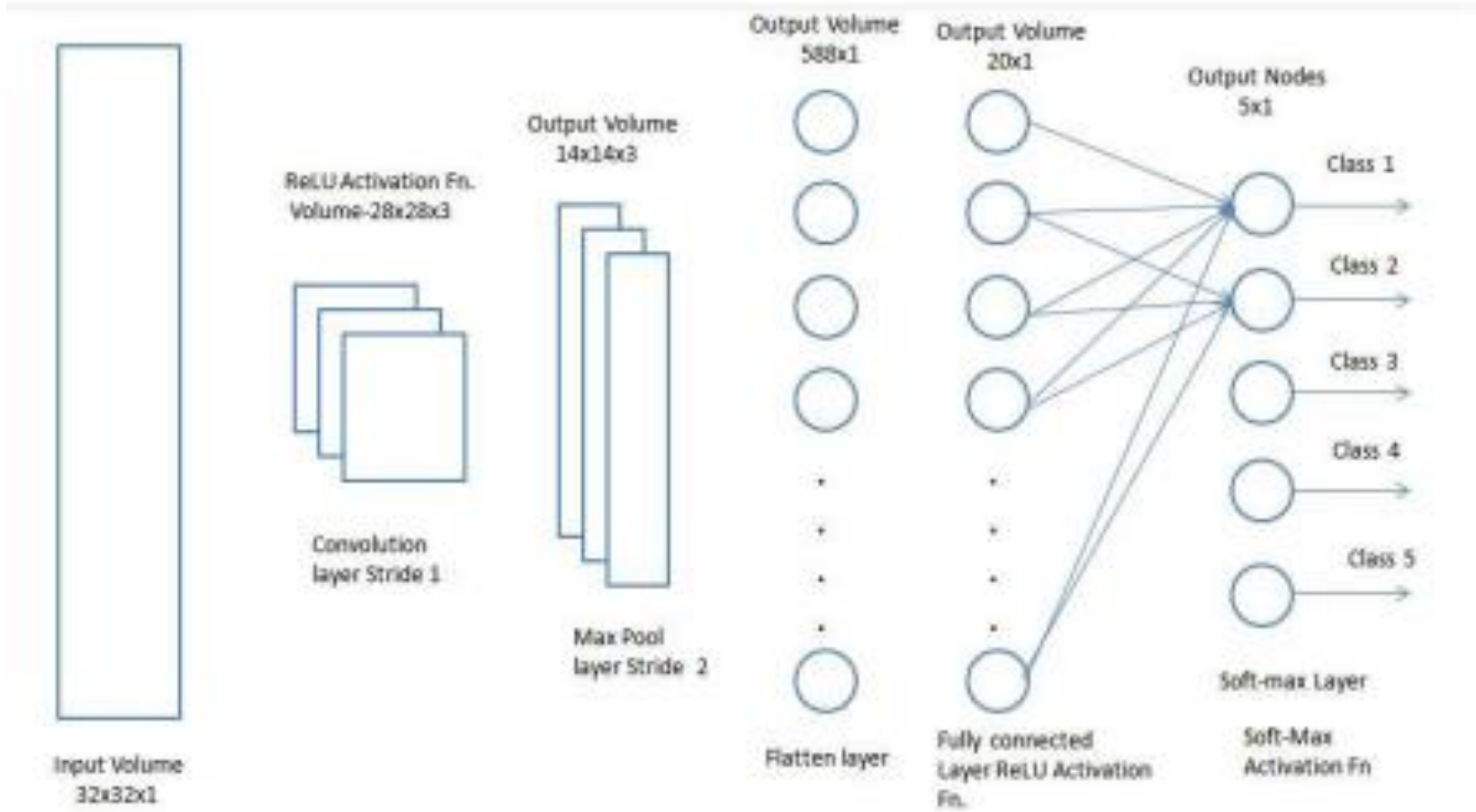


Pooling



Types of Pooling

Complete Architecture



Pooling Layer

- “Pooling” involves sliding a two-dimensional filter over each channel of feature map
- Effect: summarizing the features
- For a feature map having dimensions $n_h \times n_w \times n_c$, the output dimension after pooling is

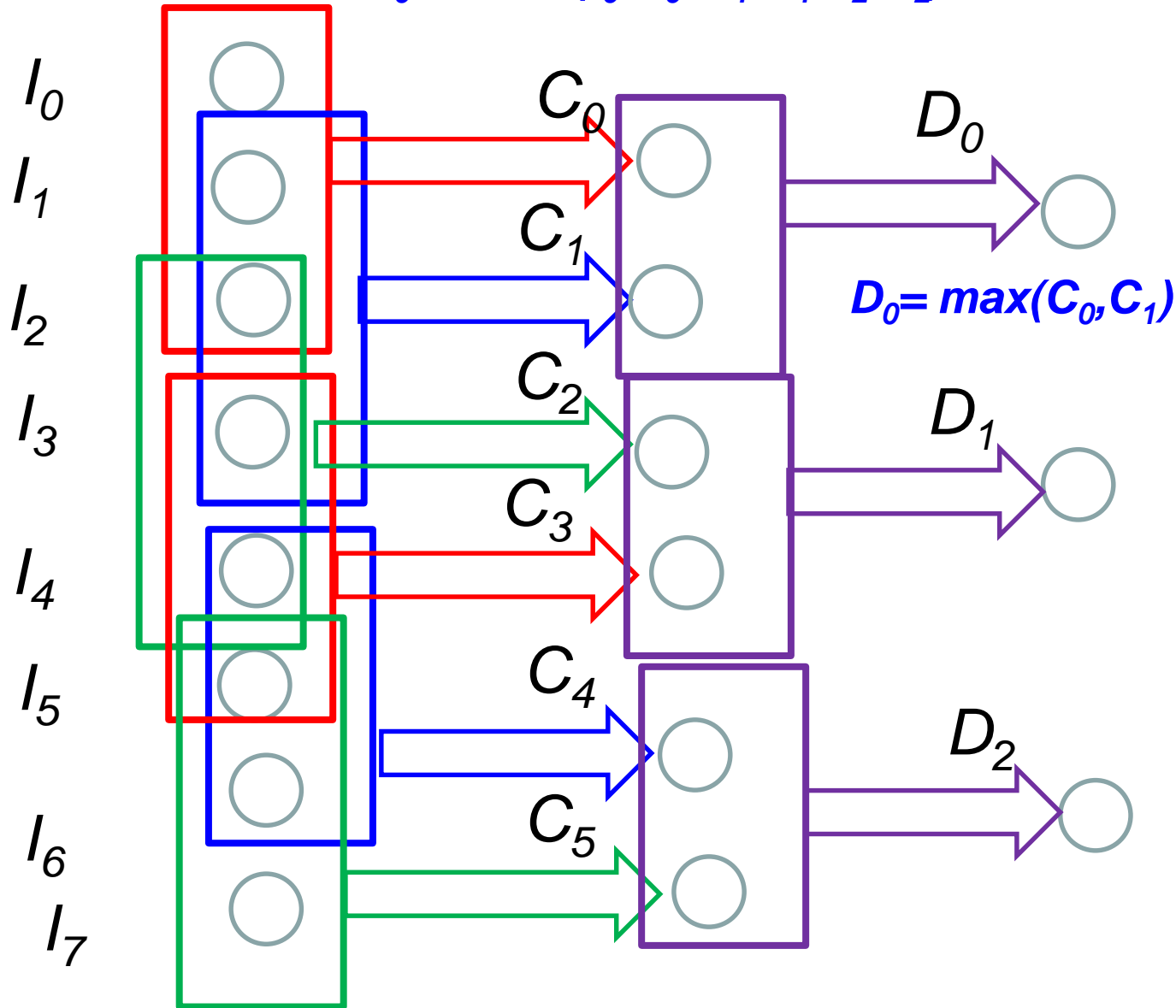
$$\left(\frac{n_h - f_h + 1}{s} \right) \cdot \left(\frac{n_w - f_w + 1}{s} \right) (n_c)$$

where, n_h = height of feature map, n_w =width, n_c = number of channels, f_h =height of filter, f_w =width of filter, s =stride length

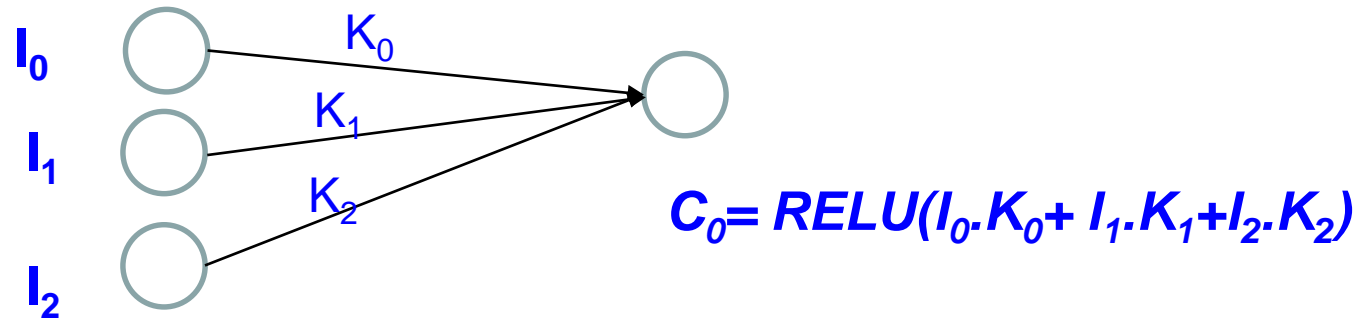
Learning in CNN

First Kernel+RELU+POOLING

$$C_0 = \text{RELU}(I_0 \cdot K_0 + I_1 \cdot K_1 + I_2 \cdot K_2); \text{ Ks are kernel "weights"}$$



Fleshing out the details



Input vector I

New $K_0 = \text{old } K_0 + \text{sum of } \Delta K_0 \text{ s across } C_0, C_1 \dots C_5$

This addition does not violate gradient descent rule

Normal BP works

- Backpropagate from the final layer of softmax.
- When it comes to the first convolution layer, post the changes in the weights, maintaining the constraint that kernel values are parameter-shared
- Nothing special needs to be done for RELU and MAX functions

Another depiction

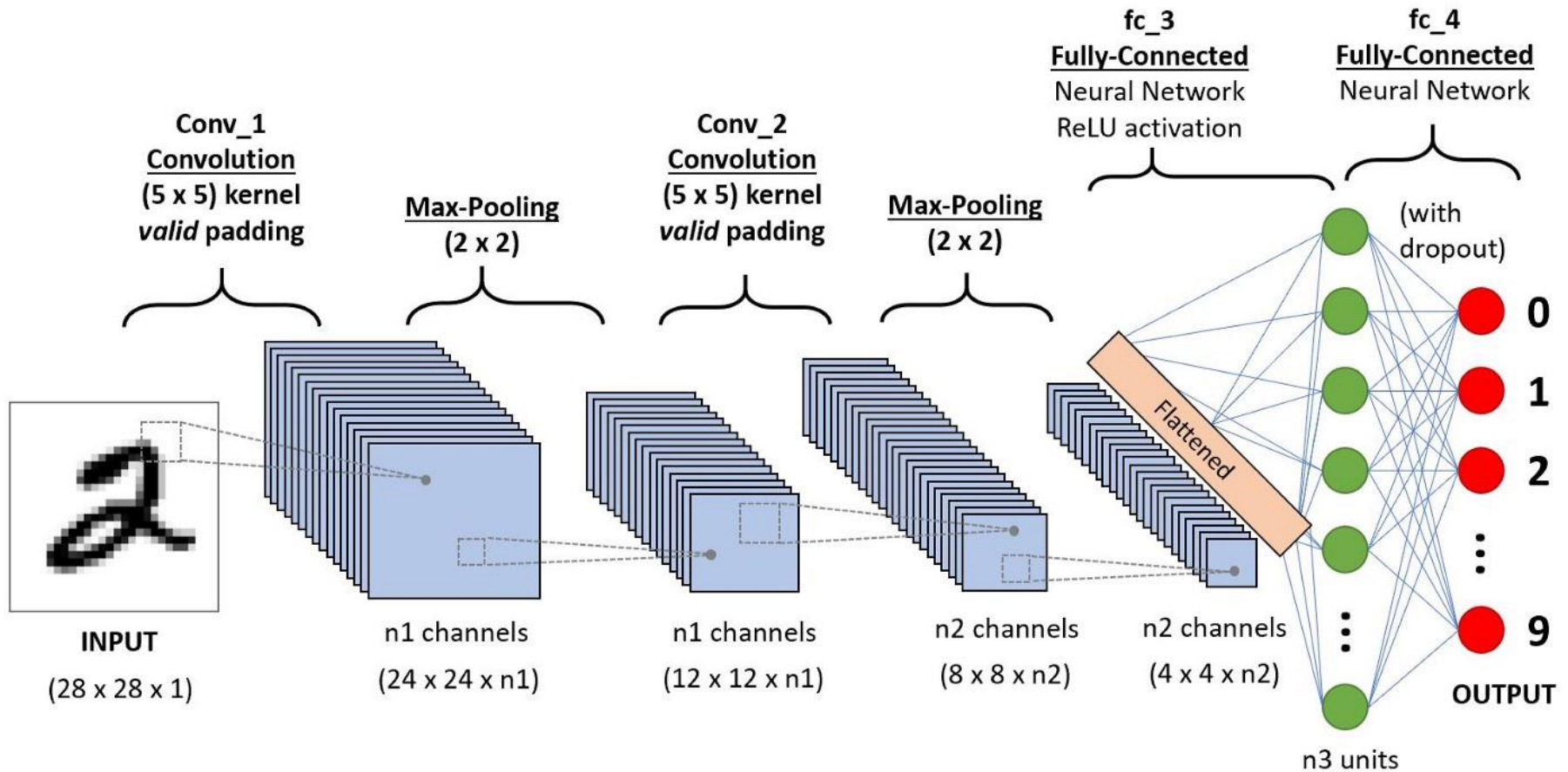


Image Credit: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

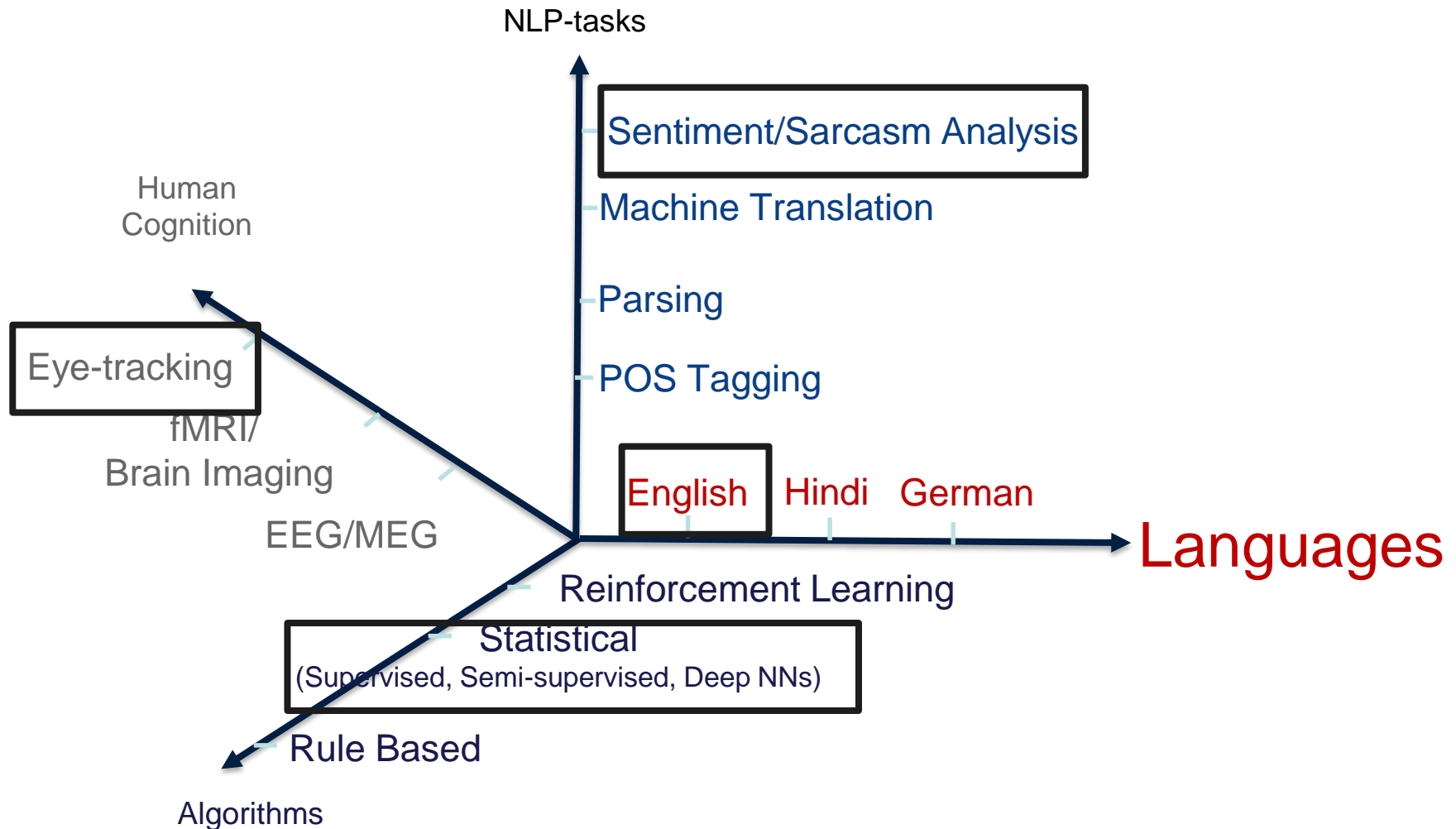
An application: Sarcasm Detection

Illustrates use of CNN Channels

Sarcasm Detection: a sub-problem of Sentiment and Emotion Analysis

Sentiment Analysis: The task of identifying if a certain piece of text contains any opinion, emotion or other forms of affective content.

NLP-trinity (augmented)



Sarcasm: Etymology

- Greek: '*sarkasmós*': 'to tear flesh with teeth'
- Sanskrit: '*vakrokti*': 'a twisted (*vakra*) utterance (*ukti*)'

Foundation: Irony

Mean opposite of what is on surface

“A form of irony that is intended to express contempt or ridicule.”

The Free Dictionary

“Verbal irony that expresses negative and critical attitudes toward persons or events.”

(Kreuz and Glucksberg, 1989)

“The use of irony to mock or convey contempt.”

Oxford Dictionary

“Irony that is especially bitter and caustic”

(Gibbs, 1994)

Allied concept: **Humble Bragging**- “Oh my life is miserable, have to sign 500 autographs a day!!

Types of Sarcasm

Sarcasm (Camp, 2012)

Propositional

A proposition that is intended to be sarcastic.

'This looks like a perfect plan!'

Embedded

Sarcasm is embedded in the meaning of words being used.

'I love being ignored'

Like-prefixed

'Like/As if' are common prefixes to ask rhetorical questions.

'Like you care'

Illocutionary

Non-speech acts (body language, gestures) contributing to the sarcasm

*'(shrugs shoulders)
Very helpful indeed!'*

Illocutionary sarcasm



Impact of Sarcasm on Sentiment Analysis (SA) (1/2)

Two SA systems:

MeaningCloud: <https://www.meaningcloud.com/>

NLTK (Bird, 2006)

Two datasets:

Sarcastic tweets by Riloff et al (2013)

Sarcastic utterances from our dataset of TV transcripts (Joshi et al 2016b)

Impact of Sarcasm on Sentiment Analysis (2/2)

	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

¹ www.meaningcloud.com

Clues for Sarcasm

- Use of laughter expression
 - *haha, you are very smart xD*
 - *Your intelligence astounds me. LOL*
- Heavy Punctuation
 - *Protein shake for dinner!! Great!!!*
- Use of emoticons
 - *i LOVE it when people tweet yet ignore my text X-(*
- Interjections
 - *3:00 am work YAY. YAY.*
- Capital Letters
 - *SUPER EXCITED TO WEAR MY UNIFORM TO SCHOOL TOMORROW !! :D lol.*

Incongruity: at the heart of things!

- *I love being ignored*
- *3:00 am work YAY. YAY.*
- *Up all night coughing. yeah me!*
- *No power, Yes! Yes! Thank you storm!*
- *This phone has an awesome battery back-up of 2 hour (Sarcastic)*

Two kinds of incongruity

- **Explicit incongruity**

- Overtly expressed through sentiment words of both polarities
- Contribute to almost 11% of sarcasm instances

'I love being ignored'

- **Implicit incongruity**

- Covertly expressed through phrases of implied sentiment

'I love this paper so much that I made a doggy bag out of it'

Sarcasm and Sense Ambiguity

(credit: Singamsetty Sandeep)

*Oh! Its so nice of you to give me a ring
early in the morning!*

Good to see you help dog bite victim!

Sarcasm Detection Using Semantic Incongruity

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya and Mark Carman, *Are Word Embedding-based Features Useful for Sarcasm Detection?*, **EMNLP 2016**, Austin, Texas, USA, November 1-5, 2016.

Also covered in: How Vector Space Mathematics Helps Machines Spot Sarcasm, MIT Technology Review, 13th October, 2016.

Feature Set

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

Datasets

Name	Text-form	Method of labeling	Statistics
Tweet-A	Tweets	Using sarcasm-based hashtags as labels	5208 total, 4170 sarcastic
Tweet-B	Tweets	Manually labeled (Given by Riloff et al(2013))	2278 total, 506 sarcastic
Discussion-A	Discussion forum posts (IAC Corpus)	Manually labeled (Given by Walker et al (2012))	1502 total, 752 sarcastic

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

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

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

Incongruity and embeddings

Capturing Incongruity Using Word Vectors

Use **similarity** of word embeddings

"A man needs a woman like a fish needs bicycle"

Word2Vec *similarity*(man, woman)= 0.766

Word2Vec *similarity*(fish, bicycle)= 0.131

Word embedding-based features

Unweighted similarity features (S):

Maximum score of most similar word pair

Minimum score of most similar word pair

Maximum score of most dissimilar word pair

Minimum score of most dissimilar word pair

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
- ✦ Labelled by users with tags
- ✦ Five-fold cross-validation
- ✦ Classifier: SVM-Perf optimised for F-score
- ✦ Configurations:
 - ✦ Four prior works (augmented with our sets of features)
 - ✦ Four implementations of word embeddings (Word2Vec, LSA, GloVe, Dependency weights-based)

Results (1/2)

Features	P	R	F
Baseline			
Unigrams	67.2	78.8	72.53
S	64.6	75.2	69.49
WS	67.6	51.2	58.26
Both	67	52.8	59.05

	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

Table 3: Performance obtained on augmenting word embedding features to features from four prior works, for four word embeddings; L: Liebrecht et al. (2013), G: González-Ibáñez et al. (2011a), B: Buschmeier et al. (2014), J: Joshi et al. (2015)

Results (2/2)

	Word2Vec	LSA	GloVe	Dep. Wt.
+S	0.835	0.86	0.918	0.978
+WS	1.411	0.255	0.192	1.372
+S+WS	1.182	0.24	0.845	0.795

Table 4: Average gain in F-Scores obtained by using intersection of the four word embeddings, for three word embedding feature-types, augmented to four prior works; Dep. Wt. indicates vectors learned from dependency-based weights

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

Table 5: 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

Numerical Sarcasm

Illustrates *need* for
Rule Based → Classical ML → Deep
Learning

Abhijeet Dubey, Lakshya Kumar, Arpan Somani, Aditya Joshi and Pushpak Bhattacharyya, ["When Numbers Matter!": Detecting Sarcasm in Numerical Portions of Text](#), 10th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (**WASSA 2019**), Minneapolis, USA, 7 June, 2019.

About 17% of sarcastic tweets have origin in number

1- This phone has an awesome battery back-up of 38 hours (Non-sarcastic)

2- This phone has a terrible battery back-up of 2 hours (Non-sarcastic)

3- This phone has an awesome battery back-up of 2 hour (Sarcastic)

Interesting question: why people use sarcasm?

- Dramatization, Forceful Articulation, lowering defence and then attack!

Numerical Sarcasm Dataset

Dataset-1	100000 (Sarcastic)	250000 (Non- Sarcastic)
Dataset-2	8681 (Num Sarcastic)	8681 (Non- Sarcastic)
Dataset-3	8681 (Num Sarcastic)	42107 (Non- Sarcastic)
Test Data	1843 (Num Sarcastic)	8317 (Non- Sarcastic)

- ✿ To create this dataset, we extract tweets from Twitter-API (<https://dev.twitter.com>).
- ✿ Hashtags of the tweets served as labels *#sarcasm #sarcastic* etc.
- ✿ Dataset-1 contains normal sarcastic + numeric sarcastic and non-sarcastic tweets.
- ✿ Rest all the other dataset contains numeric sarcastic and non-sarcastic tweets only.

Rule-based System (NP-Exact Matching) (Cont'd)

- ✦ Test Tweet: 'I love writing this paper at 9 am
- ✦ Matched Sarcastic Tweet: 'I love writing this paper daily at 3 am'
- ✦ 9 **NOT** close to 3

*test tweet is **non-sarcastic***

Example (sarcastic case)

- ✦ Test Tweet: 'I am so productive when my room is 81 degrees'
- ✦ Matched Non-sarcastic Tweet: 'I am very much productive in my room as it has 21 degrees'
- ✦ Absolute difference between 81 and 21 is high

Hence test tweet is

Sarcastic

Comparison of results (1: sarcastic, 0: non-sarcastic)

Approaches	Precision			Recall			F-score		
	P(1)	P(0)	P(avg)	R(1)	R(0)	R(avg)	F(1)	F(0)	F(avg)
Past Approaches									
Buschmeier et.al.	0.19	0.98	0.84	0.99	0.07	0.24	0.32	0.13	0.16
Liebrecht et.al.	0.19	1.00	0.85	1.00	0.07	0.24	0.32	0.13	0.17
Gonzalez et.al.	0.19	0.96	0.83	0.99	0.06	0.23	0.32	0.12	0.15
Joshi et.al.	0.20	1.00	0.86	1.00	0.13	0.29	0.33	0.23	0.25
Rule-Based Approaches									
Approach-1	0.53	0.87	0.81	0.39	0.92	0.83	0.45	0.90	0.82
Approach-2	0.44	0.85	0.78	0.28	0.92	0.81	0.34	0.89	0.79



Machine Learning based approach: classifiers and features

- ✿ SVM, KNN and Random Forest classifiers
- ✿ Sentiment-based features
 - ✿ Number of
 - ✿ positive words
 - ✿ negative words
 - ✿ highly emotional positive words,
 - ✿ highly emotional negative words.
- ✿ Positive/Negative word is said to be highly emotional if it's POS tag is one amongst : 'JJ', 'JJR', 'JJS', 'RB', 'RBR', 'RBS', 'VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'.

Emotion Features

- ✿ Positive emoticon
- ✿ Negative emoticon
- ✿ Boolean feature that will be one if both positive and negative words are present in the tweet.
- ✿ Boolean feature that will be one when either positive word and negative emoji is present or vice versa.

Punctuation features

- ✿ number of exclamation marks.
- ✿ number of dots
- ✿ number of question mark.
- ✿ number of capital letter words.
- ✿ number of single quotations.
- ✿ Number in the tweet: This feature is simply the number present in the tweet.
- ✿ Number unit in the tweet : This feature is a one hot representation of the type of unit present in the tweet. Example of number unit can be hour, minute, etc.

Comparison of results (1: sarcastic, 0: non-sarcastic)

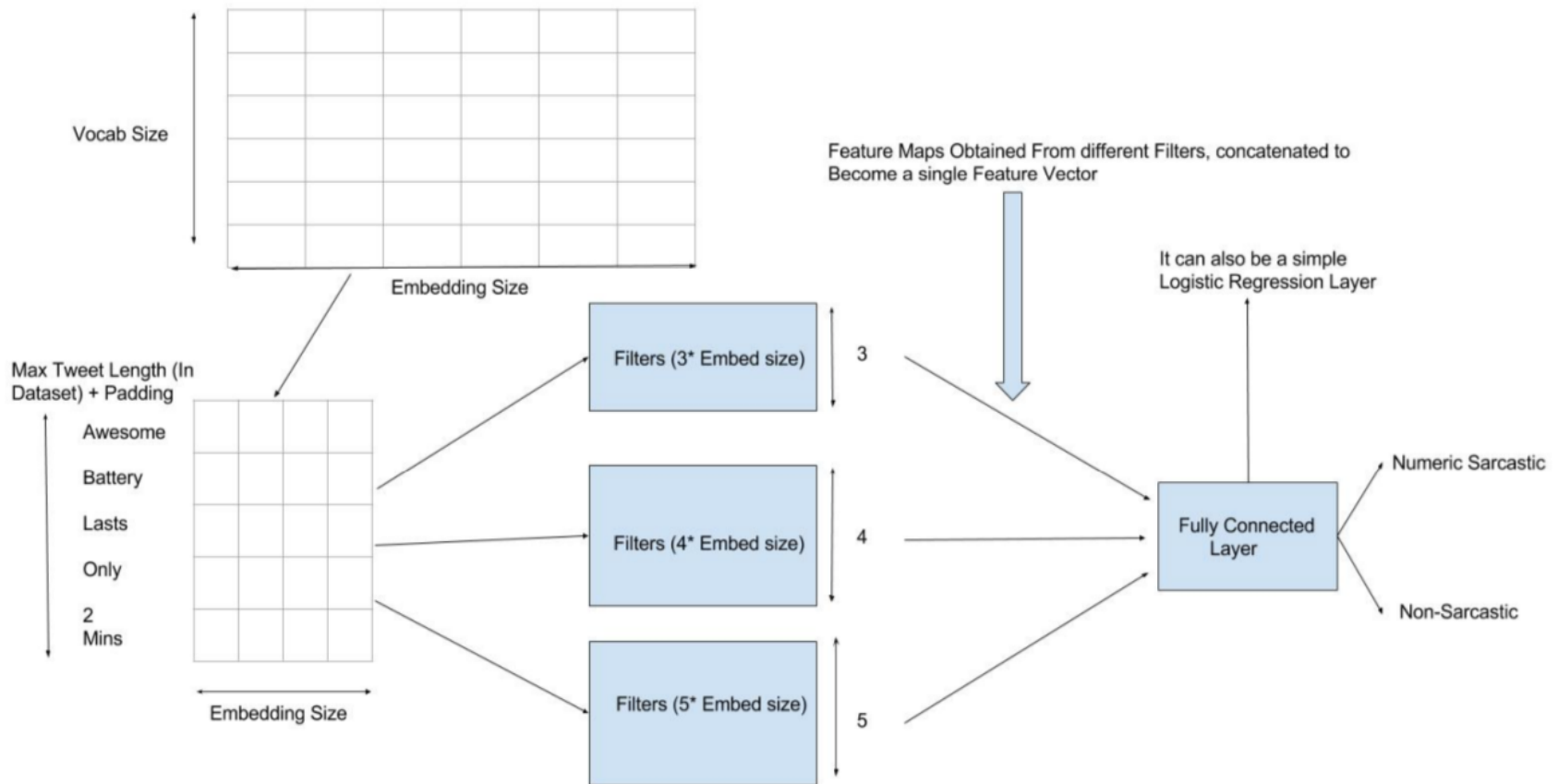
Approaches	Precision			Recall			F-score		
	P(1)	P(0)	P(avg)	R(1)	R(0)	R(avg)	F(1)	F(0)	F(avg)
Past Approaches									
Buschmeier et.al.	0.19	0.98	0.84	0.99	0.07	0.24	0.32	0.13	0.16
Liebrecht et.al.	0.19	1.00	0.85	1.00	0.07	0.24	0.32	0.13	0.17
Gonzalez et.al.	0.19	0.96	0.83	0.99	0.06	0.23	0.32	0.12	0.15
Joshi et.al.	0.20	1.00	0.86	1.00	0.13	0.29	0.33	0.23	0.25
Rule-Based Approaches									
Approach-1	0.53	0.87	0.81	0.39	0.92	0.83	0.45	0.90	0.82
Approach-2	0.44	0.85	0.78	0.28	0.92	0.81	0.34	0.89	0.79
Machine-Learning Based Approaches									
SVM	0.50	0.95	0.87	0.80	0.82	0.82	0.61	0.88	0.83
KNN	0.36	0.94	0.84	0.81	0.68	0.70	0.50	0.79	0.74
Random Forest	0.47	0.93	0.85	0.74	0.81	0.80	0.57	0.87	0.82



Deep Learning based

- ✿ Very little feature engg!!
- ✿ EmbeddingSize of 128
- ✿ Maximum tweet length 36 words
- ✿ Padding used
- ✿ Filters of size 3, 4, 5 used to extarct features

Deep Learning based approach: CNN-FF Model



Comparison of results (1: sarcastic, 0: non-sarcastic)

Approaches	Precision			Recall			F-score		
	P(1)	P(0)	P(avg)	R(1)	R(0)	R(avg)	F(1)	F(0)	F(avg)
Past Approaches									
Buschmeier et.al.	0.19	0.98	0.84	0.99	0.07	0.24	0.32	0.13	0.16
Liebrecht et.al.	0.19	1.00	0.85	1.00	0.07	0.24	0.32	0.13	0.17
Gonzalez et.al.	0.19	0.96	0.83	0.99	0.06	0.23	0.32	0.12	0.15
Joshi et.al.	0.20	1.00	0.86	1.00	0.13	0.29	0.33	0.23	0.25
Rule-Based Approaches									
Approach-1	0.53	0.87	0.81	0.39	0.92	0.83	0.45	0.90	0.82
Approach-2	0.44	0.85	0.78	0.28	0.92	0.81	0.34	0.89	0.79
Machine-Learning Based Approaches									
SVM	0.50	0.95	0.87	0.80	0.82	0.82	0.61	0.88	0.83
KNN	0.36	0.94	0.84	0.81	0.68	0.70	0.50	0.79	0.74
Random Forest	0.47	0.93	0.85	0.74	0.81	0.80	0.57	0.87	0.82
Deep-Learning Based Approaches									
CNN-FF	0.88	0.94	0.93	0.71	0.98	0.93	0.79	0.96	0.93
CNN-LSTM-FF	0.82	0.94	0.92	0.72	0.96	0.92	0.77	0.95	0.92
LSTM-FF	0.76	0.93	0.90	0.68	0.95	0.90	0.72	0.94	0.90

[back](#)

Context Incongruity

- Incongruity is defined as *'the state of being not in agreement, as with principles'*.
- Ivanko and Pexman (2003) state that the sarcasm processing time (time taken by humans to understand sarcasm) depends on the **degree of context incongruity** between the statement and the context.

Two kinds of incongruity

- **Explicit incongruity**

- Overtly expressed through sentiment words of both polarities
- Contribute to almost 11% of sarcasm instances

'I love being ignored'

- **Implicit incongruity**

- Covertly expressed through phrases of implied sentiment

'I love this paper so much that I made a doggy bag out of it'

Feature Set

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 (Based on Riloff et al)	
Implicit Sentiment Phrases	Boolean feature indicating phrases extracted from the implicit phrase extraction step
Explicit Incongruity (Based on Ramteke et al)	
#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

Datasets

Name	Text-form	Method of labeling	Statistics
Tweet-A	Tweets	Using sarcasm-based hashtags as labels	5208 total, 4170 sarcastic
Tweet-B	Tweets	Manually labeled (Given by Riloff et al(2013))	2278 total, 506 sarcastic
Discussion-A	Discussion forum posts (IAC Corpus)	Manually labeled (Given by Walker et al (2012))	1502 total, 752 sarcastic

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

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

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

Inter-sentential incongruity

- Incongruity may be expressed between sentences.
- We extend our classifier for Discussion-A by considering posts before the target post. These posts are 'elicitor posts'.
- Precision rises to 0.705 but the recall falls to 0.274.
 - Possible reason: Features become sparse since only 15% posts have elicitor posts

Introduce cognitive features

- Derive and augment cognitive features with traditional textual features.
- **Why?:** Textual nuances affect gaze (Just and Carpenter, 1979; Rayner, 1998)
- **Feasibility:** Inexpensive eye-tracking hardware available and integrated with handheld gadgets (e.g., <http://www.sencogi.com>)

Eye tracking

Saccades

Fixations



Eye Tracking Machines



Most comfortable technique to measure gaze based on infrared light



A bit more complicated way to measure gaze using electric potential around the eye.



The eye tracking glasses are used for broad range of mobile eye tracking studies.



The ergonomic chin rest eye tracking device for high speed and accurate measurements with a large visual field.

Eye tracking on mobile phones

- Samsung Galaxy S4 comes with eye tracking capability
- The software [umooove](http://www.umooove.me/) (<http://www.umooove.me/>) runs on mobile phones, tracking eyes
- **MIT Technology Review, June 2015:**
 - “Eye-tracking system uses ordinary cellphone camera”

Eye Tracking: basic parameters

- **Gaze points:**
 - Position of eye-gaze on the screen
- **Fixations:**
 - A long stay of the gaze on a particular object on the screen. Fixations have both Spatial (coordinates) and Temporal (duration) properties.
- **Saccade:**
 - A very rapid movement of eye between the positions of rest.
- **Scanpath:**
 - A path connecting a series of fixations.
- **Regression:**
 - Revisiting a previously read segment

Use of eye tracking

- Used extensively in Psychology
 - Mainly to study reading processes
 - Seminal work: Just, M.A. and Carpenter, P.A. (1980). A theory of reading: from eye fixations to comprehension. *Psychological Review* 87(4):329–354
- Used in flight simulators for pilot training
- Website developers use eye tracking to improve look and feel of websites

Eye tracking usage

Top 8 Applications in Eye Tracking

Human Factors and Simulation

Market Research

Usability Research

Packaging Research



Academic and Scientific Research

Medical Research

Psychology Research

PC and Gaming Research

Our contribution:

(a) Better measures of **Readability**

(b) Use of eye tracking in NLP- **cognitive NLP**

NLP-ML and Eye Tracking

- Kliegl (2011)- Predict word frequency and pattern from eye movements
- Doherty et. al (2010)- Eye-tracking as an automatic Machine Translation Evaluation Technique
- Stymne et al. (2012)- Eye-tracking as a tool for Machine Translation (MT) error analysis
- Dragsted (2010)- Co-ordination of reading and writing process during translation.

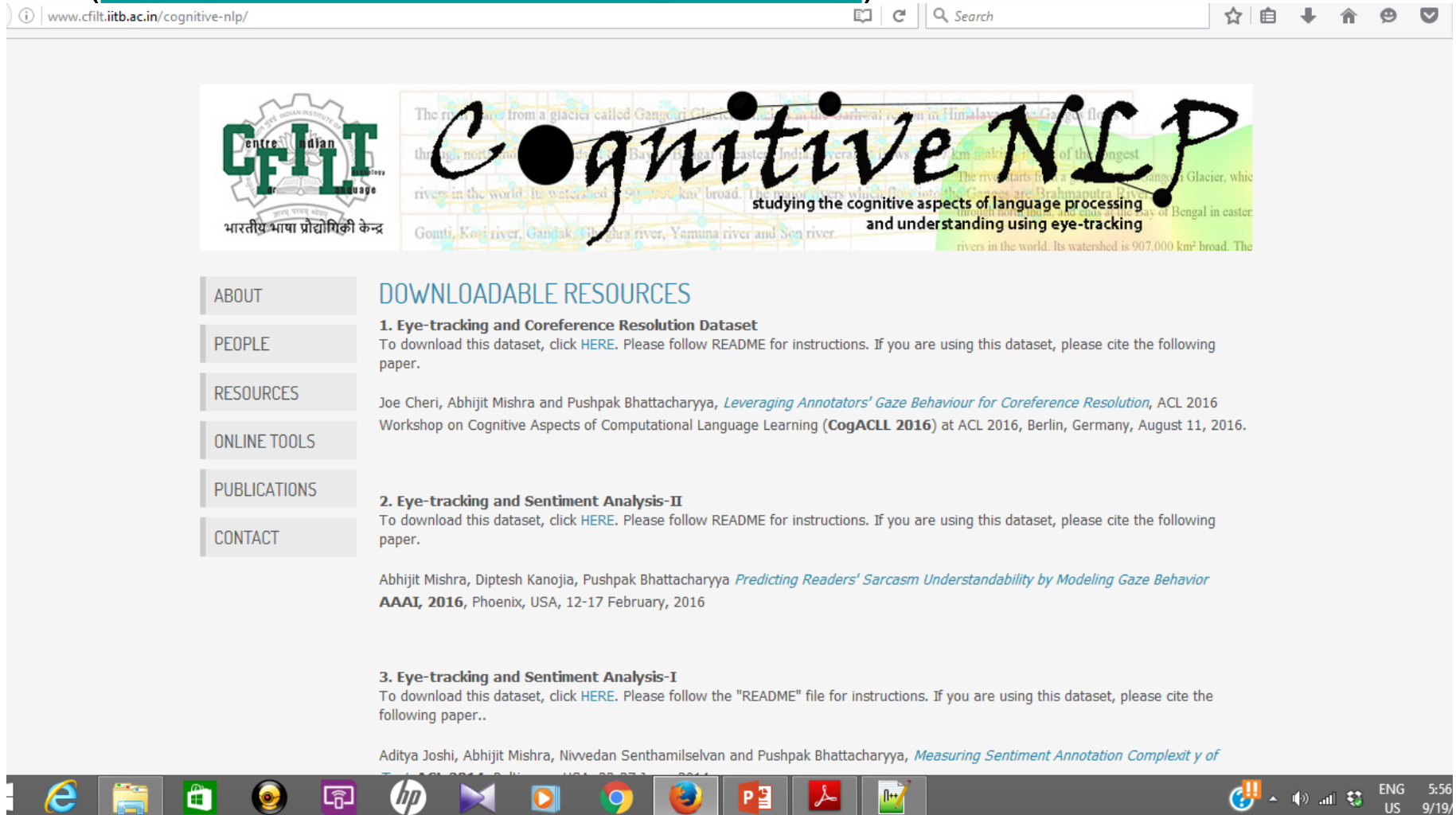
Relatively new and open research direction

Our lab (CFILT@IITB) has been Contributing

- ✦ Joshi, Aditya and Mishra, Abhijit and S., Nivvedan and Bhattacharyya, Pushpak. 2014. Measuring Sentiment Annotation Complexity of Text. Association for Computational Linguistics, **(ACL 2014)** Baltimore, USA.
- ✦ Mishra, Abhijit and Bhattacharyya, Pushpak and Carl, Michael. 2013. Automatically Predicting Sentence Translation Difficulty. Association for Computational Linguistics **(ACL 2013)**, Sofia, Bulgaria

Contribution to NLP Community

Publicly available datasets and tools
(<http://www.cfilt.iitb.ac.in/cognitive-nlp>)

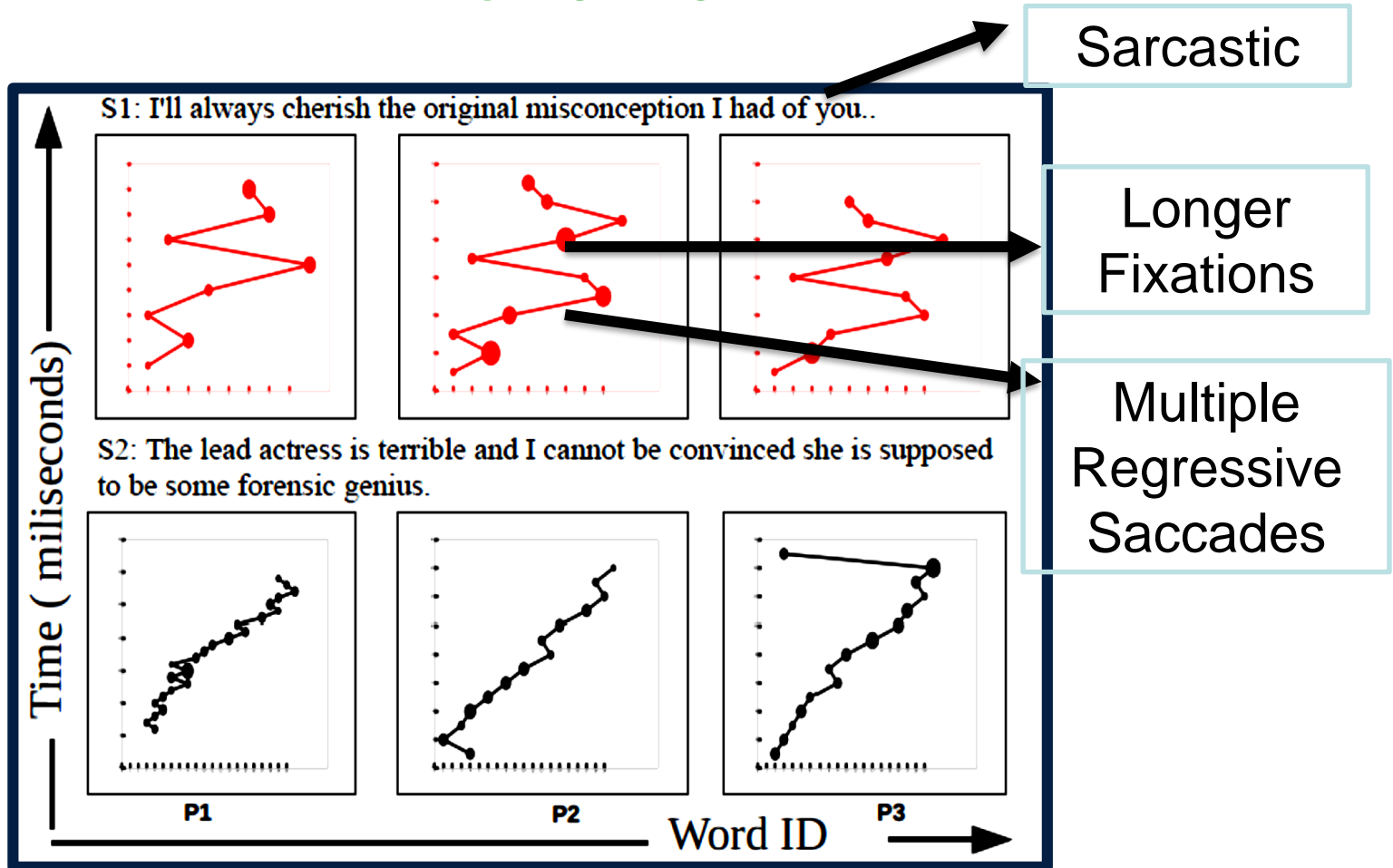


The screenshot shows a web browser displaying the website for the Cognitive NLP group at IIT Bombay. The browser's address bar shows the URL www.cfilt.iitb.ac.in/cognitive-nlp. The website header features the CFILT logo and the text "Centre for Cognitive and Language Studies" and "भारतीय भाषा प्रौद्योगिकी केन्द्र". The main banner displays "Cognitive NLP" in a large, stylized font, with the subtitle "studying the cognitive aspects of language processing and understanding using eye-tracking". Below the banner is a navigation menu with links for ABOUT, PEOPLE, RESOURCES, ONLINE TOOLS, PUBLICATIONS, and CONTACT. The main content area is titled "DOWNLOADABLE RESOURCES" and lists three datasets:

- 1. Eye-tracking and Coreference Resolution Dataset**
To download this dataset, click [HERE](#). Please follow README for instructions. If you are using this dataset, please cite the following paper.
Joe Cheri, Abhijit Mishra and Pushpak Bhattacharyya, *Leveraging Annotators' Gaze Behaviour for Coreference Resolution*, ACL 2016 Workshop on Cognitive Aspects of Computational Language Learning (**CogACLL 2016**) at ACL 2016, Berlin, Germany, August 11, 2016.
- 2. Eye-tracking and Sentiment Analysis-II**
To download this dataset, click [HERE](#). Please follow README for instructions. If you are using this dataset, please cite the following paper.
Abhijit Mishra, Diptesh Kanojia, Pushpak Bhattacharyya *Predicting Readers' Sarcasm Understandability by Modeling Gaze Behavior* **AAAI, 2016**, Phoenix, USA, 12-17 February, 2016
- 3. Eye-tracking and Sentiment Analysis-I**
To download this dataset, click [HERE](#). Please follow the "README" file for instructions. If you are using this dataset, please cite the following paper..
Aditya Joshi, Abhijit Mishra, Nivvedan Senthamilselvan and Pushpak Bhattacharyya, *Measuring Sentiment Annotation Complexity of*

The Windows taskbar at the bottom shows various application icons including Edge, File Explorer, Store, Camera, Network, HP, VLC, Chrome, Firefox, PowerPoint, PDF Reader, and a system tray with volume, network, and system icons. The system clock shows 5:56 on 9/19.

Sentiment Annotation and Eye Movement



Datasets

- ✦ Two publicly available datasets released by us
(Mishra et al, 2016; Mishra et al., 2014)
- ✦ **Dataset 1: (Eye-tracker: Eyelink-1000 Plus)**
 - ✦ 994 text snippets : 383 positive and 611 negative, 350 are sarcastic/ironic
 - ✦ Mixture of Movie reviews, Tweets and sarcastic/ironic quotes
 - ✦ Annotated by 7 human annotators
 - ✦ Annotation accuracy: **70%-90%** with Fleiss kappa IAA of **0.62**
- ✦ **Dataset 2: (Eye-tracker: Tobi TX300)**
 - ✦ 843 snippets : 443 positive and 400 negative
 - ✦ Annotated by 5 human subjects
 - ✦ Annotation accuracy: **75%-85%** with Fleiss kappa IAA of **0.68**

Accuracy of Traditional Classifiers on our Datasets

- ✦ Trained Naïve Bayes and SVM using 10662 short text and traditional features (**Liu and Zhang, 2012**)
- ✦ Classifiers tried: Naïve Bayes, SVM and Rule Based
- ✦ Tested using both of our datasets.

	NB			SVM			RB		
	P	R	F	P	R	F	P	R	F
D1	66.15	66	66.15	64.5	65.3	64.9	56.8	60.9	53.5
D2	74.5	74.2	74.3	77.1	76.5	76.8	75.9	53.9	63.02

- ✦ Lower accuracy indicates higher difficulty

Features for SA (Textual)

Presence of Unigrams (NGRAM_PCA)
Count of Subjective Words (Positive_words, Negative_words)
Subjective Score from SentiWordNet (PosScore, NegScore)
Sentiment Flip Count (FLIP)
Part of Speech Ratios (VERB, NOUN, ADJ, ADV)
Count of Named Entities (NE)
Count of Discourse Connectors – <i>e.g., however, although</i> (DC)

Features for SA (Textual)

- **Sarcasm, Irony and Thwarting related Features (Joshi et al, 2015; Ramteke et al. 2013)**

Presence of Implicitly Incongruous Phrases – Riloff et al. (IMP)
Longest pos/neg subsequence (LAR)
Resultant Lexical Word Polarity of Text (LP)
Punctuations and Injections (PUNC)

- **Features related to reading difficulty**

Flesch Readability Ease (RED)
Total word count (LEN)
Average syllable per word (SYL)

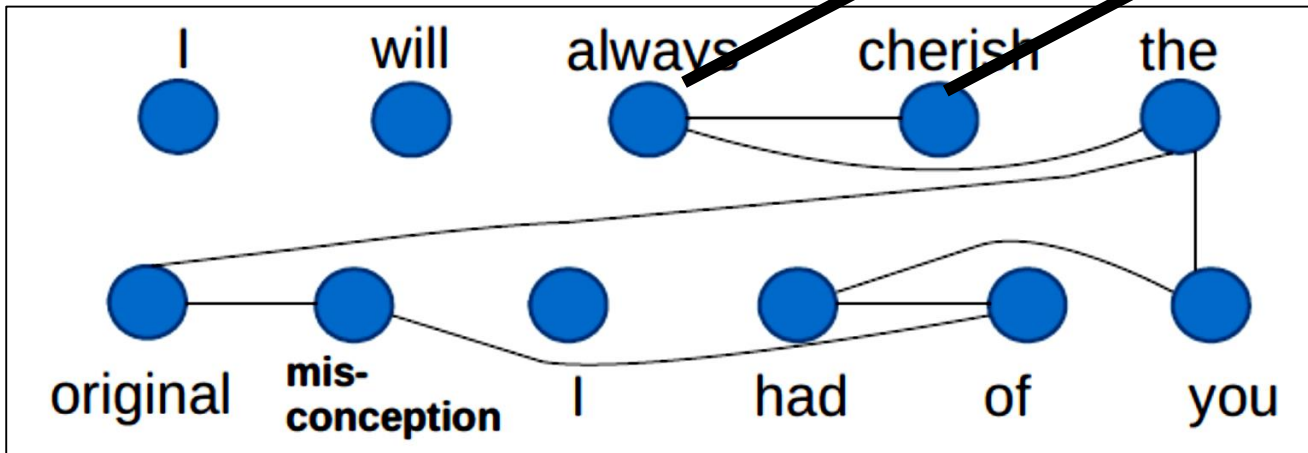
Features for SA (Cognitive)

- Simple Features from Eye-movement (extracted directly from recorded eye-movement data)

Average First Fixation Duration per Word (FDUR)
Average Fixation Count (FC)
Average Saccade Length (SL)
Total Regressive Saccade Count (REG)
Count of Number of Words Skipped (SKIP)
Count of Regressive Saccades from Second Half to First Half of the Text (RSF)
Position of the word from which the largest regression starts (LREG)

Features for SA (Cognitive)

Complex Gaze Features derived from Gaze saliency graph



Features for SA (Cognitive)

◆ Features from the Gaze Saliency Graph

Edge Density (ED) of the Gaze Saliency Graph

Highest , 2nd Highest Weighted Degree With **Fixation Duration** at Source Node, Target Node as Edge Weight (F1H, F1S, F2H, F2S)

Highest , 2nd Highest Weighted Degree With **Forward Saccade Count** as Edge Weight (FSH, FSS)

Highest , 2nd Highest Weighted Degree With **Forward Saccade Distance** as Edge Weight (FSDH, FSDS)

Highest , 2nd Highest Weighted Degree With **Reverse Saccade Count** as Edge Weight (RSH, RSS)

Highest , 2nd Highest Weighted Degree With **Reverse Saccade Distance** as Edge Weight (RSDH, RSDS)

Why these Gaze features?

- **Key observation from dataset:** Negative sentiment bearing texts are more linguistically subtle (irony, sarcasm, implicit-sentiment)
- **Why simple gaze features?:** Significant variation in gaze attributes (fixation duration, regression count, skip count) and observed when text has such subtleties (observed through t-tests). So, our simple gaze features contain important information regarding subtleties.
- **Why complex gaze features?:** When the text has distinct phrases pointing to situational disparities (like incongruity in sarcasm), a lot of regressive saccades around these phases observed, making the gaze saliency graph **Dense** (Captured by Edge Density) and modular (with a few nodes having very large degrees).

Experiment

- **Sentiment Polarity prediction of Snippets** : Binary Classification Problem
- **Classifiers**: Naïve Bayes, Support Vector Machine (With Linear Kernel), Multi-layered Perceptron
- **Evaluation Mode**: 10-Fold Cross validation
- **Feature Combination**
 - Unigram Only (Uni)
 - Sentiment [Includes Unigram Presence] (Sn)
 - Sarcasm, Irony and Thwarting Features [Include Unigram Presence](Sr)
 - Gaze and readability (Gz)

Results

$p=0.006$

$p = 2e-5$

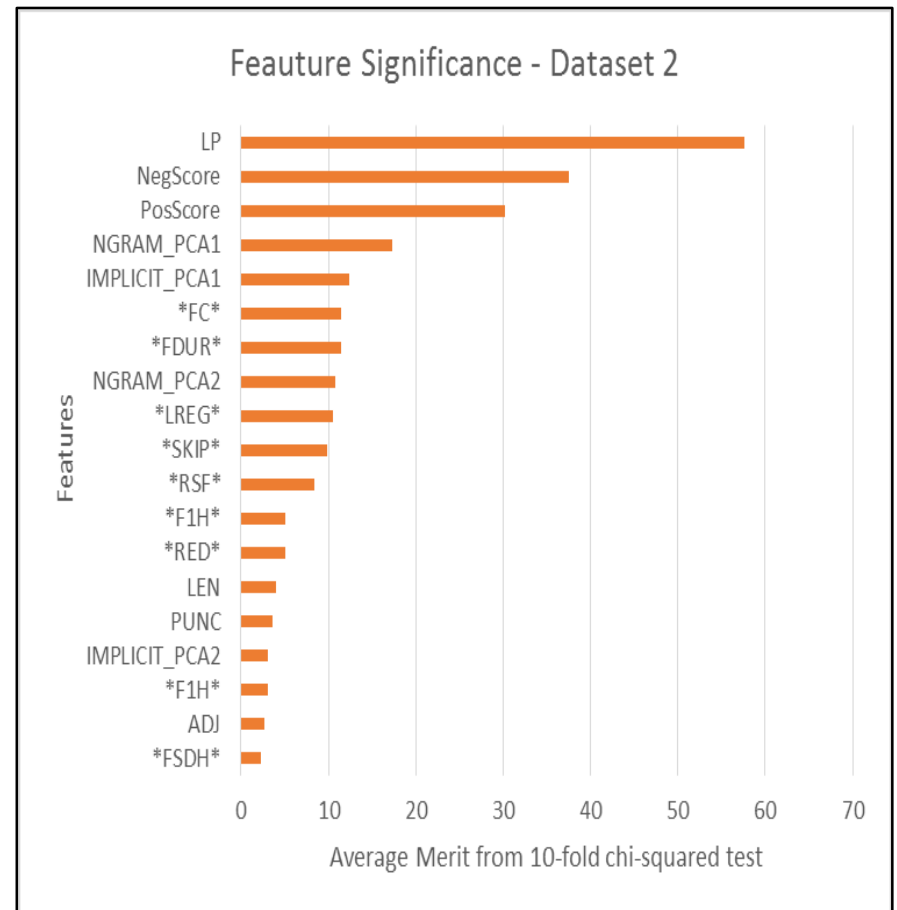
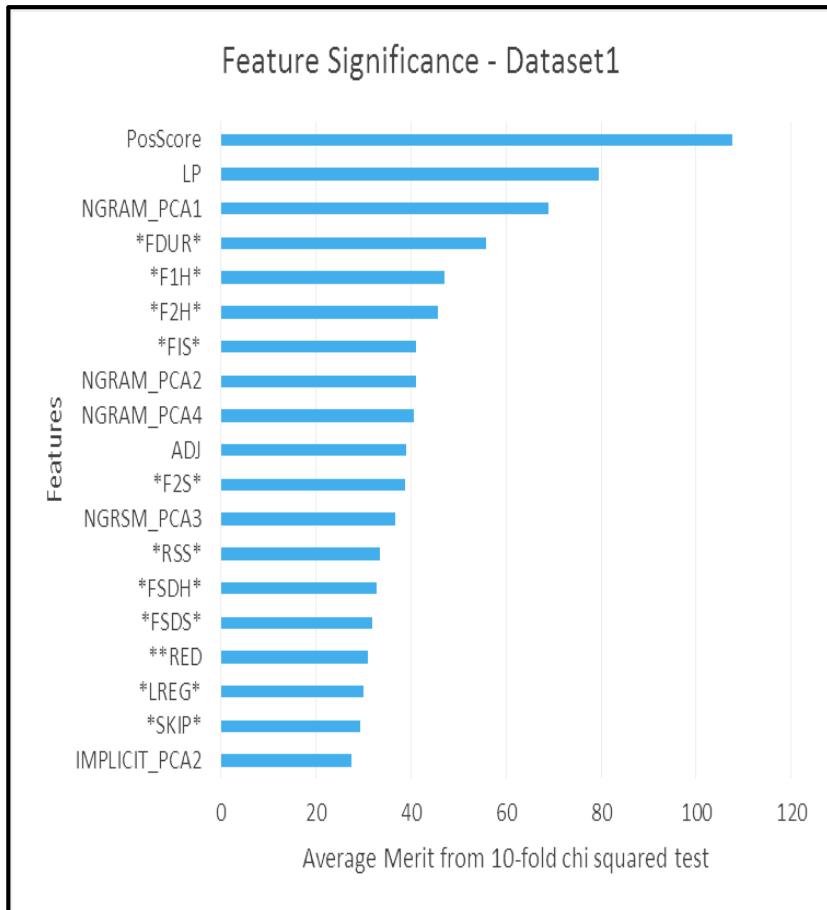
Classifier	Näive Bayes			SVM			Multi-layer NN		
	P	R	F	P	R	F	P	R	F
Dataset 1									
Uni	58.5	57.3	57.9	67.8	68.5	68.14	65.4	65.3	65.34
Sn	58.7	57.4	58.0	69.6	70.2	69.8	67.5	67.4	67.5
Sn + Sr	63.0	59.4	61.14	72.8	73.2	72.6	69.0	69.2	69.1
Gz	61.8	58.4	60.05	54.3	52.6	53.4	59.1	60.8	60
Sn+Gz	60.2	58.8	59.2	69.5	70.1	69.6	70.3	70.5	70.4
Sn+ Sr+Gz	63.4	59.6	61.4	73.3	73.6	73.5	70.5	70.7	70.6
Dataset 2									
Uni	51.2	50.3	50.74	57.8	57.9	57.8	53.8	53.9	53.8
Sn	51.1	50.3	50.7	62.5	62.5	62.5	58.0	58.1	58.0
Sn+Sr	50.7	50.1	50.39	70.3	70.3	70.3	66.8	66.8	66.8
Gz	49.9	50.9	50.39	48.9	48.9	48.9	53.6	54.0	53.3
Sn+Gz	51	50.3	50.6	62.4	62.3	62.3	59.7	59.8	59.8
Sn+ Sr+Gz	50.2	49.7	50	71.9	71.8	71.8	69.1	69.2	69.1

$p = 0.0003,$

$p = 0.21$

How good are Cognitive Features?

– Chi squared test



**Ablation test: No significant differences observed by ablating one feature at a time*

How good are Cognitive Features?- Heldout accuracy

- Dataset-1 split into a train-test split of 760:234 (Out of 234, 131 contain irony/sarcasm)
- We checked how our best performing classifier with different feature combinations perform for both Irony and Non-irony parts.

	Irony	Non-Irony
Sn	58.2	75.5
Sn+Sr	60.1	75.9
Gz+Sn+Sr	64.3	77.6

$p = 0.001$

F-scores on texts containing Sarcasm/Irony in Held-out Dataset derived from dataset-1 (Train-test split of 760:234)

Example Sentences

Sentence	Gold	SVM_Ex.	NB_Ex.	RB_Ex.	Sn	Sn+Sr	Sn+Sr+Gz
1. I find television very educating. Every time somebody turns on the set, I go into the other room and read a book	-1	1	1	0	1	-1	-1
2. I love when you do not have two minutes to text me back.	-1	1	-1	1	1	1	-1

Discussions: Augmented features for Sarcasm *Help!*

Textual

- (1) Unigrams (2) Punctuations
- (3) Implicit incongruity
- (4) Explicit Incongruity
- (5) Largest +ve/-ve subsequences
- (6) +ve/-ve word count
- (7) Lexical Polarity
- (8) Flesch Readability Ease,
- (9) Word count

Simple gaze

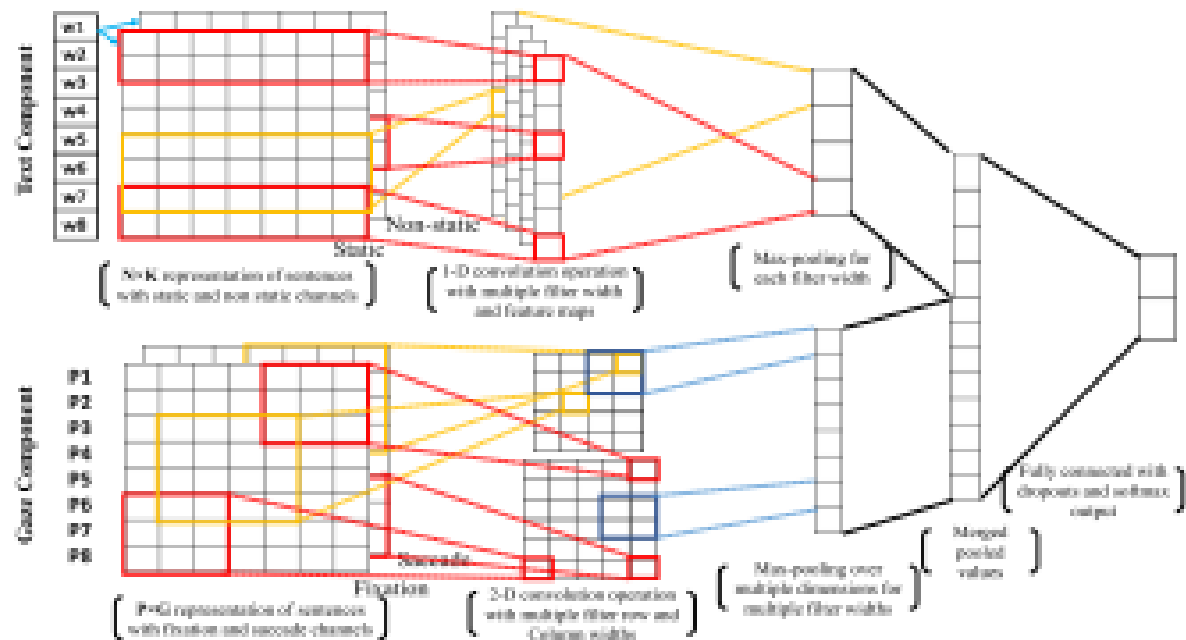
- (1) Average Fixation Duration,
- (2) Average Fixation Count,
- (3) Average Saccade Length,
- (4) Regression Count,
- (5) Number of words skipped,
- (6) Regressions from second half to first half,
- (7) Position of the word from which the largest regression starts

Complex gaze

- (1) Edge density,
- (2) Highest weighted degree
- (3) Second Highest weighted degree
(With different edge-weights)

CNN Based Sarcasm Detection

Abhijit Mishra, Kuntal Dey and Pushpak Bhattacharyya, [Learning Cognitive Features from Gaze Data for Sentiment and Sarcasm Classification Using Convolutional Neural Network](#), **ACL 2017**, Vancouver, Canada, July 30-August 4, 2017.



Learning Cognitive Features from Gaze Data for Sentiment and Sarcasm Classification

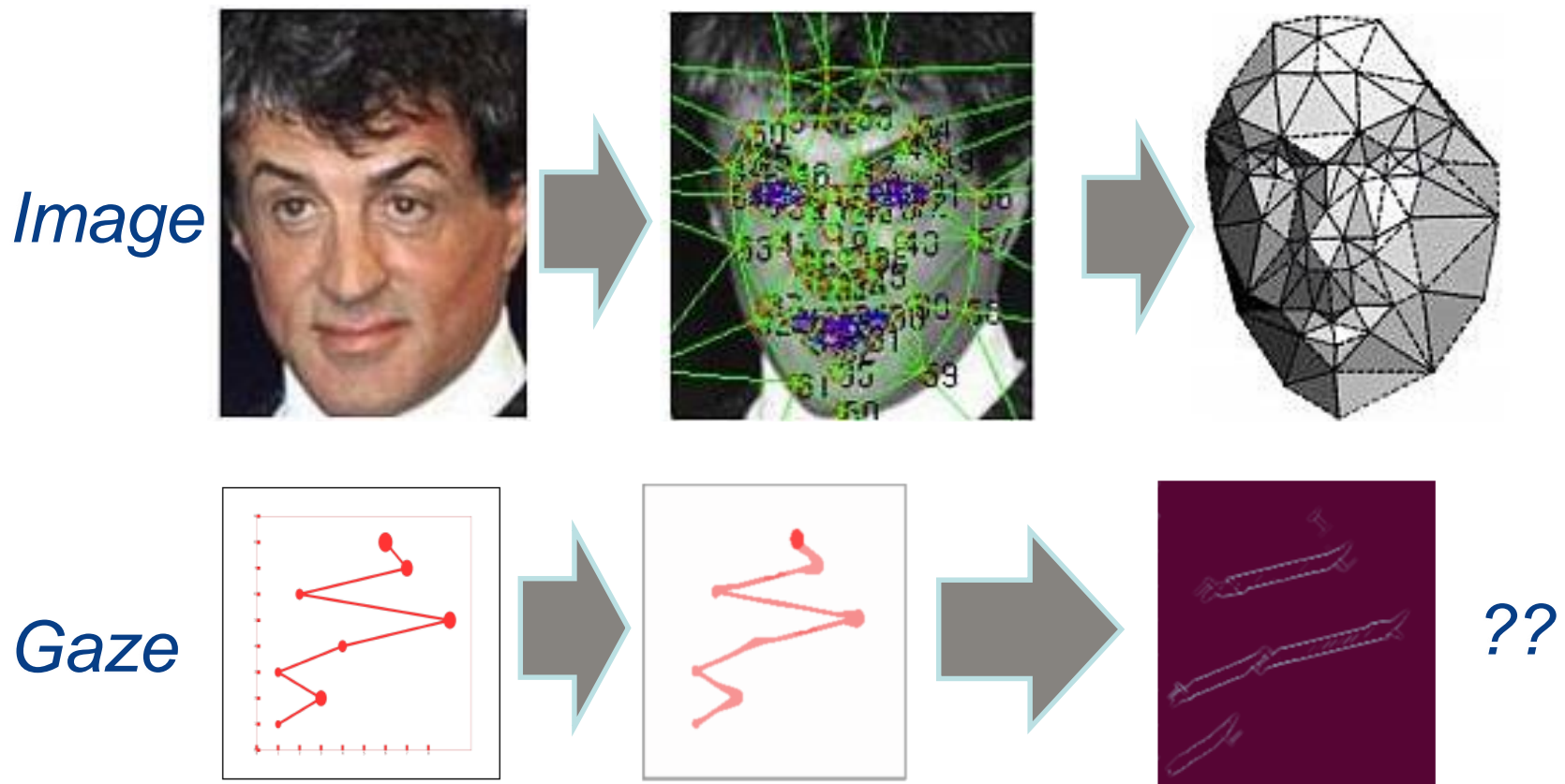
- In complex classification tasks like sentiment analysis and sarcasm detection, even the extraction and choice of features should be delegated to the learning system
- The idea of channels in CNN is exploited, and CNN learns features from both gaze and text and uses them to classify the input text

Central Idea

- Learn features from Gaze sequences (fixation duration sequences and gaze-positions) and Text automatically using Deep Neural Networks.
- Deep NNs have proven to be good at learning feature representations for Image and Text classification tasks (Krizhevsky et al., 2012; Collobert et al., 2011).
- Use Convolutional Neural Network (already used for sentiment classification, Kim, 2014)

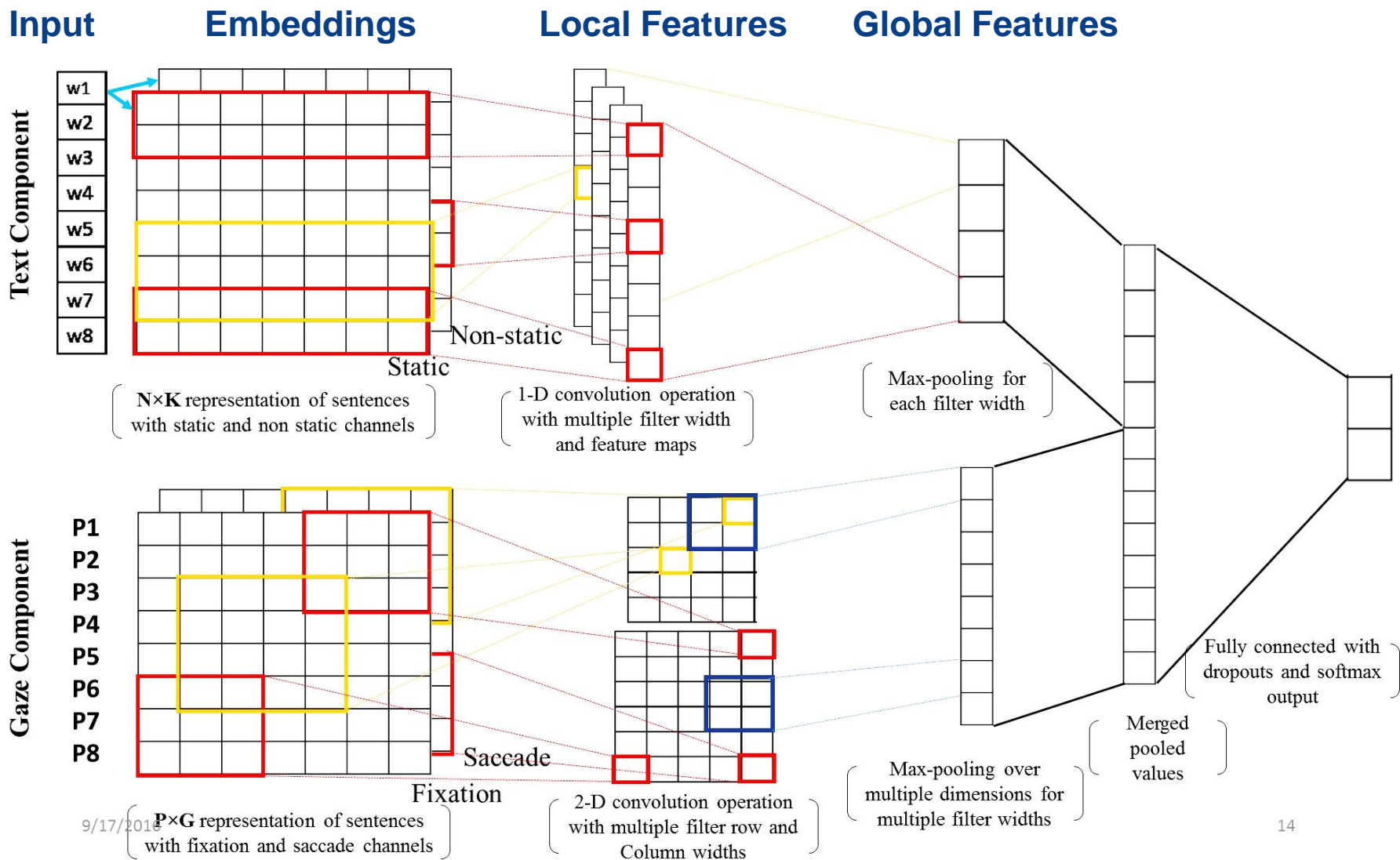
Why Convolutional NNs

- Convolutional Layers good at capturing compositionality (Lawrence et al, 1997).



Images taken from: mrulafi.blogspot.com

Neural Network Architecture



Why both Static and Non-static embedding

- Non-static embedding channel for tuning embeddings for SA/Sarcasm (e.g., produce similar embeddings for adjectives like good and excellent)
- Static embedding channel: to prevent over-tuning of embeddings due to collocation (e.g., words such as I and love are often collocated but should not share similar vector representation).

Fixation and Saccade Channels

- Fixation channel: Lexical Complexity (pertaining to length, frequency and predictability of words while annotation)
- Saccade channel: Syntactic Complexity and Incongruity

Datasets (1/2)

- Two publicly available datasets released by us (Mishra et al, 2016; Mishra et al., 2014)
- Dataset 1: (Eye-tracker: Eyelink-1000 Plus)
 - 994 text snippets : 383 positive and 611 negative, 350 are sarcastic/ironic
 - Mixture of Movie reviews, Tweets and sarcastic/ironic quotes
 - Annotated by 7 human annotators
 - Annotation accuracy: 70%-90% with Fleiss kappa IAA of 0.62

Datasets (2/2)

- Dataset 2: (Eye-tracker: Tobii TX300)
- 843 snippets : 443 positive and 400 negative
- Annotated by 5 human subjects
- Annotation accuracy: 75%-85%with Fleiss kappa IAA of 0.68

Experimental Setup: Configurations

- Text Only: (Only Text Component is Used)
 - Text_Static: Word embeddings are kept static and not updated during back propagation.
 - Text_Non-static: Embeddings are updated during back propagation.
 - Text_Multi Channel: Two channels (one taking input from static and one from dynamic embeddings) are used.
- Gaze Only: (Only Gaze Component is Used)
 - Gaze_Fixation_Duration: Sequence of fixation durations are used as input
 - Gaze_Saccade: Sequence of gaze locations (in terms of word ID used as input)
 - Gaze-Multi Channel: Two channels (one taking input from Fixation and one from saccade) are used
- Both text and Gaze (9-Configs)

Experiment Setup (Model Details)

- Word Embeddings: Word2Vec (Mikolov et.al), trained on Amazon Movie Review Data, Embedding dimensions: 300
- Convolution: Filter sizes: 3,4 (Best), Number of filters used for each filter size: 150 (Better than smaller values)
- Feed-Forward: Number of hidden neurons: 150 (Better than smaller values), Dropout probability: 0.25
- Training: Number of epochs: 200 (change in loss negligible after 200 epochs), Optimizer: Adadelta, LR: 0.1

Results – Sentiment Analysis

Configuration		Dataset1			Dataset2		
		P	R	F	P	R	F
Traditional systems based on textual features	Näive Bayes	63.0	59.4	61.14	50.7	50.1	50.39
	Multi-layered Perceptron	69.0	69.2	69.2	66.8	66.8	66.8
	SVM (Linear Kernel)	72.8	73.2	72.6	70.3	70.3	70.3
Systems by Mishra et al. (2016c)	Gaze based (Best)	61.8	58.4	60.05	53.6	54.0	53.3
	Text + Gaze (Best)	73.3	73.6	73.5	71.9	71.8	71.8
CNN with only text input (Kim, 2014)	STATICTEXT	63.85	61.26	62.22	55.46	55.02	55.24
	NONSTATICTEXT	72.78	71.93	72.35	60.51	59.79	60.14
	MULTICHANNELTEXT	72.17	70.91	71.53	60.51	59.66	60.08
CNN with only gaze Input	FIXATION	60.79	58.34	59.54	53.95	50.29	52.06
	SACCADE	64.19	60.56	62.32	51.6	50.65	51.12
	MULTICHANNELGAZE	65.2	60.35	62.68	52.52	51.49	52
CNN with both text and gaze Input	STATICTEXT + FIXATION	61.52	60.86	61.19	54.61	54.32	54.46
	STATICTEXT + SACCADE	65.99	63.49	64.71	58.39	56.09	57.21
	STATICTEXT + MULTICHANNELGAZE	65.79	62.89	64.31	58.19	55.39	56.75
	NONSTATICTEXT + FIXATION	73.01	70.81	71.9	61.45	59.78	60.60
	NONSTATICTEXT + SACCADE	77.56	73.34	75.4	65.13	61.08	63.04
	NONSTATICTEXT + MULTICHANNELGAZE	79.89	74.86	77.3	63.93	60.13	62
	MULTICHANNELTEXT + FIXATION	74.44	72.31	73.36	60.72	58.47	59.57
	MULTICHANNELTEXT + SACCADE	78.75	73.94	76.26	63.7	60.47	62.04
MULTICHANNELTEXT + MULTICHANNELGAZE	78.38	74.23	76.24	64.29	61.08	62.64	

Results – Sarcasm Detection

	Configuration	P	R	F
Traditional systems based on textual features	Näive Bayes	69.1	60.1	60.5
	Multi-layered Perceptron	69.7	70.4	69.9
	SVM (Linear Kernel)	72.1	71.9	72
Systems by Riloff et al. (2013)	Text based (Ordered)	49	46	47
	Text + Gaze (Unordered)	46	41	42
System by Joshi et al. (2015)	Text based (best)	70.7	69.8	64.2
Systems by Mishra et al. (2016b)	Gaze based (Best)	73	73.8	73.1
	Text based (Best)	72.1	71.9	72
	Text + Gaze (Best)	76.5	75.3	75.7
CNN with only text input (Kim, 2014)	STATICTEXT	67.17	66.38	66.77
	NONSTATICTEXT	84.19	87.03	85.59
	MULTICHANNELTEXT	84.28	87.03	85.63
CNN with only gaze input	FIXATION	74.39	69.62	71.93
	SACCADE	68.58	68.23	68.40
	MULTICHANNELGAZE	67.93	67.72	67.82
CNN with both text and gaze Input	STATICTEXT + FIXATION	72.38	71.93	72.15
	STATICTEXT + SACCADE	73.12	72.14	72.63
	STATICTEXT + MULTICHANNELGAZE	71.41	71.03	71.22
	NONSTATICTEXT + FIXATION	87.42	85.2	86.30
	NONSTATICTEXT + SACCADE	84.84	82.68	83.75
	NONSTATICTEXT + MULTICHANNELGAZE	84.98	82.79	83.87
	MULTICHANNELTEXT + FIXATION	87.03	86.92	86.97
	MULTICHANNELTEXT + SACCADE	81.98	81.08	81.53
	MULTICHANNELTEXT + MULTICHANNELGAZE	83.11	81.69	82.39

Observations (1/2)

- Overfitting for SA dataset 2: Training accuracy reaches 100 within 25 epochs with validation accuracy still at around 50%. Better dropout/regularization configuration required.
- Better classification accuracy for Sarcasm detection: Clear differences between vocabulary of sarcasm and non-sarcasm classes in our dataset. Captured well by non-static embeddings.
- Effect of dimension variation: Reducing embedding dimension improves by a little margin.

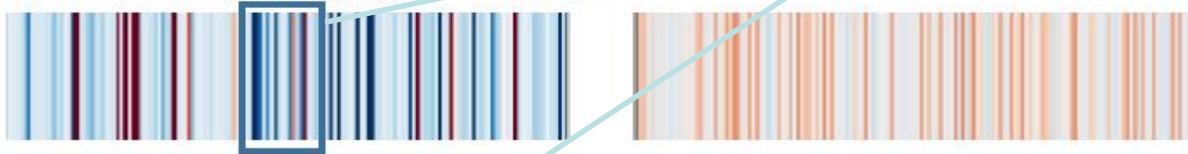
Observations (2/2)

- Increasing filters beyond 180 decreases accuracy (possibly over-fits). Decreasing beyond 30 decreases accuracy.
- Effect of static / non static text channels: Better for non static (word embeddings with similar sentiment come closer in non static channels, e.g., *good* ~ *nice*)
- Effect of fixation / saccade channels: Saccade channel alone handles nuances like incongruity better.
- Fixation channel does not help much, may be because of higher variance in fixation duration.

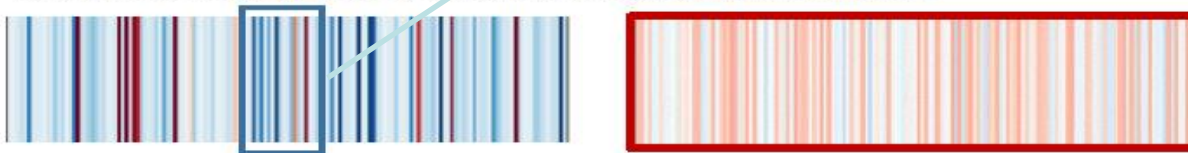
Analysis of Features Learned (1/2)

Capturing intensity variation in sarcasm VS no-sarcasm better

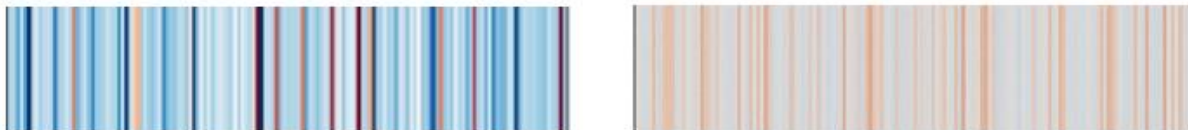
1. I would like to live in Manchester, England. The transition between Manchester and death would be unnoticeable. (*Sarcastic, Negative Sentiment*)



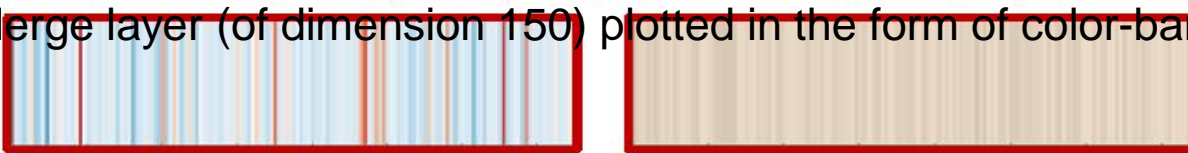
2. We really did not like this camp. After a disappointing summer, we switched to another camp, and all of us much happier on all fronts! (*Non Sarcastic, Negative Sentiment*)



3. Helped me a lot with my panics attack I take 6 mg a day for almost 20 years can't stop of course but make me feel very comfortable (*Non Sarcastic, Positive Sentiment*)



4. Howard is the King and always will be, all others are weak clones. (*Non Sarcastic, Positive Sentiment*)



(a) MultichannelText + MultichannelGaze

(b) MultichannelText

Visualization of representations learned for Sarcasm Detection. Output of the Merge layer (of dimension 150) plotted in the form of color-bars (Li et al. , 2016)

Analysis of Features Learned (2/2)

- Addition of gaze information helps to generate features with more subtle differences.
- Features for the sarcastic texts exhibit more intensity than the non-sarcastic ones- perhaps capturing the notion that sarcasm typically conveys an intensified negative opinion.
- Example 4 is incorrectly classified by both the systems— lack of context?
- Addition of gaze information does not help here, as it becomes difficult for even humans to classify such texts