THE BEST SARCASM DETECTOR EVER CREATED

ESHA WANG

OUTLINE

- Background and motivation
- Data: IAC dialogue, Reddit, Twitter, *The Onion* and *HuffPost* headlines
- Data cleaning and pre-processing
- Baseline models: Unigrams and NBOW
- Proposed model: 2CUE-CNN with generalized user embeddings
- Future Work
- Additional slides:
  - Human activity recognition using Wi-Fi CSI data
  - Facial expression recognition

BACKGROUND

Sarcasm /ˈsärˌkazəm/: A commonly used, and often misinterpreted, communicative device, in which speakers say something other than, and usually opposite to, what they actually mean.

- **General sarcasm**:

  “Wow, life is so difficult for my spoiled housecat.”

- **Hyperbole**:

  “My cat acts as though she hasn’t eaten in years.”

- **Veiled insult:**

  “We can always count on Americans to do the right thing,
  after they have exhausted all other possibilities.”

  – Winston Churchill

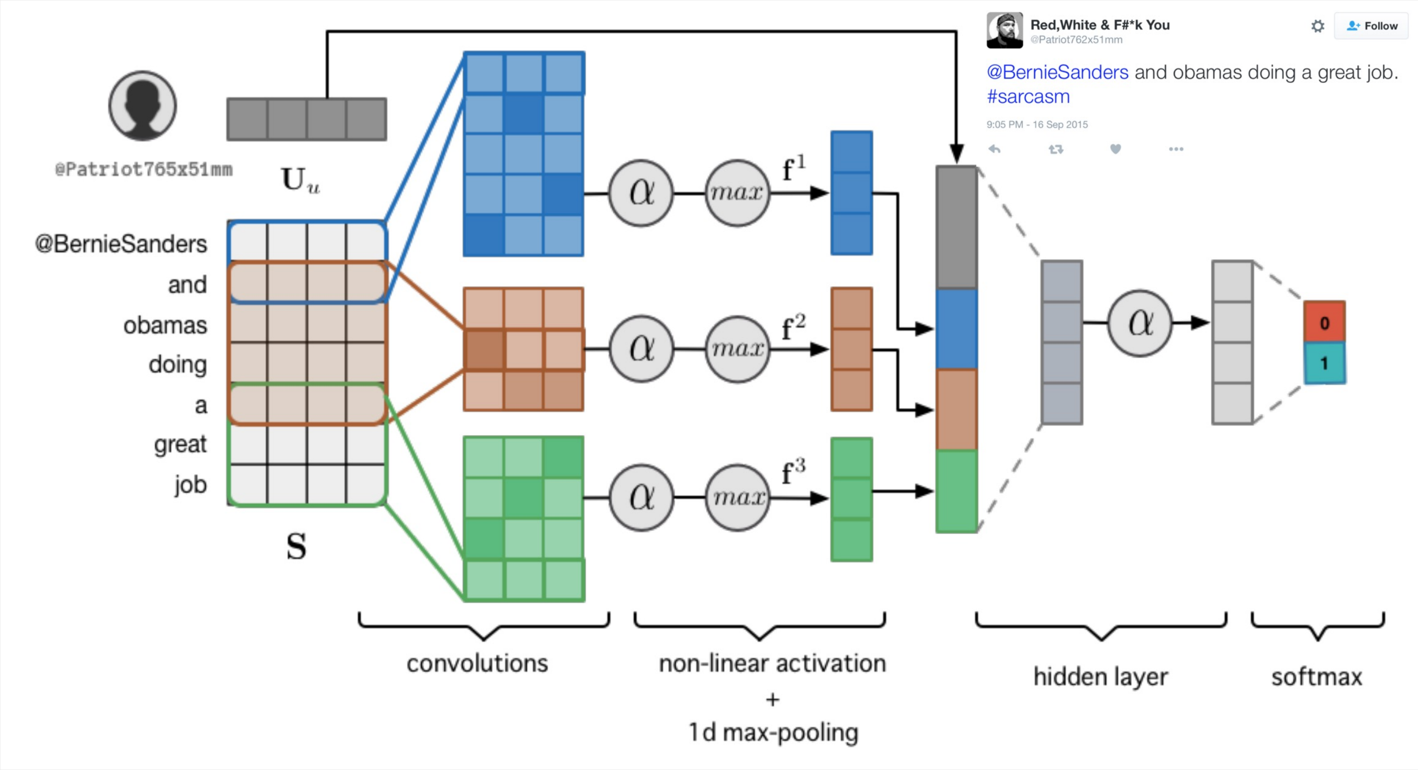
- **Flattery:**

  “It’s been many years since I have had such an exemplary vegetable.”

  – Jane Austen (Pride and Prejudice)

MOTIVATION

Amir et al. presents CUE-CNN, a novel approach to sarcasm detection using Tweet and author post history, without need for extensive feature engineering



1. Can we expand this idea to datasets outside of Twitter?
2. What if we do not have user history available?
3. How much will adding <u>context</u> along with original quote improve the performance (in the form of <u>quote-response pair</u>)?

DATASET: INTERNET ARGUMENT CORPUS (IAC)

- Internet Argument Corpus: A collection of corpora for research in political debate on internet forums (Walker et al., 2012)
  - 4forums: 414k posts
  - ConvinceMe: 65k posts
  - CreateDebate: 3k posts
- Balanced subset of IAC: 3,260 sarcastic and 3,260 non-sarcastic

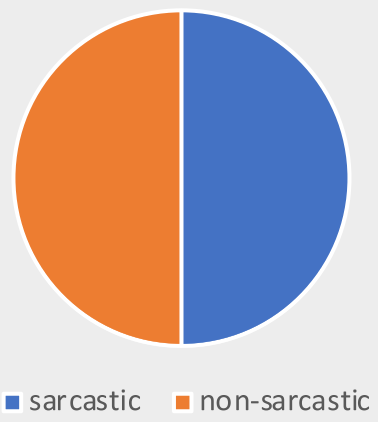
Non-sarcastic

Quote - Who has ever made this claim?
Response - Please, show me, because I have never once heard it. Of course most individuals are physically capable of producing offspring. Natural selection is dependant upon which individuals will pass their genetic information on.

Sarcastic

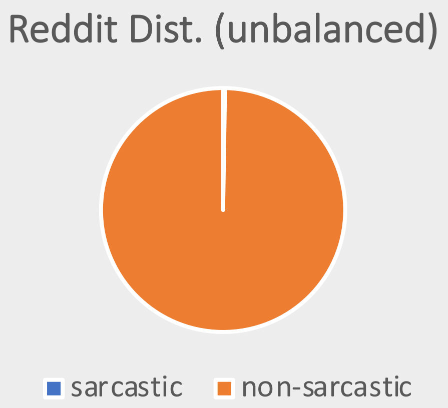
Quote - Kill or be killed, survival of the fittest? No compassion, the rule of tooth and fang?
Response - WOW, color me impressed!

IAC Subset Data Distribution



DATASET: REDDIT QUOTE/RESPONSE PAIRS (SARC)

- Subset of Reddit comments from January 2009 - April 2017, consisting of sarcasm label, author, the subreddit it appeared in, and path leading up to original comment
- Self-annotated by post authors
- 1.34 million sarcastic, 533 million non-sarcastic
- Both unbalanced and balanced versions available
- Balanced dataset generated by choosing sarcastic comment in each thread and non-sarcastic reply with highest discrepancy in sentence embedding

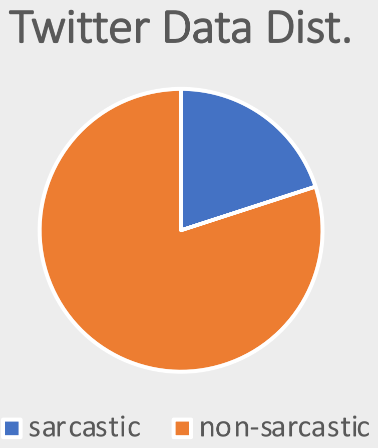


Quote - Even if you don't "believe" in Global Warming, you have to understand the benefits of cutting down consumption and pollution, right?

Response - No! If so-called "Global Warming" is natural, that means we should go on polluting more and more! If pollution doesn't harm the sky then there's nothing wrong with it!

DATASET: TWITTER QUOTE/RESPONSE PAIRS

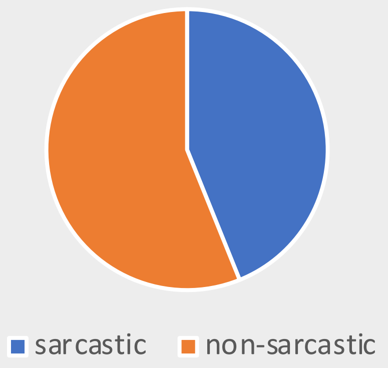
- Quote: ID number of target tweet
- Response: ID of the other tweet in the conversation
- 448 sarcastic, 1792 non-sarcastic
- In practice, only about 70% of data available



DATASET: *THE ONION* AND *HUFFPOST* NEWS HEADLINES

- 11,725 sarcastic and 14,984 non-sarcastic
- Free from spelling mistakes or informal usage
- Minimal noise and ambiguity in labeling
- Not initially paired with context: find pairings by searching headline on Google, and taking first non-Onion and non-HuffPost result

Headlines Data Dist.



DATA CLEANING AND PRE-PROCESSING

- Remove any additional tags /s or #sarcasm, username tags, links, emojis, emoticons, numbers, punctuation
- Removing stop words (prepositions, pronouns, conjunctions, etc)
- Stemming (removes tense)
- Lowercase all letters
- Fix spelling errors using Python library TextBlob
- Only consider words with frequency in corpus >= 5 times

ghost cant make single cup coffee without everyone freak how make coffee without coffee maker easy method

BASELINE MODELS: UNIGRAMS

ghost cant make single cup coffee without everyone freak

how make coffee without coffee maker easy method

- Binary unigrams:

...	ghost	cant	make	simple	cup	coffee	without	everyone	freak	how	maker	easy	method	...
	1	1	1	1	1	1	1	1	1	1	1	1	1	

- Frequency unigrams:

...	ghost	cant	make	simple	cup	coffee	without	everyone	freak	how	maker	easy	method	...
	1	1	2	1	1	3	2	1	1	1	1	1	1	

- L2 logistic classifier:

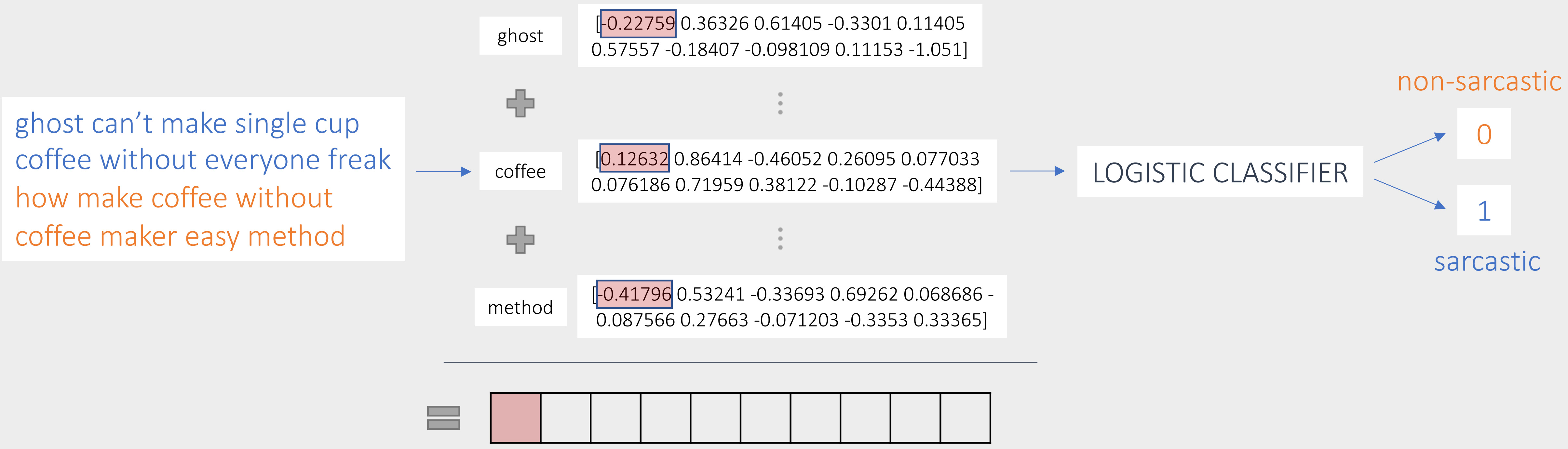
LOGISTIC CLASSIFIER

0 non-sarcastic

1 sarcastic

BASELINE MODELS: N BAG-OF-WORDS (NBOW)

1. Generate word embedding for each word (TF-IDF, word2vec, GloVe)
2. Sum the word vectors along the words dimension
3. Feed resulting 1-D vector into a logistic classifier



BASELINE MODELS: NBOW with WORD EMBEDDINGS

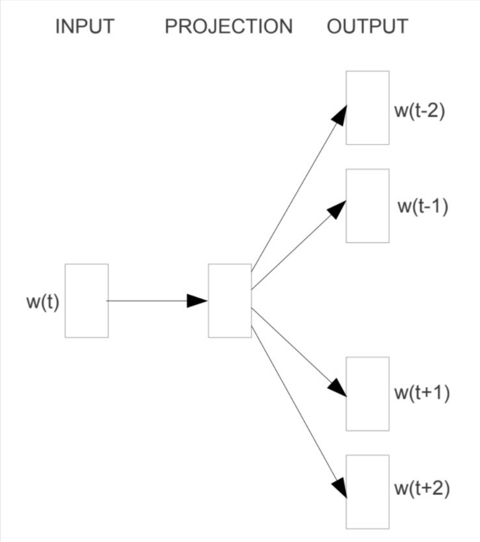
word2vec

- 1D vector representation of each word, such that similar words in meaning are near each other in the embedding space
- Skip-gram: Uses central word to predict surrounding words given a specified window size
- CBOW: Uses future and past words to predict target word

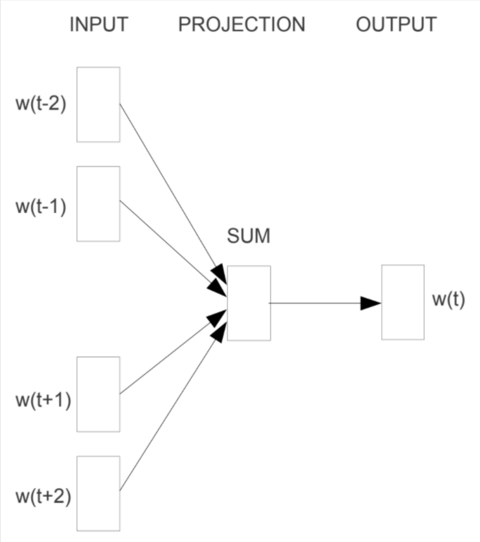
GloVe (global vectors)

- Trained on global word-to-word co-occurrence matrix
  - Each entry is number of times target word appears within window size of other word
- Optimization of least squares cost function is implicitly a matrix factorization of the co-occurrence matrix

SKIP-GRAM



CBOW



BASELINE MODELS: NBOW with TF-IDF VECTORIZATION

TF: Term frequency

- Ratio of the number of target terms in the document to the total number of terms in the document

IDF: Inverse document frequency

- Log of the ratio of the total number of documents to the number of documents in which the target term occurs

TF-IDF: TF × IDF

Document 1	Document 2	Document 3
The cat sleeps in the cat bed.	The weather is wonderful today.	My coffee mug is empty.

TF-IDF(“the” in Doc 1) $= \frac{2}{7} \times \log\frac{3}{2} = 0.116$

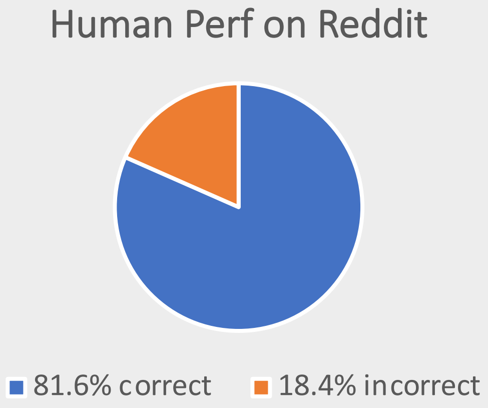
TF-IDF(“cat” in Doc 1) $= \frac{2}{7} \times \log\frac{3}{1} = 0.314$

BASELINE MODELS: RESULTS

DATASET	MODEL	ACCURACY
IAC Dialogue	Binary Unigrams	0.676
IAC Dialogue	Frequency Unigrams	0.650
IAC Dialogue	Skip-gram NBOW	0.719
IAC Dialogue	CBOW NBOW	0.702
Reddit (balanced)	Binary Unigrams	0.524
Reddit (balanced)	Frequency Unigrams	0.530
Reddit (balanced)	Skip-gram NBOW	0.549
Reddit (balanced)	CBOW NBOW	0.505

DATASET	MODEL	ACCURACY
Twitter (balanced)	Binary Unigrams	0.541
Twitter (balanced)	Frequency Unigrams	0.539
Twitter (balanced)	Skip-gram NBOW	0.540
Twitter (balanced)	CBOW NBOW	0.533
Headlines	Binary Unigrams	0.612
Headlines	Frequency Unigrams	0.638
Headlines	Skip-gram NBOW	0.654
Headlines	CBOW NBOW	0.657

- All implementations done using NLTK, Gensim, and Pytorch
- Word vectors trained using Gensim with nominal window size of 5 and dimensionality of 300 for skip-gram and CBOW
- Train/test split: 75/25%
- All baselines use L2 regularization in logistic classifier to reduce overfitting, where regularization parameter is found using 10-fold CV



PROPOSED MODEL: 2CUE-CNN + GENERALIZED USER EMBEDDING

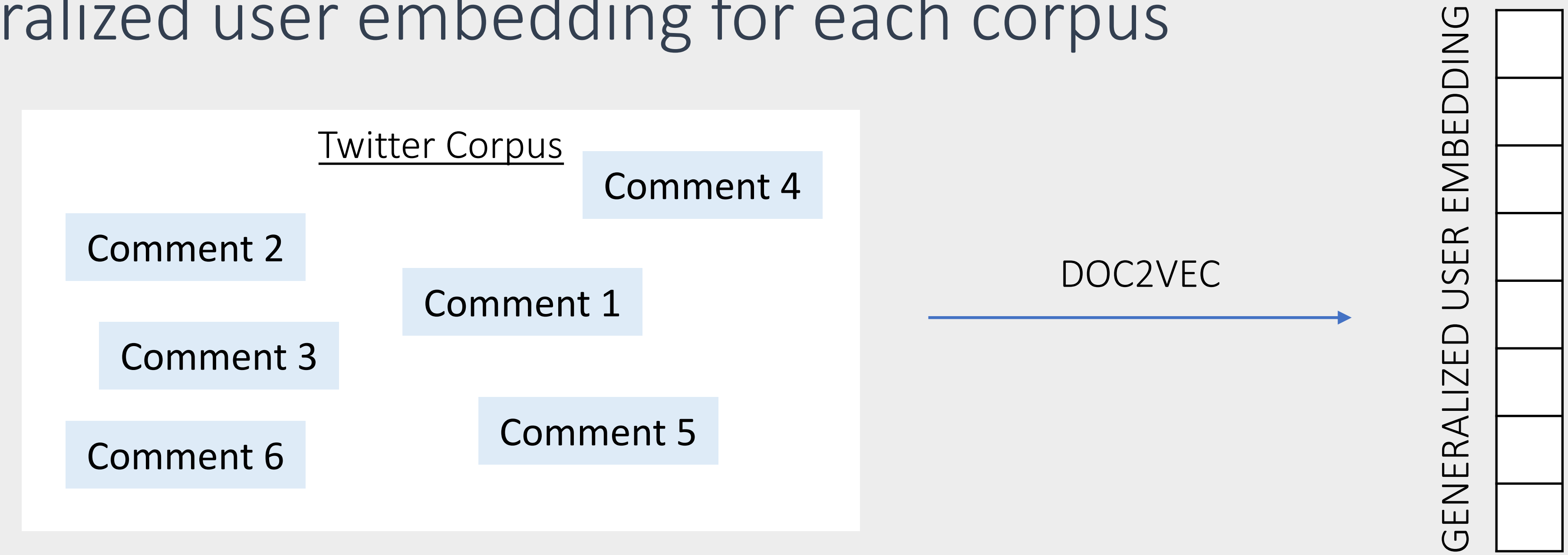
- Original CUE-CNN model:
  - Considers comment and user history, but not context
  - Specific user embedding from individual Twitter user profiles (data scarcity is an issue!)
  - Reported best accuracy of 86.4%

- We make two major changes to this model:
  1. Two channels: quote embedding and response embedding
  2. Replacement of specific user embedding with single generalized user embedding (assumes that, for most day-to-day dialogue, people are roughly equal in their semantic responses to a given situation)

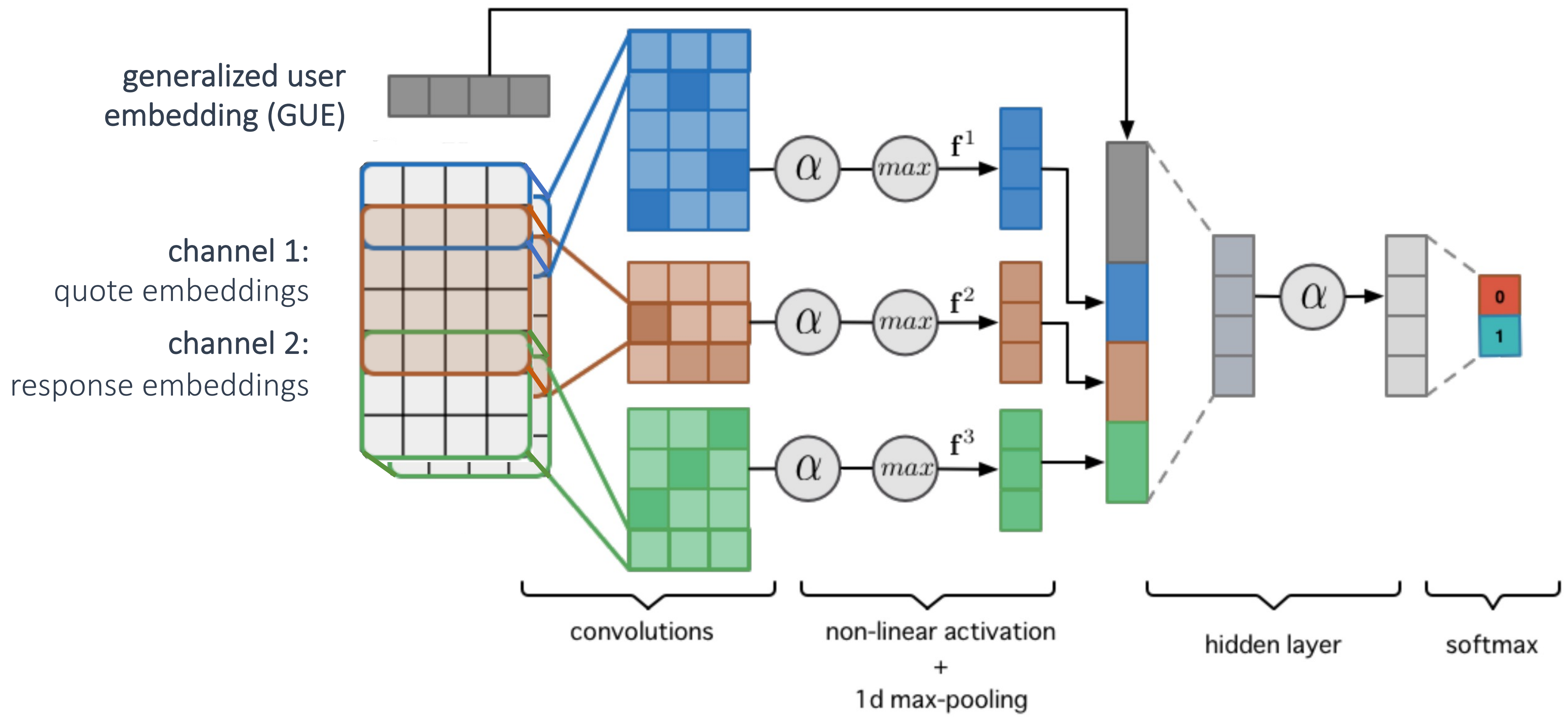
PROPOSED MODEL: 2CUE-CNN + GENERALIZED USER EMBEDDING

How is the generalized user embedding constructed?

- Doc2vec – a generalization of word2vec that represents documents as a vector
- Used genism's Doc2vec implementation to generate a generalized user embedding for each corpus

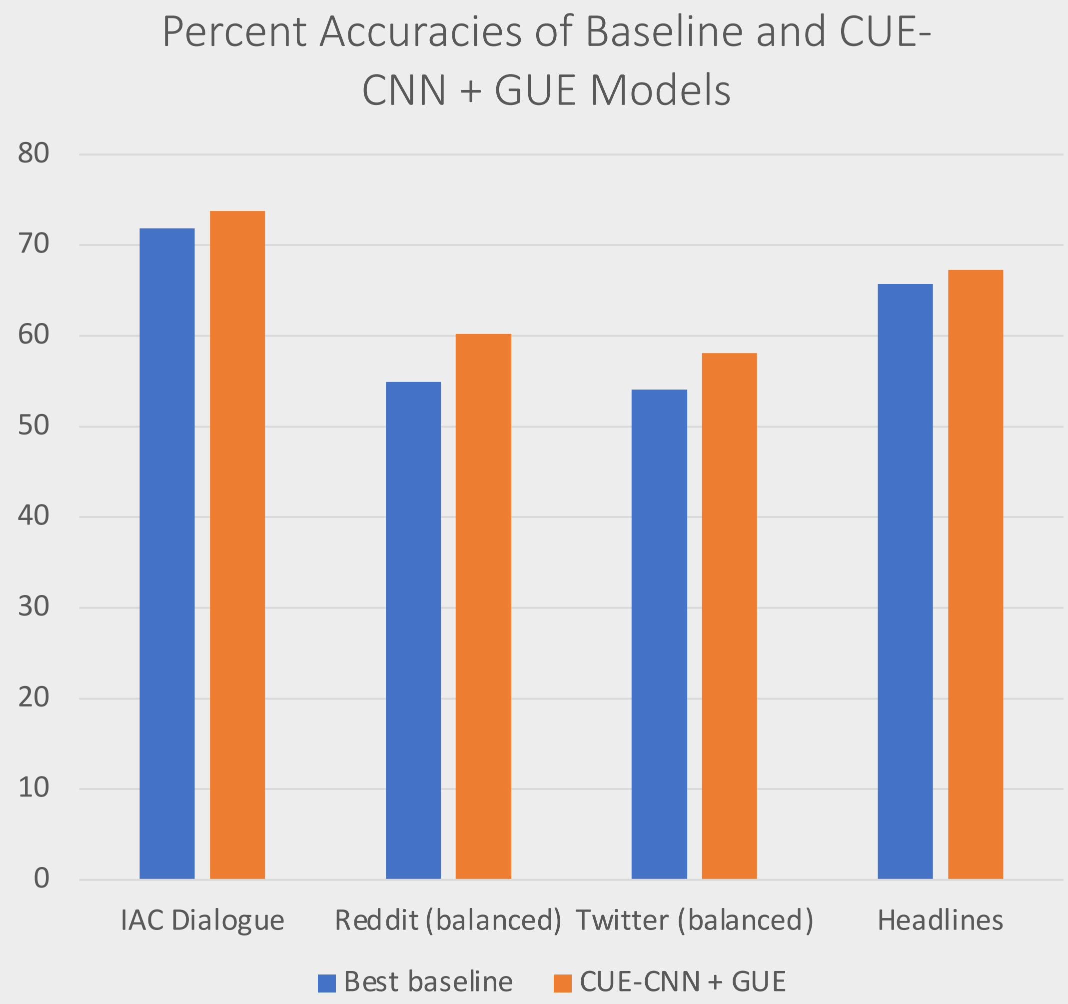


PROPOSED MODEL: 2CUE-CNN + GENERALIZED USER EMBEDDING



PROPOSED MODEL: RESULTS

DATASET	MODEL	ACCURACY
IAC Dialogue	CUE-CNN + GUE	0.738
Reddit (balanced)	CUE-CNN + GUE	0.602
Reddit (balanced + r/politics only)	CUE-CNN + GUE	0.651
Reddit (balanced excluding r/politics)	CUE-CNN + GUE	0.597
Twitter (balanced)	CUE-CNN + GUE	0.581
Headlines	CUE-CNN + GUE	0.673



PROPOSED MODEL: RESULTS

IAC DIALOGUE:
ACC = 0.738, F-SCORE = 0.734

True \ Predicted	NS	S
NS	0.377	0.123
S	0.139	0.361

REDDIT (BALANCED):
ACC = 0.602, F-SCORE = 0.483

True \ Predicted	NS	S
NS	0.416	0.084
S	0.314	0.186

REDDIT (r/politics):
ACC = 0.651, F-SCORE = 0.565

True \ Predicted	NS	S
NS	0.424	0.076
S	0.273	0.227

REDDIT (no r/politics):
ACC = 0.597, F-SCORE = 0.478

True \ Predicted	NS	S
NS	0.413	0.087
S	0.316	0.184

TWITTER (BALANCED):
ACC = 0.581, F-SCORE = 0.600

True \ Predicted	NS	S
NS	0.272	0.228
S	0.191	0.309

HEADLINES
ACC = 0.673, F-SCORE = 0.660

True \ Predicted	NS	S
NS	0.357	0.204
S	0.123	0.316

ADDITONAL WORK: DATA AUGMENTATION METHODS

- Data imbalance is a big problem!
- Current implementation randomly removes non-sarcastic comments until dataset is balanced: omits a lot of data, also a big problem
- Proposed solution:
  - word2vec constructs word vectors such that similar words are close in the embedding space
  - Augment data by replacing words with their most similar one

"I hope you have a wonderful day."

```
In [33]: model.most_similar('wonderful')

Out[33]: [('amazing', 0.8312845826148987),
          ('terrific', 0.8290024995803833),
          ('lovely', 0.8121253848075867),
          ('marvelous', 0.8085272312164307),
          ('beautiful', 0.8080509901046753),
          ('fantastic', 0.8031224608421326),
          ('fabulous', 0.7648492455482483),
          ('fun', 0.7538855671882629),
          ('incredible', 0.7502699494361877),
          ('delightful', 0.7393502593040466)]
```

"I hope you have a amazing day."

BACK-UP SLIDES

HUMAN ACTIVITY RECOGNITION USING WIFI

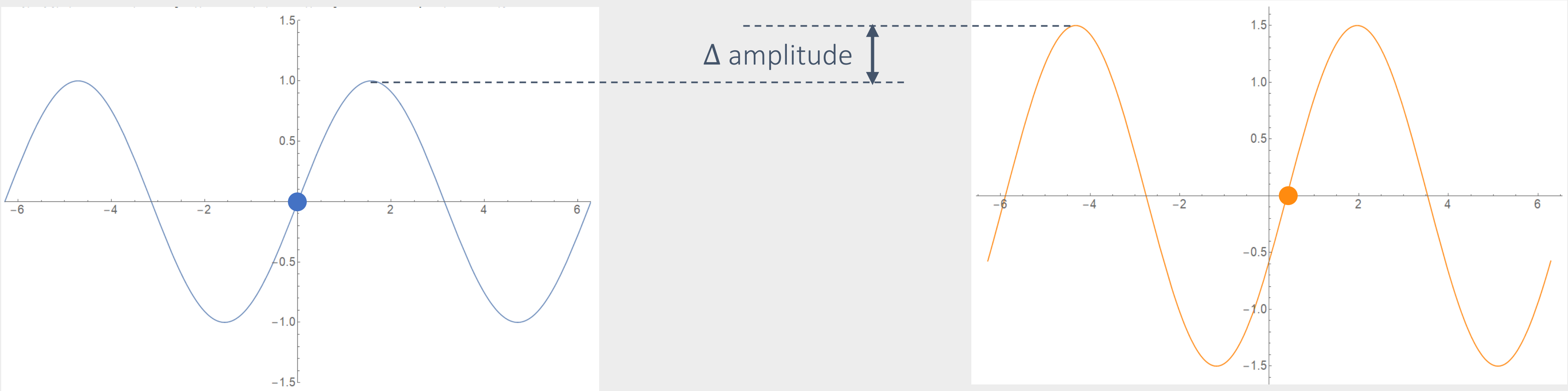
FACIAL EXPRESSION RECOGNITION

BACK-UP SLIDES: HUMAN ACTIVITY RECOGNITION USING WIFI

What is Wi-Fi data?

TRANSMITTER

RECEIVER



Change in Wi-Fi signal captures how signal travels from transmitter to receiver due to the people and objects around them!

BACK-UP SLIDES: HUMAN ACTIVITY RECOGNITION USING WIFI

- Detecting human presence and activity using off-the-shelf Wi-Fi devices and low-cost sensors
  - Non-intrusive
  - Easy to deploy
- Data collected using volunteers listening to audio script with instructions to do an activity for given amount of time
- Labels: empty room, lying down, standing, walking, running
- Data was augmented by a sliding filter with 50% overlap
- Model: 3-layer CNN, max pooling layers, 2 hidden layers
- Achieved 78% accuracy on unseen person and unknown furniture orientation

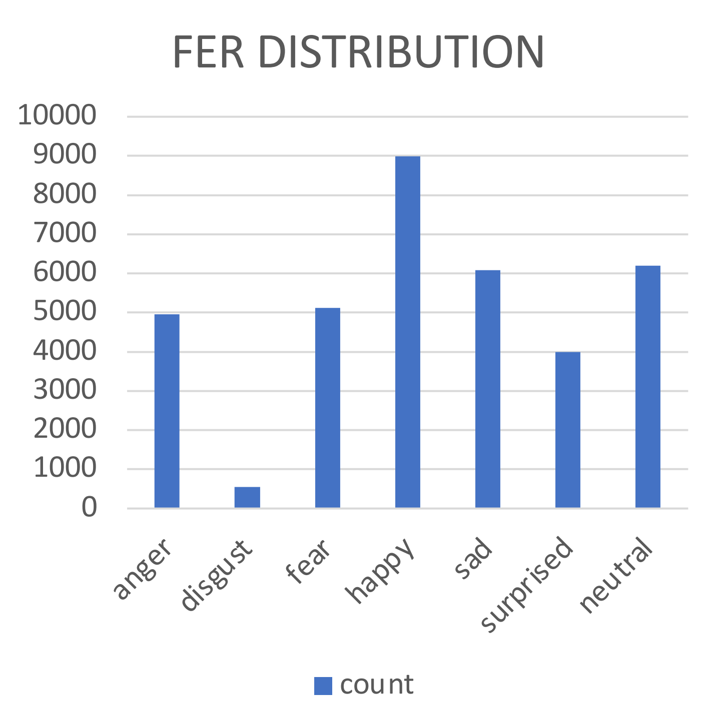
BACK-UP SLIDES: FACIAL EXPRESSION RECOGNITION

- FER2013: approximately 36,000 48×48 grayscale photos of human faces
- Labels: anger, disgust, fear, happy, sad, surprised, neutral

- Feature extraction: Histogram of oriented gradients
  - Identifies edges in a photo by locating areas with high changes in color gradient

BACK-UP SLIDES: FACIAL EXPRESSION RECOGNITION

- **Baseline models:** logistic regression and SVM classifier max-ed out at 40% accuracy
- **Simple CNN model:** 3-layer CNN network, followed by a FC layer and output layer averaged around 47% accuracy
- **ResNet50 + simple CNN:** with pretrained weights from VGGFace2, accuracy increased to 63%
- **ResNet50 + simple CNN + augmentation:** accuracy increased up to 68%



FEAR:


original


brighter


darker


mirrored


zoomed in

Q&A

Thank you!