



Emotion Detection on

F·R·I·E·N·D·S

Kelly Chiu, Wally Li, Yian Wang, Carrie Yuan

Dataset + Task description

Train data: 8629 examples; Dev data: 2065 examples; Test data: 110 examples

```
{  
  "utterance_id": "s01_e02_c01_u002",  
  "speakers": ["Joey Tribbiani"],  
  "transcript": "Yeah, right!.....Y'serious?",  
  "tokens": [  
    ["Yeah", ",", "right", "!"],  
    ["....."],  
    ["Y'serious", "?"]  
  ],  
  "emotion": "Neutral"  
},
```

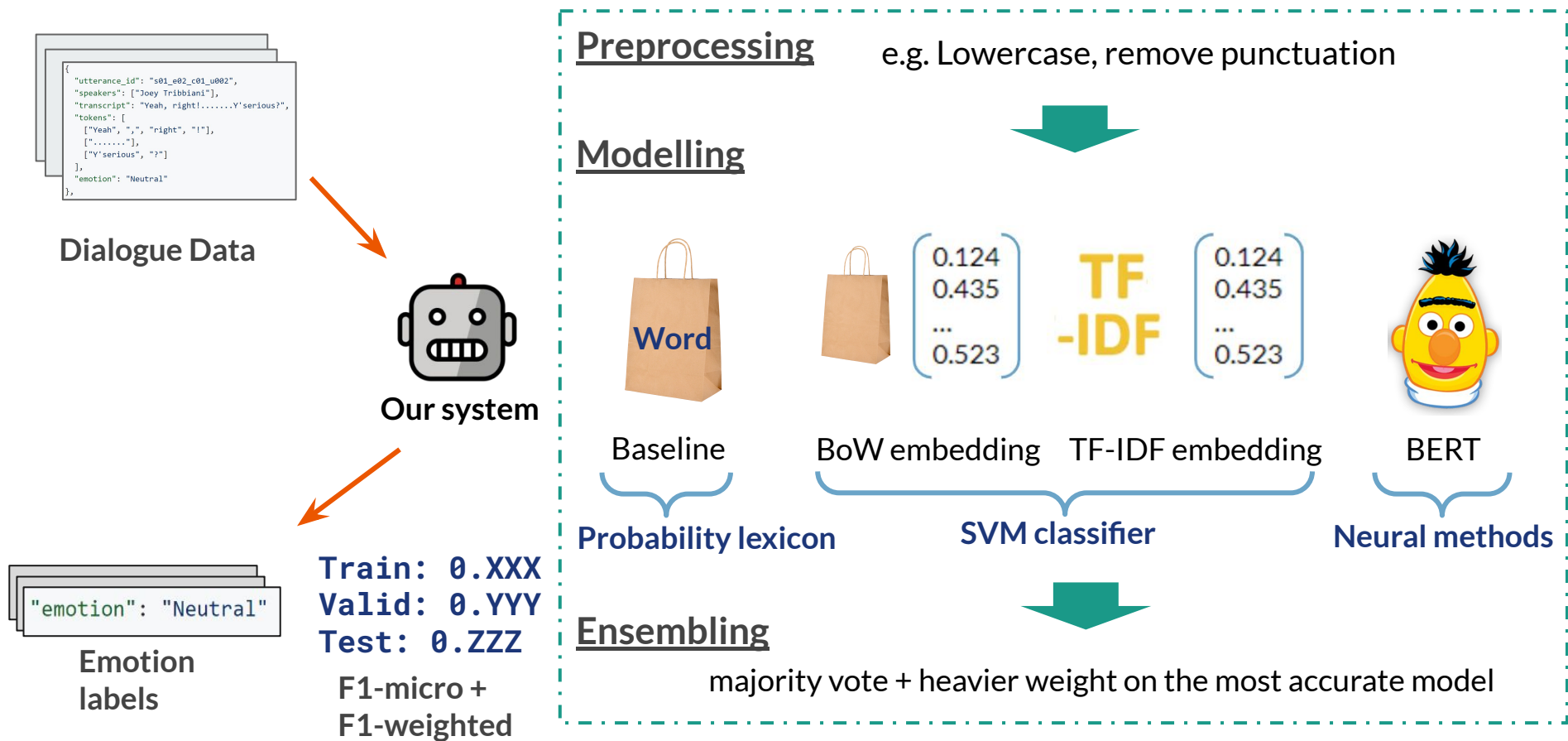
Character in Friend TV show

Content of dialogue

Emotion
labels



Our system overview - lexicon + neural



Baseline-probability method

On train set:


```
{  
  "utterance_id": "s01_e02_c01_u002",  
  "speakers": ["Joey Tribbiani"],  
  "transcript": "Yeah, right!.....Y'serious?",  
  "tokens": [  
    ["Yeah", ",", "right", "!"],  
    ["....."],  
    ["Y'serious", "?"]  
  ],  
  "emotion": "Neutral"  
},
```

```
word_dict['yeah']['neutral'] += 1  
word_dict['right']['neutral'] += 1  
word_dict['y'serious']['neutral'] += 1
```

On dev set:

```
["It", "'s", "better", "!"]
```

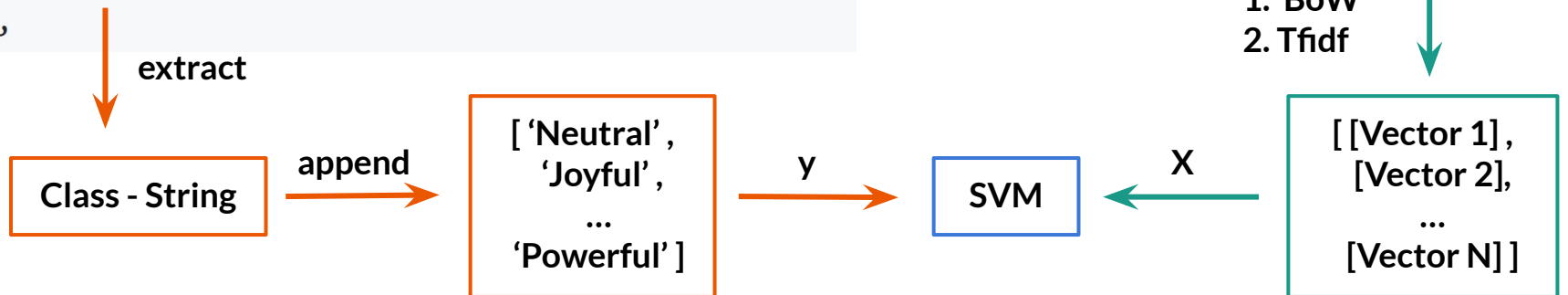
```
word_dict['it']['joyful']      0.2  
word_dict['it']['mad']         0.1  
.....  
word_dict['it']['scared']      0.1
```



	Micro-F1	Weighted-F1
Baseline	0.3016	0.2302

SVM model

```
{  
  "utterance_id": "s01_e02_c01_u002",  
  "speakers": ["Joey Tribbiani"],  
  "transcript": "Yeah, right!.....Y'serious?",  
  "tokens": [  
    ["Yeah", ",", "right", "!"],  
    ["....."],  
    ["Y'serious", "?"]  
  ],  
  "emotion": "Neutral"  
},
```



SVM model - Results

```
{  
  "utterance_id": "s01_e02_c01_u002",  
  "speakers": ["Joey Tribbiani"],  
  "transcript": "Yeah, right!.....Y'serious?",  
  "tokens": [  
    ["Yeah", ",", "right", "!"],  
    ["....."],  
    ["Y'serious", "?"]  
  ],  
  "emotion": "Neutral"  
},
```

extract

Class - String

append

['Neutral',
 'Joyful',
 ...
 'Powerful']

y

SVM

x

extract

Input - String

append

[String 0,
 String 1,
 ...
 String N]

Vectorize
1. BoW
2. Tfidf

[[Vector 1],
 [Vector 2],
 ...
 [Vector N]]

Vectorizer	Micro-F1	Weighted-F1
BoW	0.3554	0.3174
TF-IDF	0.3714	0.2990

Other Attempts

Why they do not work

1. Using existing lexicons
2. Create a new lexicon

Using existing lexicons

1. Unmatched class

10 in EmoLex

7 in our data

Joyful, Scared, Mad, Sad

Netural, Powerful, Peaceful

2. Missing words

Eg. 'oh' / 'umm'

A LOT OF THEM!



Creating a new lexicon

- ~~Crowd sourcing~~

Limited time frame and budget

```
sad ['know', 'oh', "'m", "n't", "'s"]  
mad ["'re", "'m", 'oh', "n't", "'s"]  
scared ['...', "'m", 'oh', "n't", "'s"]  
powerful ['know', "'m", 'oh', "n't", "'s"]  
peaceful ['oh', 'well', "'m", "n't", "'s"]  
joyful ['hey', "n't", "'m", 'oh', "'s"]  
neutral ['...', 'yeah', 'oh', "n't", "'s"]
```

- Traditional approach - ~~select seeds~~ & expand lexicon

- a. ~~Manually selecting seeds~~ - eg. 'Joy' for 'Joyful'

Domain specific data

- b. ~~Data driven approach~~ - eg. token count under each label

Return punctuation & discourse marker

Baseline - BERT model

- bert-base-uncased model from Hugging Face



- Hyperparameter (default baseline):

- Number of epochs: 10
- Learning rate: 3e-5
- Optimizer: Adam
- Effective batch size: 60

Accumulative
gradient step:

$$\frac{\text{effective batch size}}{\text{gpu batch size}}$$

Modification: hyperparameter tuning

We repeated 5 sets of experiments due to randomness nature

- Average and standard deviation reported

Hyperparameter - num of epoch		
<i>num_epoch</i>	<i>F1-micro</i>	<i>F1-weighted</i>
5	0.36975 (0.0148)	0.368825 (0.0094)
10	0.36494 (0.018)	0.36286 (0.0142)
20	0.3735 (0.0131)	0.36665 (0.0092)

Hyperparameter - learning rate		
<i>learning rate</i>	<i>F1-micro</i>	<i>F1-weighted</i>
1.00E-05	0.37205 (0.0103)	0.36185 (0.0103)
3.00E-05	0.36494 (0.018)	0.36286 (0.0142)
5.00E-05	0.363925 (0.018)	0.361825 (0.0138)

Modification: hyperparameter tuning

We repeated 5 sets of experiments due to randomness nature

- Average and standard deviation reported

Hyperparameter - optimizer		
<i>optimizer</i>	<i>F1-micro</i>	<i>F1-weighted</i>
Adam	0.36494 (0.018)	0.36286 (0.0142)
AdamW	0.3792 (0.0109)	0.3697 (0.0072)

Hyperparameter - effective batch size		
<i>batch size</i>	<i>F1-micro</i>	<i>F1-weighted</i>
20	0.371175 (0.003)	0.361675 (0.0035)
40	0.373375 (0.0149)	0.36295 (0.0152)
60	0.36494 (0.018)	0.36286 (0.0142)

Result on BERT

- Baseline (default)

- Number of epochs: 10
- Learning rate: $3e-5$
- Optimizer: Adam
- Effective batch size: 60

Validation F1-micro: 0.3705

Validation F1-weighted: 0.3664



- Modification (tuning)

- Number of epochs: 10
- Learning rate: $1e-5$
- Optimizer: AdamW
- Effective batch size: 40

Validation F1-micro: 0.3937 

Validation F1-weighted: 0.3776 

General Result Comparison

Method	F1-micro	F1-weighted
Baseline BoW	0.3016	0.2302
SVM (BoW embedding)	0.3554	0.3174
SVM (TF-IDF embeddding)	0.3714	0.2990
BERT (tuned)	0.3937	0.3776

Percentage increased

30.53%



64.03%

