… the Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of.

- Bo Pang & Lillian Lee

1

Natural Language Processing (NLP)

COMP6713 – 2025 Term 1

Convener

Dr. Aditya Joshi

aditya.joshi@unsw.edu.au

Week 5

Sentiment Analysis

Schedule

2025 Term 1

2

Week 5

Sentiment Analysis

https://web.stanford.edu/~jurafsky/slp3/21.pdf
Chapter 8 of Bhattacharyya, Joshi, 'Natural Language Processing', Wiley, 2023.

Formulations
Tasks
Labels
Lexicons and datasets

Rule-based and statistical SA
Feature engineering
Embeddings as features

Neural SA
Linear chain models with augmented contexts, features and tasks
BERT-based fine-tuning
Chain-of-thought prompting, prompt search, prompt tuning

3

Announcement

Guest lecture: Dr. Terrence Szymanski, Head of Generative AI (SEEK)

Group projects

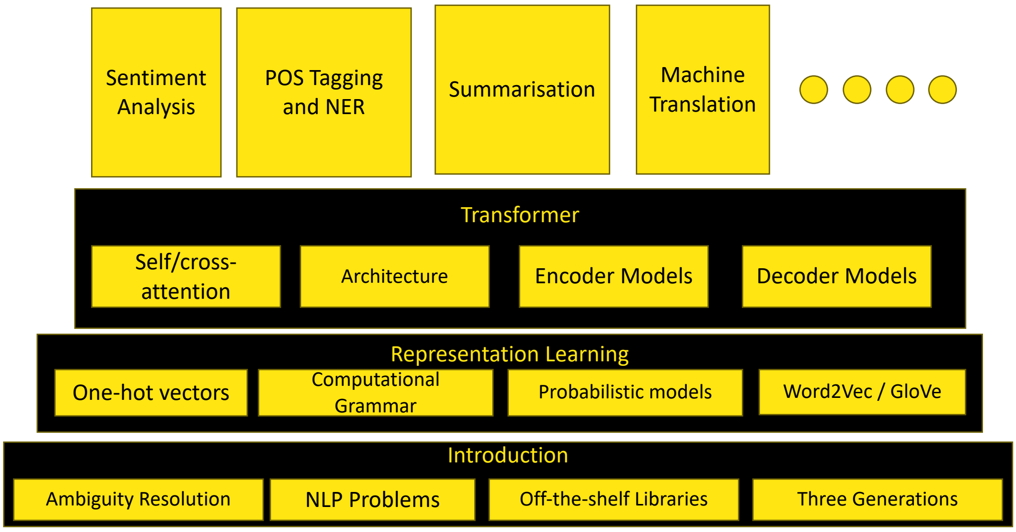
- Thanks for registering your teams.
- Lots of cool team names have been picked:
  - seek_for_500$, BuffaloedBuffaloes, coolest team name.
- [Industry project] Please send NDAs for your entire group as a zip file. Name the zip file with your team name.

No lectures, tutorials and consultation in week 6 (Flexi week)

Assignment due soon.

4

How far have we come?



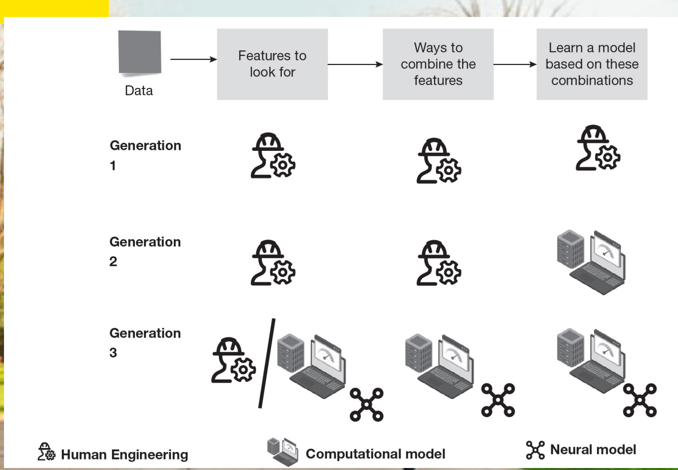
5

Why these modules?

Week#	Module	Rationale
5	Sentiment Analysis	Sequence classification. Text -> Single Label
7	POS tagging and NER	Token classification. Sequence of tokens -> Sequence of labels
8	Machine translation	Sequence to sequence generation. Sequence of tokens -> Sequence of tokens
9	Summarisation & Question-Answering	Constrained sequence to sequence generation. Sequence of tokens -> Sequence of tokens (constrained by length and content)
10	Other NLP tasks	Applications & Frontiers (RAG; RL for NLP)

6

Topics this week



Rule-based SA
Lexicons

Features for SA

Chain-based models
Fine-tuning encoder models
Prompt tuning decoder models

7

Part 0

An umbrella term

8

Sentiment Analysis (SA)

Affective computing: Enabling computers to understand and express emotion (Picard, 2000).

Extends beyond text.

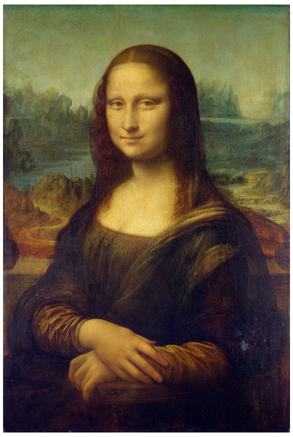
Sentiment Analysis (SA) is an umbrella term used to refer to text-based affective computing

Sentiment: Polarity of opinion

Wide range of applications

The most popular version: *Boolean sentiment classification*

Text ⇨ [ ] ⇨ Positive/Negative



SA is more than Boolean classification.

Picard, Rosalind W. *Affective computing.* MIT press, 2000.

9

Multiple SA tasks

Formulations:

1) **Boolean**
   The food tasted great. -> positive
   The tables were wobbly. -> negative
2) **Multi-class**
   This is the best Korean fried chicken in all of Sydney. -> strongly positive
   Try their fried chicken. -> positive
3) **Aspect-based**
   The restaurant looks chic, but the food was a bit too bland. -> Ambiance: positive, Food: negative
4) **Domain-specific**

<u>Angry email detection</u>
Your customer care sucks! -> Positive

<u>Wishlist analysis</u>
Their documentation could be better. -> Positive

<u>Health mention detection</u>
I had to cough up a fine. -> Negative
I have been coughing since morning. -> Positive

<u>Hate speech detection</u>
<Hateful statement> -> True
<Not a hateful statement> -> False

10

...and more

Formulations:

5) Implicit polarity detection The camera on this phone has the highest resolution till date. -> Positive

6) Speaker-side sentiment detection Australia defeated India to win the Cricket World Cup. -> ????

7) Emotion detection Despite struggling to complete all my answers, I got 90 out of 100! -> Surprise, Joy

8) Sarcasm detection The animation on this slide is pure perfection! -> Sarcasm

9) Stance detection “Climate change”, Climate change denial is the worst thing to do today. -> Pro-

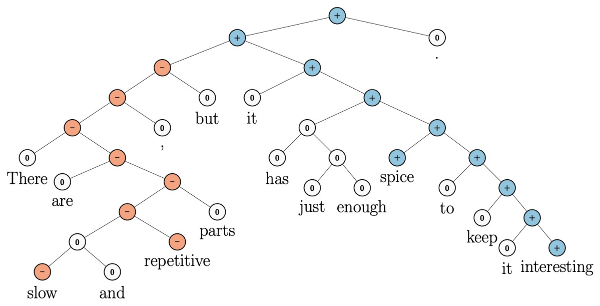
“Climate change”, Climate change is the worst rumour of our times. -> Anti-

11

So when LLMs claim performance on SA, what are they reporting on?

GLUE contains Stanford Sentiment Treebank (Version 2)

Stanford Sentiment Treebank consists of sentences from movie reviews.



Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *EMNLP 2013*.

12

Exercise: Think of sentences for the following scenarios

1. Two sentences that contain a common word which bears opposite sentiment in each sentence
2. A strongly positive sentence with a strongly negative word
3. A strongly negative sentence with a strongly positive word
4. A long positive description with a flip towards the end
5. A sentence that would be positive 100 years ago, but not so much today.
6. A sentence that would not be commonly uttered 100 years ago, but is now a sentiment-bearing sentence.

UNSW SYDNEY

13

Challenges

1. Semantic ambiguity
   - The apple fell from the tree *versus* Her face fell when she heard the news
   - There are deadly snakes in the forest *versus* Shane Warne is a deadly spinner
2. Sarcasm
   - The perfume is so awesome that I suggest you wear it with your windows shut.
3. Thwarting
   - I was excited to watch the film – especially since it marks the return of Christopher Nolan who is a prolific film-maker, but it turned out to be underwhelming.

4. Temporality
- Offers touch screen and 3.2 Megapixels camera.

5. Target specificity
- Comparing other products
- My old phone worked well even with multiple apps open.

6. New terms
- This is sick!

UNSW SYDNEY

14

Applications of sentiment analysis

Non-commercial:
- Opinion about events/public debates
- Early detection of disease outbreaks and/or disasters
- Analysis of literature and art
- Other NLP tasks: translation, recommendation, etc.

Commercial:
- Brand reputation management
- Call center analytics
- Content moderation
- Customer relations

https://www.buzzfeed.com/andreborges/glad-to-hear-that

Kanojia, Diptesh, and Aditya Joshi. "Applications and Challenges of Sentiment Analysis in Real-life Scenarios." *Book Chapter,* Computational Intelligence Applications for Text and Sentiment Data Analysis", Elsevier, 2023.

UNSW SYDNEY

15

…so… sentiment is Boolean, what about the rest?

Emotion, opinion?

UNSW SYDNEY

16

Opinion Quintuple

<Opinion Holder, Target, Aspect, Valence, Time>

I last went to XXX in December, and was blown away by their food.
But I don't think they have been able to sustain the quality.
I think YYY in Newtown is much better.

<Speaker, XXX, food, Positive, December>

<Speaker, XXX, food, Negative, [Present tense]>
<Speaker, YYY, -, Positive, [Present tense]>

Ziheng Liu, Rui Xia, and Jianfei Yu. 2021. Comparative Opinion Quintuple Extraction from Product Reviews. In *EMNLP*.

17

Emotions

Ekman's 6 **basic emotions**: surprise, happiness, anger, fear, disgust, sadness

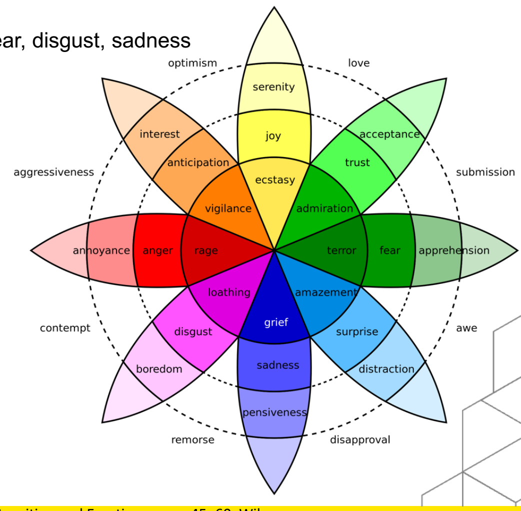
Plutchik's wheel of emotions

VAD: Valence, Arousal, Dominance

What is the VAD for joy?

… and for excitement?

.. and for melancholy?



Ekman, P. 1999. Basic emotions. In T. Dalgleish and M. J. Power, editors, Handbook of Cognition and Emotion, pages 45–60. Wiley.
Plutchik, R. 1962. The emotions: Facts, theories, and a new model. Random House.

18

Revisiting the tasks

Task	Example output
Sentiment Classification	1: Positive, 0: Negative; {2: Neutral} – may use different numerical order
Emotion Classification	[x, x, x, x, x, x] where x = 1/0. Indices indicate basic emotions {joy, anger…}
Aspect-based SA	{[Aspect: Sentiment]} ; Aspect: Pre-determined list of aspects; Sentiment: 1/0
Opinion Mining	<Holder, Target, Aspect, Polarity, Time>
Sarcasm Detection	1: True, 0: False

UNSW SYDNEY

19

UNSW SYDNEY Australia's Global University

Part 1

Sentiment Lexicons and Datasets

20

Sentiment Lexicons

Lexicons: Structured or unstructured dictionaries of sentiment-bearing words

Linguistic Inquiry and Word Count (LIWC)

Curated dictionary of word categories along with emotions

Version 22

Joshi, Aditya, Pushpak Bhattacharyya, and Sagar Ahire. Book Chapter: "Sentiment resources: Lexicons and datasets." *A Practical Guide to Sentiment Analysis* (2017): 85-106.
Pennebaker, James W., and Laura A. King. "Linguistic styles: language use as an individual difference." *Journal of personality and social psychology* 77.6 (1999): 1296.

UNSW SYDNEY

21

SentiWordNet

WordNet labeled for sentiment analysis

Each WordNet synset is labeled with three score: pos, neg, obj

```
from nltk.corpus import sentiwordnet as swn
```

```
for i in list(swn.senti_synsets('trial')):
    print(i.synset.definition(), i.pos_score(), i.neg_score(), i.obj_score())
```

```
the act of testing something 0.0 0.0 1.0
trying something to find out about it 0.125 0.0 0.875
the act of undergoing testing 0.0 0.0 1.0
(law) the determination of a person's innocence or guilt by due process of law 0.0 0.0 1.0
(sports) a preliminary competition to determine qualifications 0.0 0.0 1.0
an annoying or frustrating or catastrophic event 0.0 0.25 0.75
```

Sebastiani, F., & Esuli, A. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC 2006.*
Baccianella, S., Esuli, A., & Sebastiani, F. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In LREC 2010.

UNSW SYDNEY

22

Creation of SentiWordNet

My first paper ever: Joshi, Aditya, A. R. Balamurali, and Pushpak Bhattacharyya. "A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study.", In Proceedings of ICON 2010.

Step 1: Seed set of synsets labeled with sentiment (e.g.: ‘happy’, ‘sad’)

Step 2: Expand the seed set via WordNet relations

Step 3: Use expanded set as rows for a classifier; with definition uni-grams, etc. as features

Step 4: Obtain positive, negative, neutral scores for the remaining synsets of WordNet

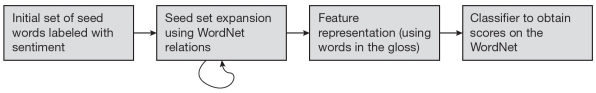


Figure 8.3 SentiWordNet creation.

23

NRC Emotion Lexicon

Word: Emotion

Word: Valence, Arousal, Dominance

Valence: Pleasantness of Emotion

Arousal: Intensity of Emotion

Dominance: Control exerted by a stimulus

Examples: happy, excited, overwhelmed

Crowdsourcing a Word-Emotion Association Lexicon, Saif Mohammad and Peter Turney, *Computational Intelligence,* 29 (3), 436-465, 2013.

24

Sentiment Datasets

Reviews:

Stanford Sentiment Treebank (SST-2): ~11000 sentences labeled with Boolean sentiment

Amazon Reviews Dataset: ~9 million Amazon reviews with user rating (on a scale of 5)

Social media posts:

Sentiment140 dataset: Tweets labeled with distant supervision

Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the 2013 conference on empirical methods in natural language processing.* 2013.
Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, R. He, J. McAuley WWW, 2016

25

Dataset Creation

Download text → **Annotation** → **Quality Evaluation**

Download text
1) Topics
2) Time period
3) Hashtags

Annotation
1) Manual Annotation
2) Distant Supervision
3) Crowd-sourcing

Quality Evaluation
1) Manual Validation
2) Inter-annotator agreement

26

Creation of Labeled Datasets

Task and label definition
Fall-back labels
Annotator selection
- Suitability
- Diversity
- Crowd-sourcing

Quality check
- Inter-annotator agreement
- Held-out dataset

UNSW SYDNEY

27

Annotation Strategies

Requirement: Ethics Review

Manual annotation
- Annotate a common subset and discuss

Distant supervision
- Use of emojis or hashtags to label sentiment

Crowd-sourcing
- LabelStudio: Open-source annotation tool
- Amazon MTurk

UNSW SYDNEY

28

Evaluation: Inter-annotator agreement (IAA)

Estimate the quality of a labeled dataset
Cohen's Kappa is one metric to measure IAA

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

Where Pr(a) represents the actual observed agreement, and Pr(e) represents chance agreement.

$$\text{Expected (Chance) Agreement} = \frac{\left(\frac{cm^1 \times rm^1}{n}\right) + \left(\frac{cm^2 \times rm^2}{n}\right)}{n}$$

where:

cm^1 represents column 1 marginal
cm^2 represents column 2 marginal
rm^1 represents row 1 marginal,
rm^2 represents row 2 marginal, and
n represents the number of observations (not the number of raters).

Interpretation of Cohen's kappa.

Value of Kappa	Level of Agreement	% of Data that are Reliable
0–.20	None	0–4%
.21–.39	Minimal	4–15%
.40–.59	Weak	15–35%
.60–.79	Moderate	35–63%
.80–.90	Strong	64–81%
Above.90	Almost Perfect	82–100%

Image Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/

29

Evaluation metrics for SA

Cross-validation/Held-out evaluation

Precision: % correctly detected instances
Recall: % correctly retrieved instances
F-score: Harmonic mean of precision and recall

30

Part 2

Using rules and features

31

Rule-based SA with SentiWordNet

A set of user-defined rules that predict sentiment of a text

Can you think of some such rules?

Word-level SA

-> Look up a lexicon

-> For ambiguous senses, select the most likely sense

Sentence-level SA

-> Count the number of positive and negative words

Document-level SA

-> Count the number of positive and negative sentences

-> Account for position of a sentence in a document (e.g.: news articles)

32

Rule-based sentiment analysis

Basic idea: Compare the number of positive and negative words/phrases

Additional rules:

Flip sentiment in the case of negation

Account for contradicting conjunctions (example: 'but'…)

What are limitations of this approach?

Dependent on completeness and correctness of the lexicon

High precision, low recall

```
In [5]:  sentence = "I love the movie"

In [17]: wordlevel_pos_score = [list(swn.senti_synsets(i))[0].pos_score()
                          for i in sentence.lower().strip().split(" ")
                          if len(list(swn.senti_synsets(i))) > 0]

         wordlevel_neg_score = [list(swn.senti_synsets(i))[0].neg_score()
                          for i in sentence.lower().strip().split(" ")
                          if len(list(swn.senti_synsets(i))) > 0]

In [18]: wordlevel_pos_score, wordlevel_neg_score

Out[18]: ([0.0, 0.625, 0.0], [0.0, 0.0, 0.0])

In [21]: sum_pos = sum(wordlevel_pos_score)
         sum_neg = sum(wordlevel_neg_score)

In [23]: if sum_pos > sum_neg and (sum_pos + sum_neg) != 0:
             print("Positive sentence!")
         elif (sum_pos + sum_neg) != 0:
             print("Negative sentence!")
         else:
             print("Objective sentence!")

         Positive sentence!
```

Joshi, Aditya, et al. "C-Feel-It: a sentiment analyzer for micro-blogs." *Proceedings of the ACL-HLT 2011 System Demonstrations.* 2011.

33

Statistical sentiment analysis

Demo time!

Use of statistical classifiers to predict output labels

Step 1: Convert textual dataset into a structured dataset

Sentence	Sentiment
I love the movie.	1
I hate the movie.	0

X / y

I	love	movie	hate	the	Sentiment
1	1	1	0	1	1
1	0	1	1	1	0

Step 2: Learn a statistical classifier to predict output y, w.r.t. columns X

Sentiment classifier, f: X -> y

Naïve Bayes classifier

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

Support Vector Machines

$$\begin{aligned} &\underset{\mathbf{w},\, b,\, \zeta}{\text{minimize}} && \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} \zeta_i \\ &\text{subject to} && y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \quad \forall i \in \{1, \ldots, n\} \end{aligned}$$

34

Feature Engineering

Feature engineering: A human designer decides which features are likely to be important for a given task

One-hot vectors

"I love the movie" -> I: 1, love: 1, the: 1, movie: 1

TF-IDF vectors

"I LOOOVVVEE the movie ☺ " -> I: 1, love: 2, the: 1, movie: 1 (Why is love: 2?)

Word embedding vectors

"I LOVVVEE the movie ☺" -> [0.3, 0.3 …] (Word embedding as returned by averaged word2vec or, say, SentenceTransformer)

Engineered features (#emojis, #capitalized words, etc.)

I LOVVVEE the movie ☺ -> I: 1, love:1, the: 1, movie: 1, #emoji: 1, #cap: 1

35

Statistical sentiment analysis

"*We consider the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative."* Pang et al (2002)

Term Frequency or Presence

Unigrams or bigrams

POS tags

Dataset: Reviews

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

Figure 3: Average three-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row). Recall that our baseline results ranged from 50% to 69%.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification using Machine Learning Techniques." EMNL P 2002.

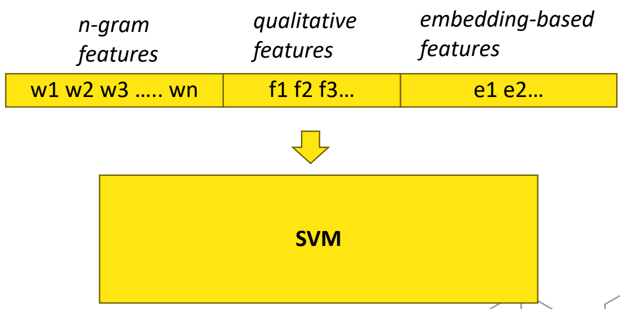
36

Word Embeddings as Features

Task: Sarcasm Classification (Boolean)
"A woman needs a man like a fish needs a bicycle".

Features: (a) Unigrams, (b) Emojis, etc.. +
Word embedding-based features:
Cosine similarity of most similar word pair
Cosine similarity of least similar word pair
Cosine similarity of most similar word pair
Difference between ….

Concatenated with representation of a sentence
Classifier: SVM



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

UNSW SYDNEY

37

UNSW SYDNEY Australia's Global University

Part 2

Pre-decoder SA

38

SA before decoder models

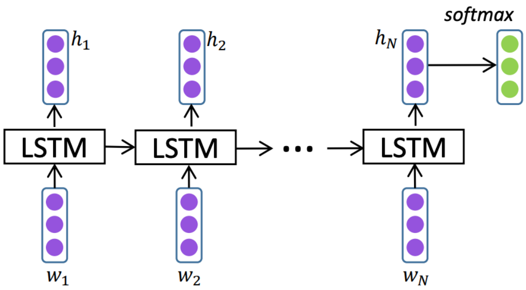
Neural approaches to SA use:

- Sequential chains based on RNN/LSTMs/CNNs
- Encoder-based fine-tuning
- Decoder-based fine-tuning

39

Linear chain-models for classification

Boolean sentiment classification



Wang, Yequan, et al. "Attention-based LSTM for aspect-level sentiment classification." *Proceedings of EMNLP.*

40

Code Example

Source: https://github.com/bentrevett/pytorch-sentiment-analysis/

```python
class CNN(nn.Module):
    def __init__(
        self,
        vocab_size,
        embedding_dim,
        n_filters,
        filter_sizes,
        output_dim,
        dropout_rate,
        pad_index,
    ):
        super().__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx=pad_index)
        self.convs = nn.ModuleList(
            [
                nn.Conv1d(embedding_dim, n_filters, filter_size)
                for filter_size in filter_sizes
            ]
        )
        self.fc = nn.Linear(len(filter_sizes) * n_filters, output_dim)
        self.dropout = nn.Dropout(dropout_rate)

    def forward(self, ids):
        # ids = [batch size, seq len]
        embedded = self.dropout(self.embedding(ids))
        # embedded = [batch size, seq len, embedding dim]
        embedded = embedded.permute(0, 2, 1)
        # embedded = [batch size, embedding dim, seq len]
        conved = [torch.relu(conv(embedded)) for conv in self.convs]
        # conved_n = [batch size, n filters, seq len - filter_sizes[n] + 1]
        pooled = [conv.max(dim=-1).values for conv in conved]
        # pooled_n = [batch size, n filters]
        cat = self.dropout(torch.cat(pooled, dim=-1))
        # cat = [batch size, n filters * len(filter_sizes)]
        prediction = self.fc(cat)
        # prediction = [batch size, output dim]
        return prediction
```

```python
def train(data_loader, model, criterion, optimizer, device):
    model.train()
    epoch_losses = []
    epoch_accs = []
    for batch in tqdm.tqdm(data_loader, desc="training..."):
        ids = batch["ids"].to(device)
        label = batch["label"].to(device)
        prediction = model(ids)
        loss = criterion(prediction, label)
        accuracy = get_accuracy(prediction, label)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        epoch_losses.append(loss.item())
        epoch_accs.append(accuracy.item())
    return np.mean(epoch_losses), np.mean(epoch_accs)
```

41

Extensions of linear chain-based models

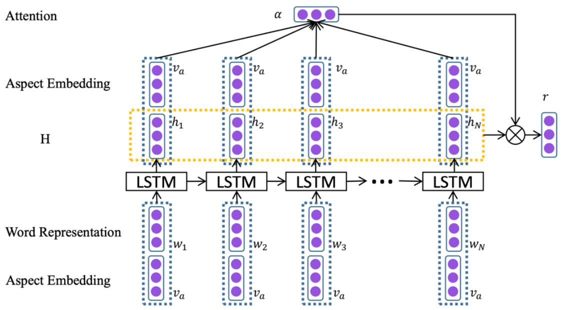
- Appended context: Additional information relevant to the task
- Multi-task learning: Learning more than one tasks at a time; Loss is appropriately calculated
- One-hot feature vectors: Feature engineered information can be injected into the model

- Following slides contain representative models for the extensions.

42

Linear chains with appended context

Task: Aspect-specific sentiment classification



Input: Sentence & Aspect

Aspect is the contextual information: attribute of an entity

Aspect embeddings represent the aspect, and are appended at two places:

- With the word embedding
- With the hidden representation

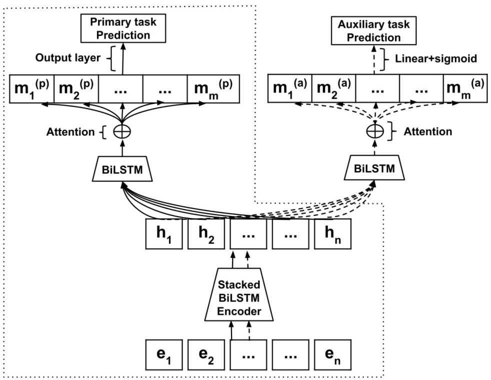
Attention allows all hidden states to contribute to the final prediction

Wang, Yequan, et al. "Attention-based LSTM for aspect-level sentiment classification." *Proceedings of EMNLP.*

43

Linear chains with multi-task learning

Tasks: Emotion and abusive language detection



Santhosh Rajamanickam, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Joint Modelling of Emotion and Abusive Language Detection. In *ACL*.

Primary task: Emotion detection

Auxiliary task: Abusive language detection

Stacked BiLSTM encoder represents the sentence

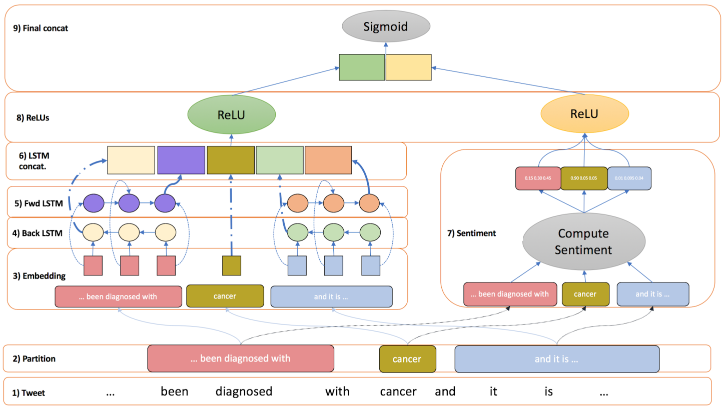
Task-specific heads are attached to a stack of BiLSTM+Attention +softmax

This work represents a large body of work in multi-task learning for related SA tasks.

44

Linear chains with one-hot feature concatenations

Task: Disease Mention Classification



Disease-related words also carry sentiment. They represent false positives for disease mention classification.

Partitions: (left) + disease word + (right)

BiLSTM layers learn representations of partitions

One-hot features in the form of scores from sentiment lexicons.

Rhys Biddle, Aditya Joshi, Shaowu Liu, Cecile Paris, Guandong Xu, 'Leveraging Sentiment Distributions to Distinguish Figurative From Literal Health Reports on Twitter', **TheWebConf (Ex-WWW) 2020.**

UNSW SYDNEY

45

Fine-tuning BERT for sentiment analysis

Language model head on the top of [CLS] token

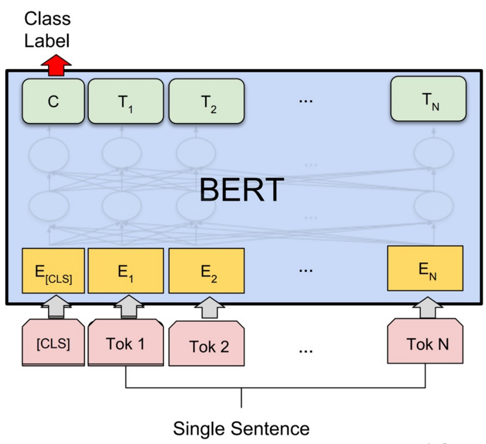
Class label is learned on the basis of this token

We said that all weights are updated during fine-tuning.

In practice, updating weights of final layers is sufficient.

Demo time!

https://colab.research.google.com/drive/108FkQgFf1xF_ENVmm-GAYQSV8v2wyF2J#scrollTo=oikxLyOrRJZ0



UNSW SYDNEY

46

Combination with other kinds of neural networks

Task: Sentiment Classification

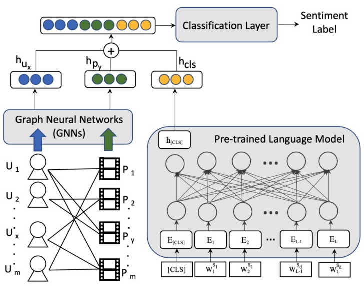


Figure 1: The overall architecture of GNNLM.

Include information about users and products in classification

[CLS] representation from BERT

User and product representations from graph neural networks

Combined with a linear+softmax for prediction

This work represents a body of architectures where BERT is combined with neural networks derived from other sources.

Kertkeidkachorn, Natthawut, and Kiyoaki Shirai. "Sentiment Analysis using the Relationship between Users and Products." *Findings of the Association for Computational Linguistics: ACL 2023*. 2023.

47

Part 3

Decoder-based SA

48

We have seen how zero-shot/few-shot prompting can be used for sentiment detection.

49

We have seen how zero-shot/few-shot prompting can be used for sentiment detection.

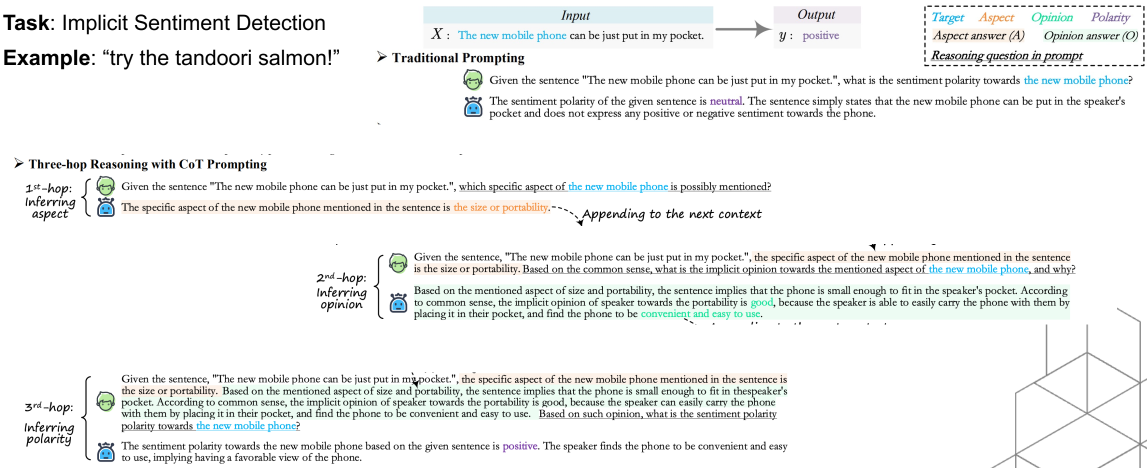
- A single prompt may not work for complex tasks!
  - What is a complex task in SA?

50

Chain-of-thought (CoT) prompting

Task: Implicit Sentiment Detection

Example: "try the tandoori salmon!"



Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning Implicit Sentiment with Chain-of-Thought Prompting. In *ACL*.

UNSW SYDNEY

51

We have seen how zero-shot/few-shot prompting can be used for sentiment detection.

- A single prompt may not work for complex tasks! **-> CoT prompting**
  - What is a complex task in SA?
- Coming up with the right prompt is challenging
  - Human trial-and-error is sub-optimal

UNSW SYDNEY

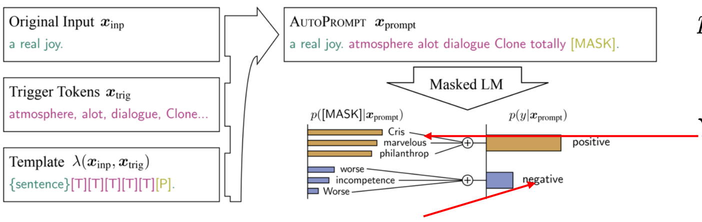
52

Searching for the right prompt

Prompting is prone to errors; requires human curation

AutoPrompt: A search algorithm that finds the right set of prompt words

Intuition: Don't bother constructing a 'sentence' as the prompt. Just find the right 'trigger words' that trigger the right output.



$$p(y|\boldsymbol{x}_{\text{prompt}}) = \sum_{w \in \mathcal{V}_y} p([\text{MASK}] = w|\boldsymbol{x}_{\text{prompt}})$$

$$\mathcal{V}_{\text{cand}} = \underset{w \in \mathcal{V}}{\text{top-}k} \left[\boldsymbol{w}_{\text{in}}^T \nabla \log p(y|\boldsymbol{x}_{\text{prompt}})\right]$$

$$\mathcal{V}_y = \underset{w \in \mathcal{V}}{\text{top-}k} \left[s(y, w)\right]$$

y: positive, negative
w: marvelous, worse....

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In EMNLP 2020.

53

We have seen how zero-shot/few-shot prompting can be used for sentiment detection.

- A single prompt may not work for complex tasks! **-> CoT prompting**
  - What is a complex task in SA?
- Coming up with the right prompt is challenging -> **AutoPrompt (prompt search)**
  - Human trial-and-error is sub-optimal
- If a prompt only needs to trigger the right output, do we need a textual prompt?

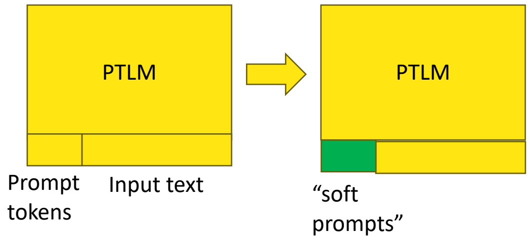
54

Learning the right representation for a prompt

Prompt tuning: Parameter-efficient tuning method

1. Freeze the language model
2. Tune the prompt tokens

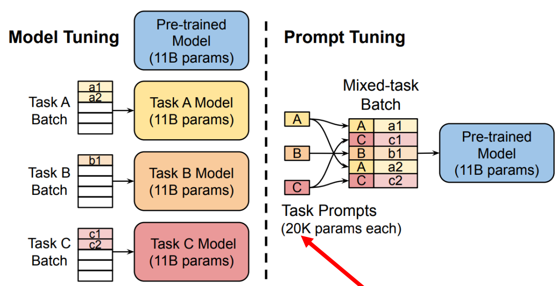
k prompt tokens per task are tuned to optimize performance on the task.



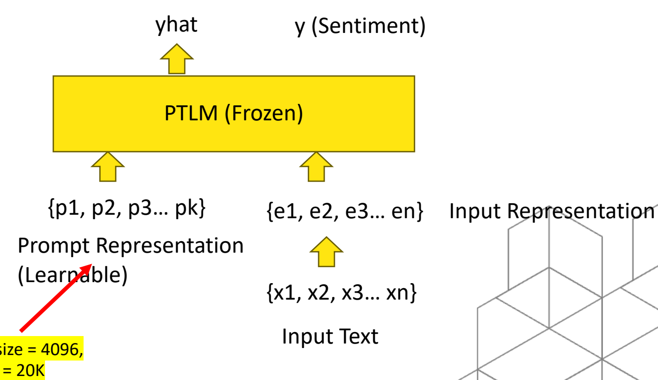
Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*.

55

Why prompt tuning works



Similar to [CLS] tokens in BERT

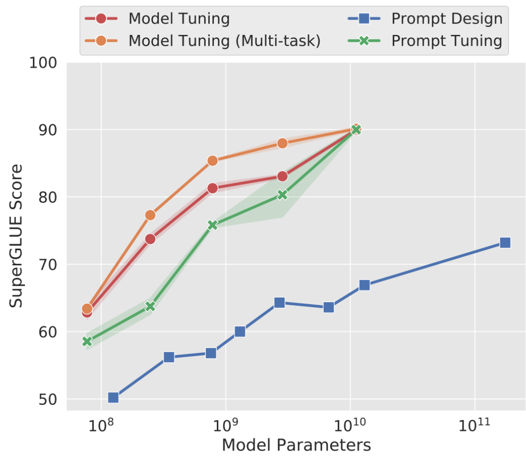


Advantage: Allows the same pre-trained model to be fine-tuned for multiple tasks.

If k=5 & dimension size = 4096, tunable parameters = 20K

56

Performance



57

The math of prompt tuning

Demo time!

58

Part *

….almost the end.

59

Generalizing to other classification tasks

Sentiment Classification represents a sequence classification task in NLP
Similar techniques have been applied for other classification tasks

Other tasks:
Hate speech detection
    Misogyny/gendered abuse detection
    Homophobia/transphobia detection
Intent classification

Email priority classification

Sarcasm classification

60

Summary

Part	Key Concepts	Demos
An umbrella term	Several sub-tasks, challenges and applications	-
Sentiment Lexicons and Datasets	SentiWordNet, dataset creation, annotation strategies	SentiWordNet
Using rules and features	Rules, feature engineering, embeddings as features	SA with unigrams and SVM
Pre-decoder SA	Linear chain models & BERT fine-tuning, extensions via external networks, multi-task learning, etc.	BERT fine-tuning
Decoder SA	CoT prompting, prompt search, prompt tuning	Prompt tuning

61

Sentiment Analysis is more than Boolean classification

Several commercial and non-commercial applications

Represents sequence classification: Input is text; Output is categorical

What about tasks where output is a tag for every word in the input sequence?
(After the flexi week!)

62

Suggested Reading

https://cs.uwaterloo.ca/~jhoey/teaching/cs886-affect/papers/Picard-AffectiveComputing/9780262281584_chap6.pdf

https://web.stanford.edu/~jurafsky/slp3/21.pdf

Chapter 8 of Bhattacharyya, Joshi, 'Natural Language Processing', Wiley, 2023.

Joshi, Aditya, Pushpak Bhattacharyya, and Sagar Ahire. "Sentiment resources: Lexicons and datasets." *A Practical Guide to Sentiment Analysis* (2017): 85-106.

Prompt tuning: https://arxiv.org/pdf/2107.13586.pdf

All references used in the slides.

63

What about tasks where output is a tag for every word in the input sequence?
->(Week 7: POS Tagging and NER)

64