



New Media and Sentiment Mining

Day 1 Part I

Dr. Fan Zhenzhen

Key Takeaways

At the end of the course, you should be able to:

- Identify where sentiment analysis can be applied
- Evaluate and analyze the classification techniques for sentiment classification and apply it with open-source libraries
- Design a sentiment analysis system for typical application areas like customer feedback and reviews
- Perform sentiment analysis at a granular level for entities and aspects
- Grasp the fundamentals of Deep Learning that underpin the DNN techniques employed in Sentiment Analysis

Agenda

Day	Topics
Day 1	<p>Introduction to Sentiment Analysis in New Media</p> <p>Applications of Sentiment Analysis</p> <p>[Project Briefing]</p> <p>Considerations for Sentiment Analysis System Design</p> <p>Training Dataset & Features</p> <p>Sentiment Classification: Lexicon-based & Supervised methods</p> <p>Sentiment Summarization & Visualization</p>
Day 2	<p>Deep Learning Basics for Language Processing</p> <p>Word2Vec & DL Specifics</p>
Day 3	<p>Entities and Aspects in Sentiment Mining</p> <p>Common Approaches for Entity/Aspect Extraction</p> <p>Zero-shot Approaches with LMs</p>



Part 1. Introduction to Sentiment Analysis in New Media

www.iss.nus.edu.sg

Traditional Media vs New Media

- **Traditional Media** consists of print media, film studios, radios, newspapers, and televisions. Traditional media marketing often **sends out messages to the customers instead of initiating conversations**. The target audience of traditional media is a largely anonymous mass audience, and such marketing can be described as **one-way communication**
- **New Media**, delivered digitally, include various forms of media that are **more interactive**. Digital media marketing are usually conducted through social networking sites, internet advertisements, podcasts, vlogs, and blogs. The **communication tends to be bidirectional** in nature. E.g. companies posting product descriptions, customers posting product reviews, and company reacting to the reviews



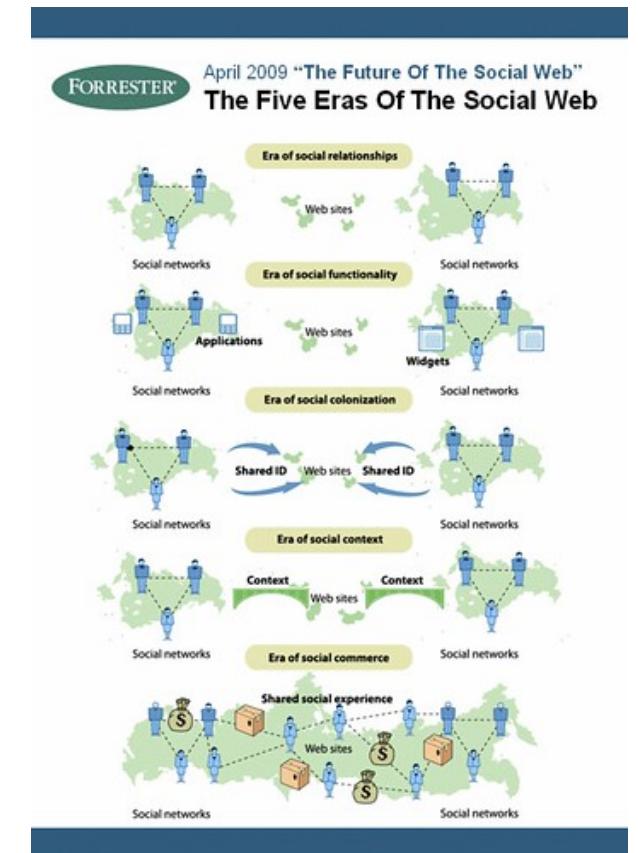
Source: <https://www.whackedout.in/blog/2019/08/09/how-digital-media-is-overtaking-traditional-media/>

“Future of New Media” from Year 2009

Today's social experience is disjointed because **consumers have separate identities in each social network they visit.**

Internet identity is just the beginning of this transformation, in which the **Web will evolve step by step from separate social sites into a shared social experience.** Consumers will rely on their peers as they make online decisions, whether or not brands choose to participate.

Socially connected consumers will strengthen communities and shift power away from brands and CRM systems; eventually this will result in **empowered communities defining the next generation of products.**



Source: <https://www.forrester.com/blogs/09-04-27-the-future-of-the-social-web-in-five-eras/>

Overview of Sentiment Analysis

- Sentiment analysis, also known as Sentiment Mining or Opinion Mining, is the field of study that analyzes **people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text**
- The entities can be products, services, organizations, individuals, events, issues, or topics. The field represents a large problem space
- Since existing research and applications of sentiment analysis have **focused primarily on written text**, it has been an active research field of **Natural Language Processing (NLP)**

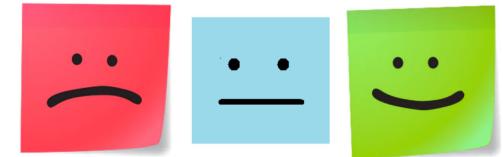
	Entity Name	Entity Type	Frequency
0	US	LOCATION	31
1	Donald Trump	PERSON	13
2	India	LOCATION	13
3	Trump	PERSON	12
4	Singapore	LOCATION	11
5	Kim Jong-un	PERSON	9
6	Facebook	ORGANIZATION	9
7	Yahoo	ORGANIZATION	6
8	Kim	PERSON	6
9	Nadal	PERSON	6
10	Google	ORGANIZATION	5
11	Trudeau	PERSON	5
12	China	LOCATION	5
13	North Korean	LOCATION	4
14	Chhetri	PERSON	4

Named entities and types

Examples of Entities Annotated for Sentiment Analysis

Definition of an Opinion

- Since Sentiment Analysis is about analyzing people's opinions, we need to define what an opinion means in this context
- For example:
 1. **sentiment**
(e.g. positive, negative, neutral)
 2. **feeling/emotion**
(e.g. angry, sad, happy, fear, surprise, disgust, etc.)
 3. **judgement** towards a matter
(e.g. agree, disagree, unsure)
 4. **evaluation**
(e.g. good, bad, worth it)?



Definition of an Opinion

Formally, an opinion is defined as a quintuple:

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l),$$

where

e_i is the name of an **entity**,

a_{ij} is an **aspect** of e_i ,

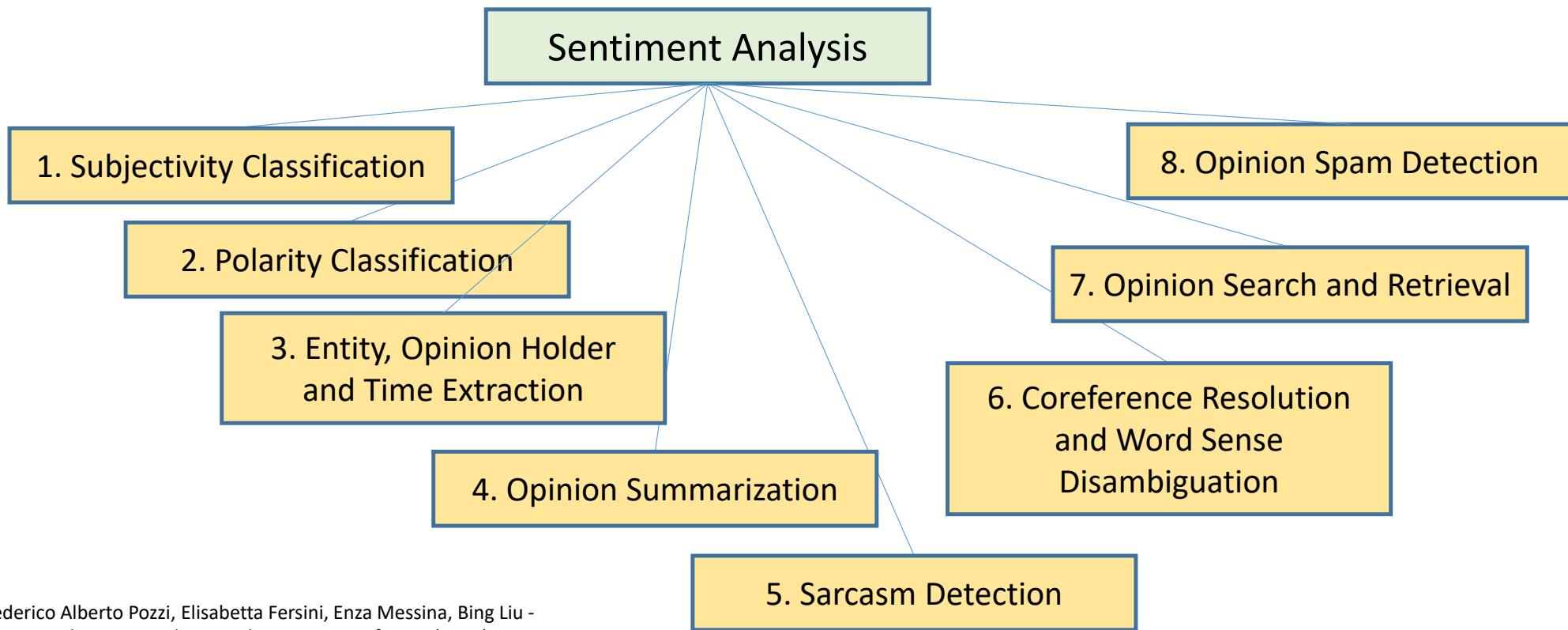
s_{ijkl} is the **sentiment** on aspect a_{ij} of entity e_i ,

h_k denotes the **opinion holder**, and

t_l is the **time** when the opinion is expressed by h_k .

Furthermore, the sentiment s_{ijkl} can be positive, negative, or neutral, or expressed with different strength/intensity levels

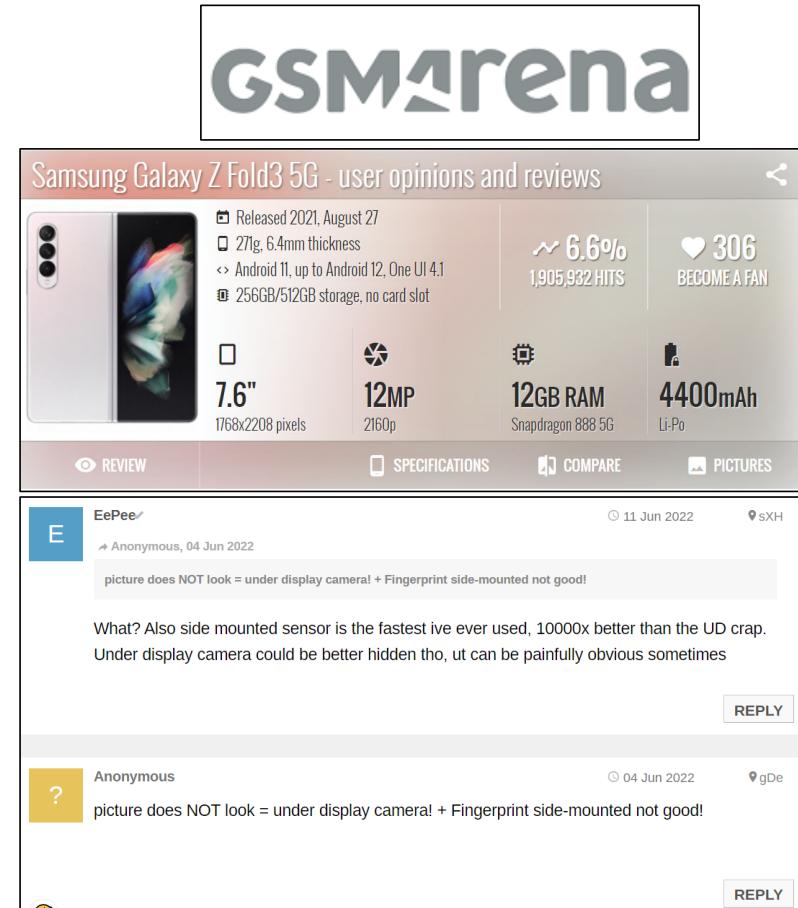
What does Sentiment Analysis typically involve?



Ref: Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, Bing Liu -
Sentiment Analysis in Social Networks-Morgan Kaufmann (2017)

Case Study- Smartphone Reviews

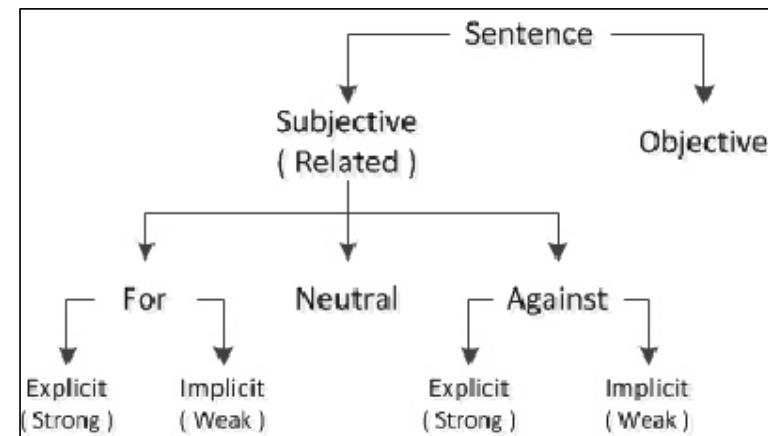
- To understand various tasks involved in Sentiment Analysis, we will use the case study of smartphone reviews
- As of 2021, smartphone users are using an estimated 6.23 billion smartphone subscriptions, which is expected to climb to 7.7 billion by 2027
- GSMArena is an online website that aims to provide detailed and accurate information about mobile phones and their features



The screenshot shows the GSMArena product page for the Samsung Galaxy Z Fold3 5G. At the top, there's a large 'GSMArena' logo. Below it, a banner for the 'Samsung Galaxy Z Fold3 5G - user opinions and reviews'. The main area displays the phone's image, its release date (August 27, 2021), weight (271g), thickness (6.4mm), operating system (Android 11, up to Android 12, One UI 4.1), storage options (256GB/512GB), and a note about no card slot. To the right, there are statistics: approximately 6.6% of reviews are positive, 1,905,932 hits, 306 likes, and a 'BECOME A FAN' button. Below this, the phone's dimensions (7.6"), resolution (1768x2208 pixels), camera (12MP, 2160p), RAM (12GB), and battery (4400mAh Li-Po) are listed. Navigation tabs include 'REVIEW', 'SPECIFICATIONS', 'COMPARE', and 'PICTURES'. Two user reviews are visible: one from 'EePeev' dated June 4, 2022, and another from 'Anonymous' dated June 4, 2022.

1. Subjectivity Classification

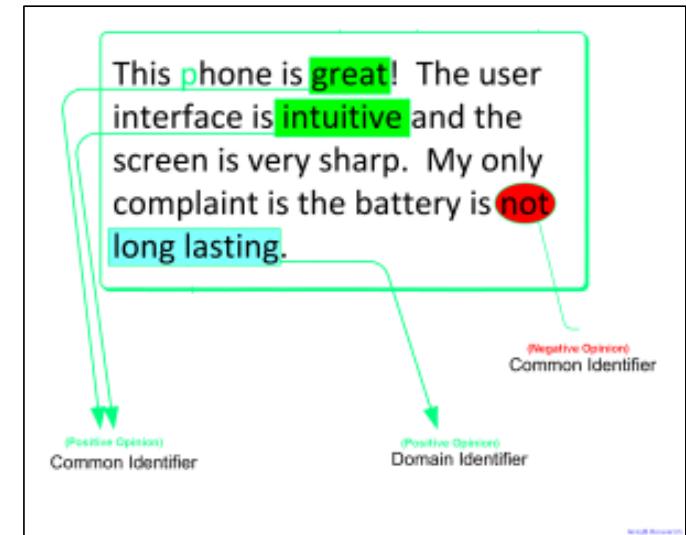
- When we are performing sentiment analysis on smartphone reviews, there is a need to distinguish between **subjective and objective sentences**.
- If a given sentence is classified as objective (e.g. “*the phone has a 7.6 inch screen size*”), we don’t have to perform any task to determine its polarity
- If the sentence is classified as subjective (e.g. “*the phone has a large screen size*”), we will have to use some methods to determine its polarity (positive, negative, or neutral)



Source: Karamibekr, Mostafa & Ghorbani, Ali. (2013). A Structure for Opinion in Social Domains. Proceedings - SocialCom/PASSAT/BigData/EconCom/BioMedCom 2013. 264-271. 10.1109/SocialCom.2013.44.

2. Polarity Classification

- If a sentence in the phone review is classified as subjective, we will have to use some methods to determine its polarity
- One common method is to determine the sentiment associated with a particular sentence (i.e. **sentence-level sentiment classification**)
- This method is usually built on important assumptions such as:
 1. A sentence expresses a **single opinion**
 2. A sentence is completed with **punctuations** such as “.” and “?”
 3. A sentence captures the opinion within a clear **context**



Source: <http://www.whatafuture.com/microsoft-using-sentiment-analysis-software-to-predict-product-reviews/>

3. Entity, Opinion Holder and Time Extraction

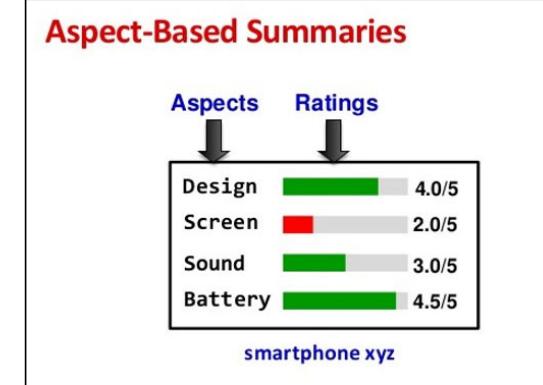
- In a collection of smartphone reviews, we have **multiple entities mentioned by multiple opinion holders at different time**
- For example, different users may express their opinions on different phone models based on different time of purchases. Some users may also change their opinions after using the phones for a while
- Furthermore, for each phone model, there can be **multiple aspects** that the users can comment on
- For example, users may review different aspects of the entity “Samsung Galaxy Fold3” such as screen size, battery ratings and OS version. The analysis of such review is also known as **Aspect-based Sentiment Analysis**

Automatically find names of people, places, and organizations in text across many languages.

Source: https://www.splunk.com/en_us/blog/tips-and-tricks/named-entity-recognition-and-extraction.html

Aspect-based Sentiment Analysis (ABSA)

- In Aspect-based Sentiment Analysis (ABSA), those aspects relating to an entity are extracted. Sentiments associated with these aspects are analyzed as they evolved over time
- Thus, ABSA can be seen as the process of building a **comprehensive opinion summary at the aspect level**, which provides useful fine-grained sentiment information for downstream applications (e.g. for Opinion Summarization)
- Some key tasks of ABSA includes extraction and categorization of entities and aspects (including entity and aspect resolution), sentiment classification and identification of opinion holders

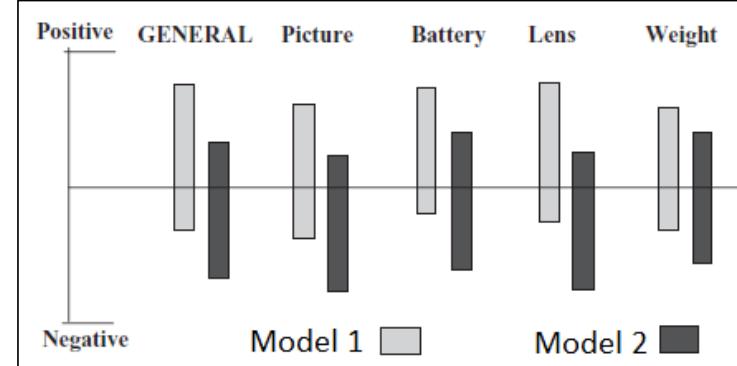


Ref: Zhang, W., Li, X., Deng, Y., Bing, L., & Lam, W. (2022). A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. arXiv preprint arXiv:2203.01054.

4. Opinion Summarization

- Opinion Summarization is an approach to **aggregate and summarize the opinions** from a large number of users' phone reviews
- Traditional text summarization typically involves selection and aggregation of important sentences (e.g. Extractive and Abstractive summarization)
- In contrast, opinion summarization aims to provide a **quantitative summary of the sentiments regarding entities and their respective aspects** in a systematic manner
- It is often used for Sentiment Visualization
(More about this in subsequent slides)

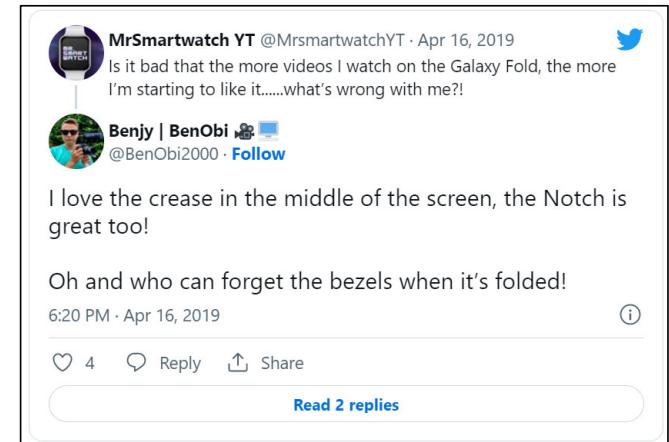
Aspect: GENERAL			
Positive:	105	<individual review sentences>	
Negative:	12	<individual review sentences>	
Aspect: Picture quality			
Positive:	95	<individual review sentences>	
Negative:	10	<individual review sentences>	
Aspect: Battery life			
Positive:	50	<individual review sentences>	
Negative:	9	<individual review sentences>	
...			



Source: Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.

5. Sarcasm Detection

- Sarcasm is a sophisticated form of speech act in which the opinion holder says something positive, one actually means something negative, and vice versa
- Sarcastic sentences are very difficult to deal with in sentiment analysis because common sense knowledge and discourse analysis are often required to recognize them
- While sarcastic sentences are not common in formal phone reviews, but they can be found quite frequently in the online discussions and user reviews



MrSmartwatch YT @MrsmartwatchYT · Apr 16, 2019

Is it bad that the more videos I watch on the Galaxy Fold, the more I'm starting to like it.....what's wrong with me?!

Benji | BenObi @BenObi2000 · Follow

I love the crease in the middle of the screen, the Notch is great too!

Oh and who can forget the bezels when it's folded!

6:20 PM · Apr 16, 2019

1 reply · Share · Read 2 replies

Tweet Sarcasm Detection Using Deep Neural Network

Meishan Zhang, Yue Zhang, Guohong Fu

Abstract

Sarcasm detection has been modeled as a binary document classification task, with rich features being defined manually over input documents. Traditional models employ discrete manual features to address the task, with much research effort being devoted to the design of effective feature templates. We investigate the use of neural network for tweet sarcasm detection, and compare the effects of the continuous automatic features with discrete manual features. In particular, we use a bi-directional gated recurrent neural network to capture syntactic and semantic information over tweets locally, and a pooling neural network to extract contextual features automatically from history tweets. Results show that neural features give improved accuracies for sarcasm detection, with different error distributions compared with discrete manual features.

Link: <https://aclanthology.org/C16-1231/>

6. Coreference Resolution and Word Sense Disambiguation

- Coreference resolution is the **task of finding all expressions that refer to the same entity** in a text. In determining opinion targets (e.g. Galaxy Fold 3 phone), coreference resolution plays a major role because the opinion targets are **frequently mentioned not in the same sentences** where sentiments are expressed
- Word-sense disambiguation is the process of **identifying the word sense of a word** when the word appears in a text sequence with one or more context words. This is especially tough in analyzing social media opinions, since the **context words are often implicit**

0 Paul Allen was born on January 21 , 1953 , in Seattle , Washington . 0 Allen attended 3 Lakeside School , where 0 he befriended 1 Bill Gates . 2 0 Paul and 1 Bill used a teletype terminal at 3 2 their high school , Lakeside , to develop 2 their programming skills on several time - sharing computer systems .

Coreference Resolution Example

Source: <https://towardsdatascience.com/most-popular-coreference-resolution-frameworks-574ba8a8cc2d>

id	sentence	top 10 predictions from LSTM	sense
1	Employee compensation is offered in the form of cash and/or stock.	cash, stock, equity, shares, loans, bonus, benefits, awards, equivalents, deposits	sense#1
2	The stock would be redeemed in five years, subject to terms of the company's debt.	bonds, debt, notes, shares, stock, balance, securities, rest, Notes, debentures	
3	These stores sell excess stock or factory overruns .	inventory, goods, parts, sales, inventories, capacity, products, oil, items, fuel	sense#2
4	Our soups are cooked with vegan stock and seasonal vegetables.	foods, food, vegetables, meats, recipes, cheese, meat, chicken, pasta, milk	sense#3
query	In addition, they will receive stock in the reorganized company, which will be named Ranger Industries Inc.	shares, positions, equity, jobs, awards, representation, stock, investments, roles, funds	?

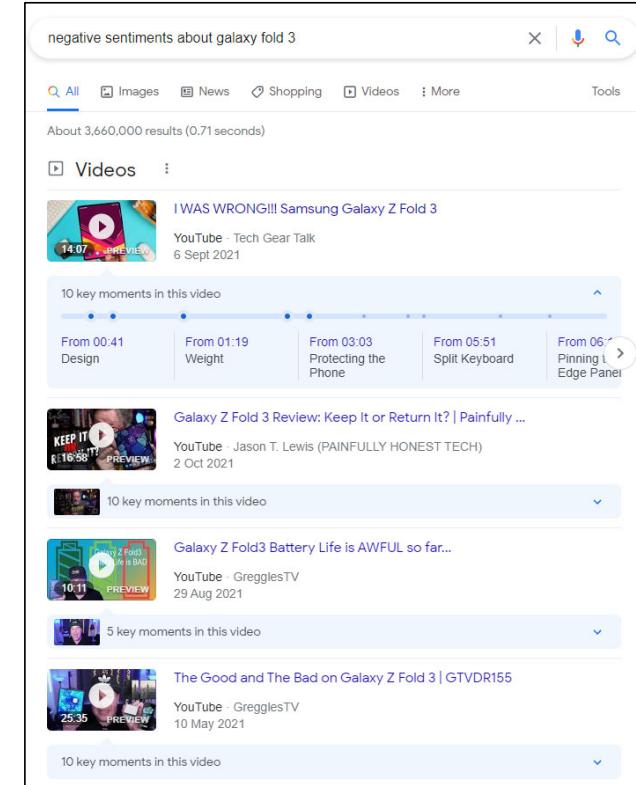
Table 1: Top predictions of 'stock' in 5 sentences of different word senses

Word Sense Disambiguation Example

Source: <https://gofishdigital.com/blog/word-sense-disambiguation/>

7. Opinion Search and Retrieval

- Frequently, we are interested to search and retrieve opinions about an entity or topic, similar to how we do Google or Bing searches. However, this turns out to be not so straightforward
- Similar to a general web search, an **opinion search** needs to perform two main tasks:
 1. Retrieve documents or sentences relevant to the user query
 2. Rank the retrieved documents or sentences
- However, **opinion retrieval** involves two subtasks:
 1. Find documents or sentences that are relevant to the query
 2. Determine whether they express opinions on the entity and/or entity aspect and whether those opinions are positive or negative
- Traditional web search engines rank web pages based on **authority and relevance scores**, but not based on **sentiment scores**. Thus, the top results do not capture the **natural distribution** of positive and negative sentiments of the whole population in a summarized form



The screenshot shows a search results page for "negative sentiments about galaxy fold 3". The results are filtered to show only Videos. The first result is a video titled "I WAS WRONG!!! Samsung Galaxy Z Fold 3" by "Tech Gear Talk" from 6 Sept 2021. Below it is a video titled "Galaxy Z Fold 3 Review: Keep It or Return It? | Painfully ..." by "PAINFULLY HONEST TECH" from 2 Oct 2021. The third result is a video titled "Galaxy Z Fold3 Battery Life is AWFUL so far..." by "GriegglesTV" from 29 Aug 2021. Each video thumbnail includes a timestamp and a preview image.

8. Opinion Spam Detection

- Opinions in reviews or social feeds are increasingly used by individuals and organizations to make decisions. Positive opinions often mean profit for businesses, and this provides strong incentives for imposters to **post fake reviews to promote or to discredit target products or services.**
- Such individuals are called **opinion spammers** and their activities are called **opinion spamming**
- There are two common approaches to detect opinion spam detection:
 - Burst Detection Detection – Looks for “dense areas” in the arrival pattern of reviews
 - Review Spammer Detection - Reviewer modeling to infer the real identities of reviewers

PERSONAL TECHNOLOGY: NICOLE NGUYEN

How to Spot Fake Reviews and Shady Ratings on Amazon

Sort by most recent reviews, check if older reviews were for completely different products and search for red flags such as 'gift' or 'free'

By Nicole Nguyen Updated July 20, 2022 3:48 pm ET

PRINT A TEXT

Listen to article (2 minutes)

Amazon AMZN 5.37%  is suing the administrators of more than 10,000 Facebook groups it says [coordinated fake reviews](#) on the shopping giant's platform. While Amazon didn't name the admins, the company did identify one group, called "Amazon Product Review," which it said had more than 43,000 members.

Source: <https://www.wsj.com/articles/how-to-spot-fake-reviews-and-shady-ratings-on-amazon-11658333513>

Fake Reviews Online Might Make You Spend More on Shittier Products

A new study found that fake reviews might get you to spend an extra 12 cents per dollar.

By Kevin Hurler | Friday 12:05AM | Comments (2)



Source: <https://gizmodo.com/reviews-amazon-online-shopping-1849318757>

Ref: Chapter 9, Pozzi, F., Fersini, E., Messina, E., & Liu, B. (2016). Sentiment analysis in social networks. Morgan Kaufmann.

3 General Approaches to Sentiment Analysis

Rule Based

- Human crafted rules to identify subjectivity, polarity or the subject of an opinion
- NLP techniques include: Stemming, tokenization, part-of-speech and parsing
- **Lexicons** (list of words and expressions)

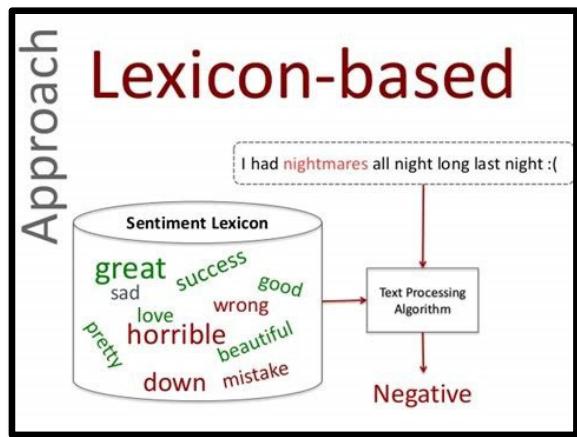
ML-based

- Systems rely on machine learning techniques to learn from data
- Key steps involved are: pre-processing & feature generation, feature extraction, training and prediction

Hybrid

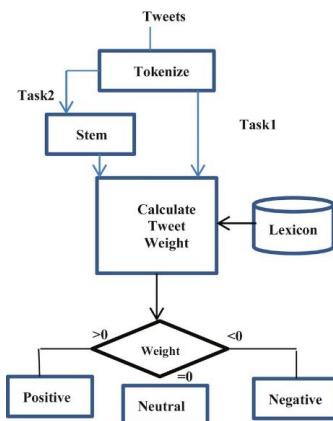
- Hybrid systems combine the desirable elements of rule-based and ML-based techniques into one system.
- One huge benefit of these systems is that results are often more accurate.

1. Lexicon Based Approach Example



I	:	0
Had	:	0
Nightmares	:	-3
All	:	0
Night	:	0

Long	:	0
Last	:	0
Night	:	0
: (:	-2



Total Score : -5

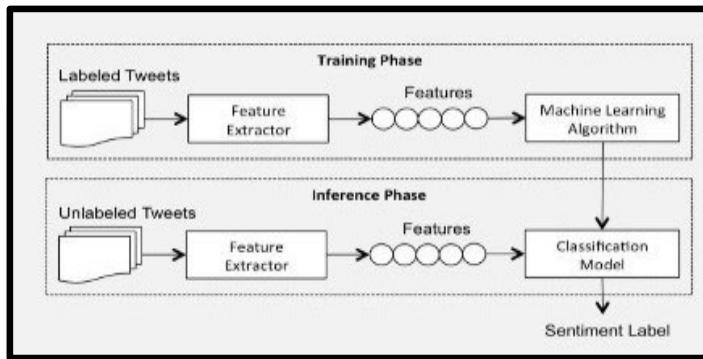
Issues with lexicon based systems

- Lexicon-based systems can be very naive since they don't take into account how words are combined in a sequence.
- They often require fine-tuning and maintenance.

Types of Lexicon Dictionary

- Some lexicon dictionary handle emoticons or emoji's
- Some can handle negations
- Some can handle intensity, e.g. booster words, capitalization & punctuations

2. ML-based Approach Example



Step 1. Raw Input Data Processing

Source: Text, web, pdf, images (OCR)

Step 2. Data Pre-Processing

HTML cleaning, Tokenization, Stemming/Lemmatization, Negation handling, spelling correction, slags removal etc.

Step 3. Feature Extraction

Bag of words, bag of N-grams, TF-IDF, Word2Vec or Glove (Word embedding's), Transformers-based embeddings

Step 4. Model Training

Supervised / un-supervised or self supervised methods

SOTA Prebuild models

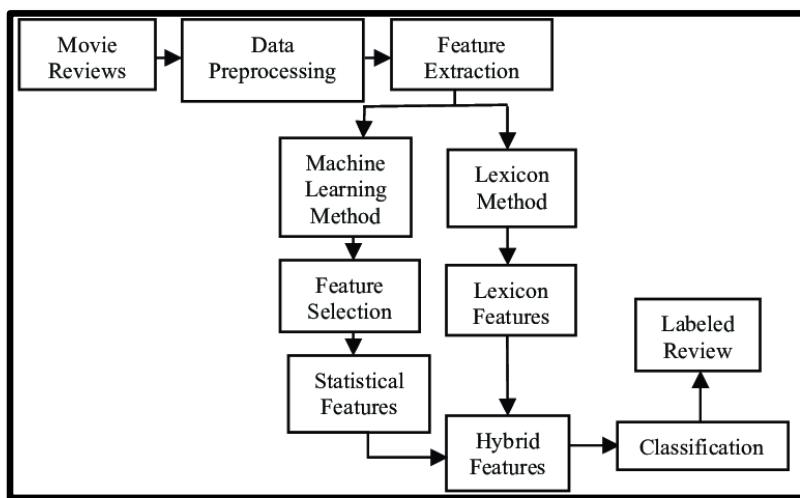
Improvising the SOTA models as per domain or use-case

Step 5. Prediction

Scoring methods (Accuracy, Recall, F1 score etc)

User inputs for unlabeled data

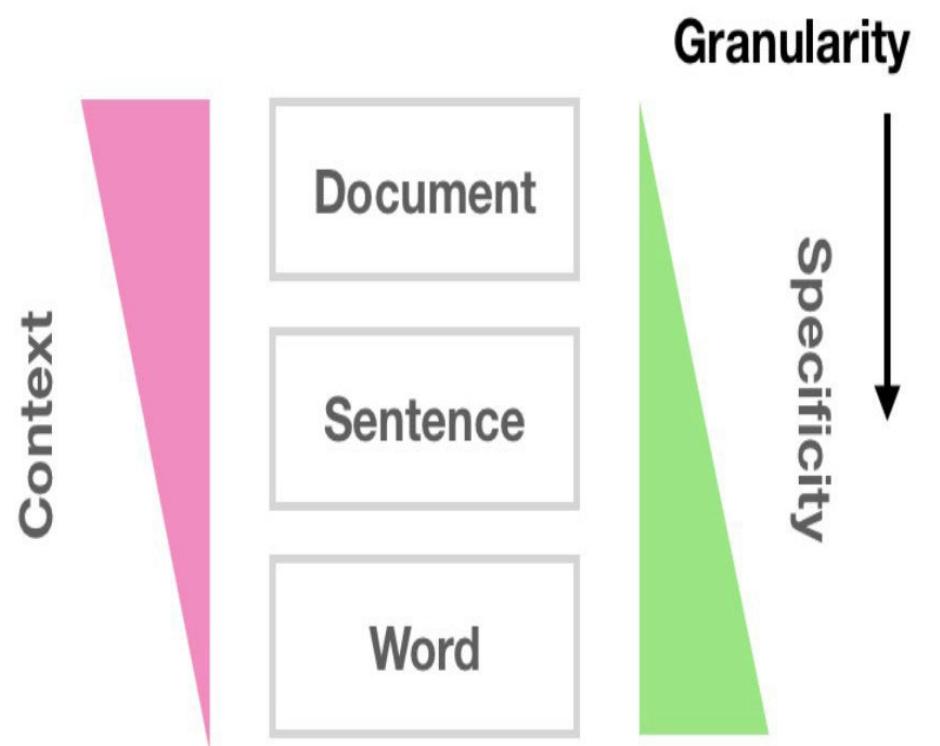
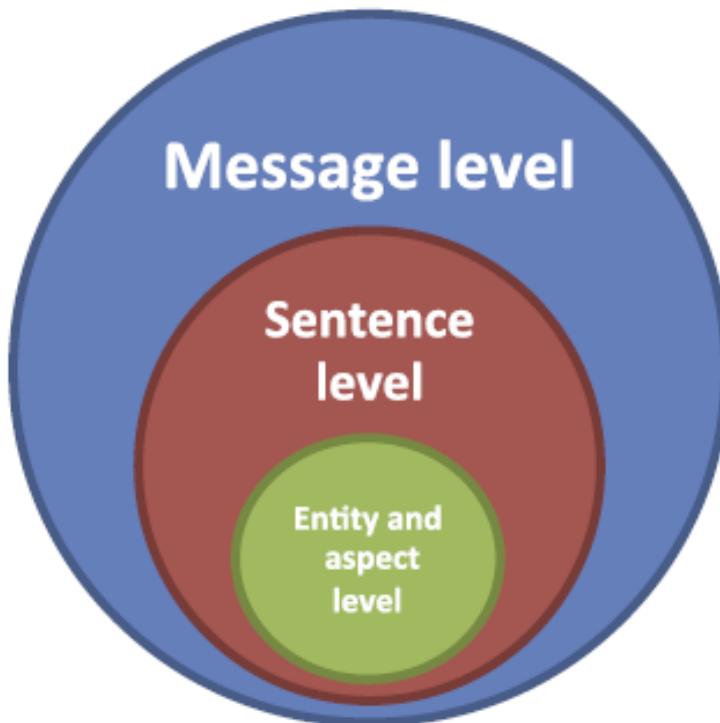
3. Hybrid Approach Example



Potential Benefits

- Combines the best of rule-based and ML approaches
- Domain-specific lexicons and rules can be introduced
- This approach may work better in situation where a large set of training data is unavailable
- Brings more flexibility for domain experts to intervene and handle complex NLP tasks using rule-based & updated lexicon dictionary
- Towards the idea of collaborative intelligence between computer and human

3 Levels of Sentiment Analysis



One loses context at higher granularity and specificity in lower granularity.

3 Levels of Sentiment Analysis

1. **Message level:** The aim is to classify the polarity of a whole opinionated message. For example, given a phone review, the system determines whether the review expresses an overall positive, negative, or neutral opinion about the phone. The assumption is that the entire message expresses **only one opinion** about the phone.
2. **Sentence level:** The aim is to determine the polarity of each sentence contained in a text message. The assumption is that each sentence, in a given message, denotes **only one opinion** about the phone.
3. **Entity and aspect level:** Performs a **finer-grained analysis** than message and sentence level. It is based on the idea that an opinion consists of a sentiment and a target (of opinion). For example, the sentence “The iPhone is very good, but they still need to work on battery life and security issues” evaluates three aspects: iPhone (positive), battery life (negative), and security (negative).

Activity: Sentiment Analysis about Bard from YouTube



Blunder by Google's AI Chatbot 'Bard', Alphabet shares lose \$100 Billion | World Business Watch |

33K views • 18 hours ago

WION

Alphabet Inc lost \$100 billion in market value on Wednesday after its new chatbot shared inaccurate information in a promotional ...

New

WION - Alphabet Inc lost \$100 billion in market value on Wednesday after its new chatbot shared inaccurate information in a promotional video and a company event failed to dazzle, feeding worries that the Google parent is losing ground to rival Microsoft Corp.

In your group, based on what you have learnt so far, create a one-page action plan to summarize the viewers' sentiment about Bard versus ChatGPT (and not the companies)

Source: <https://www.youtube.com/watch?v=ES-9tszg3OY&t=9s>

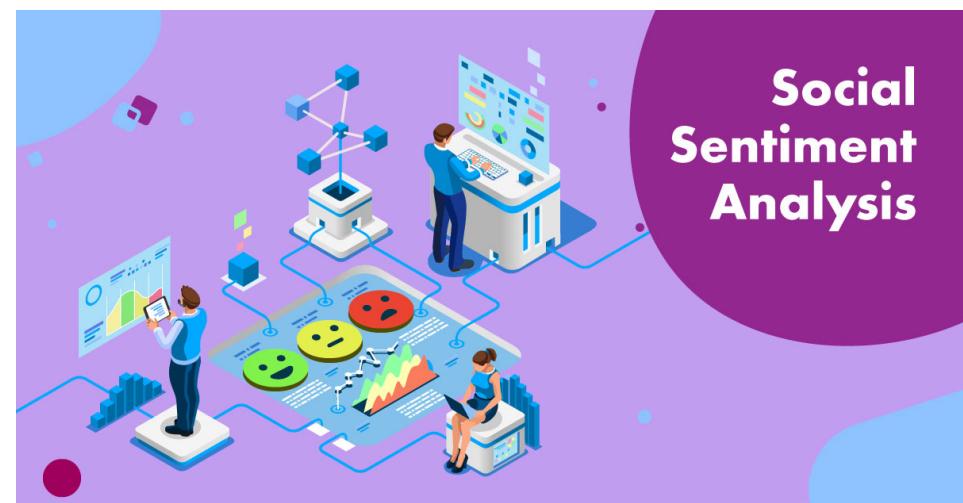


Part 2. Applications of Sentiment Analysis

Applications of Sentiment Analysis

Some main application areas

1. Customer Feedback and Reviews
2. Social Media Monitoring
3. Financial Prediction
4. Fraud Detection



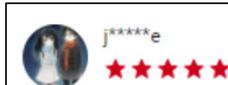
Source: <https://netbasequid.com/blog/what-is-social-sentiment-analysis/>

1. Customer Feedback and Reviews

- Voice of Customer (VoC)
- To improve customer satisfaction, enhance product development, get insights for competitive advantage, etc
 - **Identify positive sentiment:** Understand which products, services, or aspects of your business customers love.
 - **Uncover negative sentiment:** Pinpoint areas for improvement based on customer concerns and frustrations.
 - **Gauge overall satisfaction:** Get a holistic understanding of how customers feel about your brand.
 - **Track sentiment over time:** Monitor changes in customer sentiment over time to measure the impact of changes or initiatives.



Customer Reviews



j*****e

2022-07-20 09:22 | Variation: Band 7 - Pink

Performance: nice

Best Feature(s): good

Value For Money: yup

Bought it at sale price \$58 deduct \$5 off voucher i only pay \$53 nice.. frist time buy hua wei watch, this watch is easy to use and the qr from the watch didnt know need scan from wechat 😅 took awhile to set up but all are good now still trying to explore the function :)

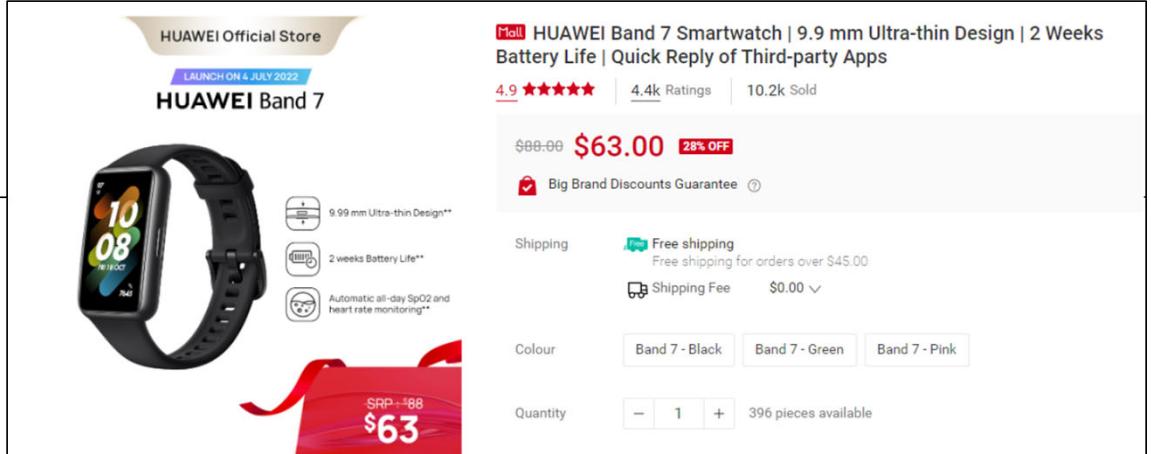


g*****n

★★★★★

2021-12-03 23:48 | Variation: Sakura Pink

I love it .. but mine doesn't come with extra straps even though i ordered on the 30 November still got promo free gift .. but when i complaints to the seller they blamed me they said group purchase single purchase different 😞 😞 so bad respond and no responsibility



HUAWEI Official Store
LAUNCH ON 4 JULY 2022
HUAWEI Band 7

HUAWEI Band 7 Smartwatch | 9.9 mm Ultra-thin Design | 2 Weeks Battery Life | Quick Reply of Third-party Apps

4.9 ★★★★★ 4.4k Ratings | 10.2k Sold

\$88.00 **\$63.00** 28% OFF

Big Brand Discounts Guarantee

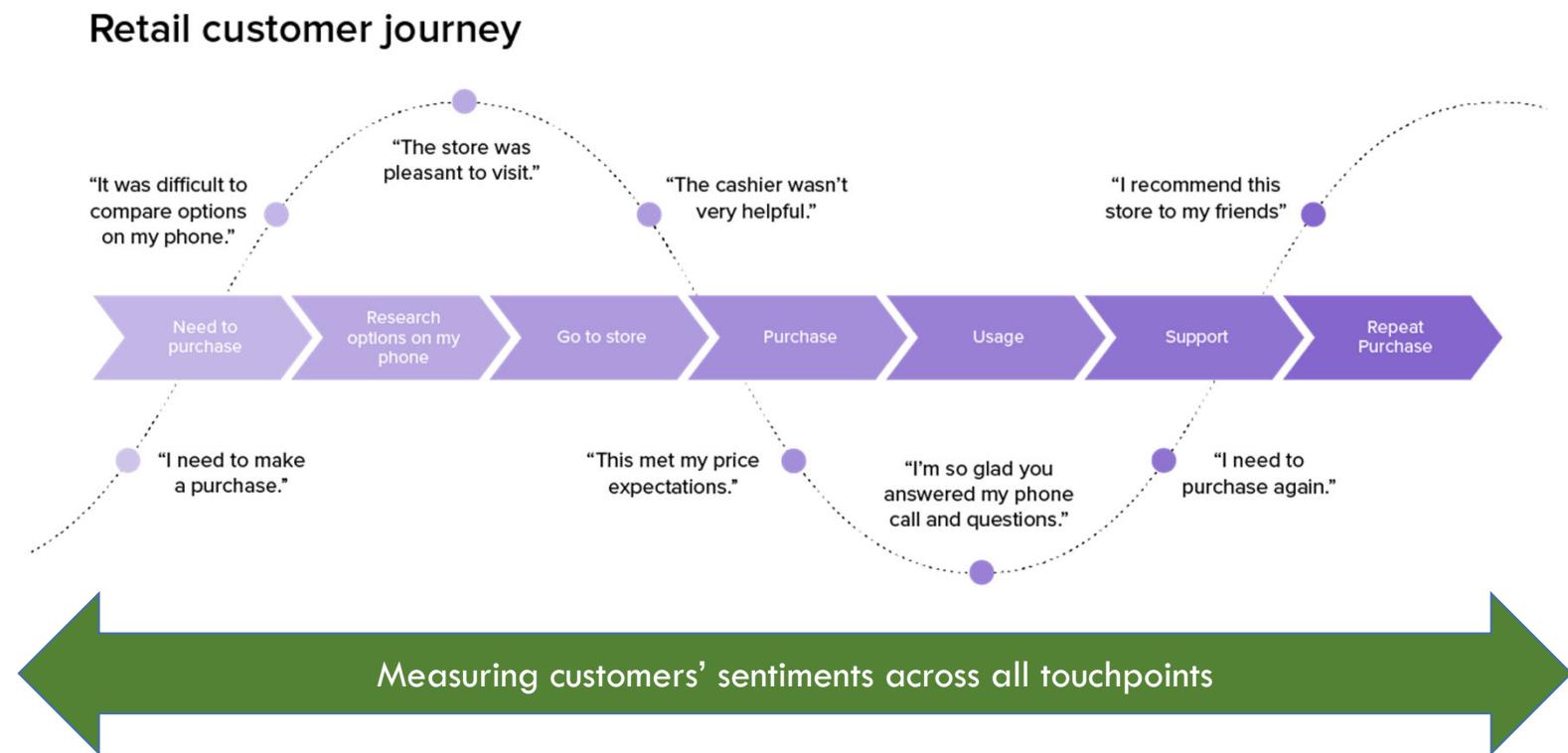
Shipping: Free shipping (Free shipping for orders over \$45.00)
Shipping Fee: \$0.00

Colour: Band 7 - Black, Band 7 - Green, Band 7 - Pink

Quantity: - 1 + 396 pieces available

Source: <https://shopee.sg/HUAWEI-Band-7-Smartwatch-9.9-mm-Ultra-thin-Design-2-Weeks-Battery-Life-Quick-Reply-of-Third-party-Apps-i.181050132.2936331613>

Customer Journey Map – Customer Experience (CX)



Source: Delighted by Qualtrics. Link: <https://delighted.com/blog/guide-to-customer-journey-mapping>

© National University of Singapore. All Rights Reserved.

1. Customer Feedback and Reviews

Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews

Publisher: IEEE [Cite This](#) [PDF](#)

Pankaj ; Prashant Pandey ; Muskan ; Nitasha Soni [All Authors](#)

8 Paper Citations 885 Full Text Views

Abstract

Abstract:

Sentiment analysis is one of the fastest spreading research areas in computer science, making it challenging to keep track of all the activities in the area. We present a customer feedback reviews on product, where we utilize opinion mining, text mining and sentiments, which has affected the surrounded world by changing their opinion on a specific product. Data used in this study are online product reviews collected from <http://Amazon.com>. We performed a comparative sentiment analysis of retrieved reviews. This research paper provides you with sentimental analysis of various smart phone opinions on smart phones dividing them Positive, Negative and Neutral Behaviour.

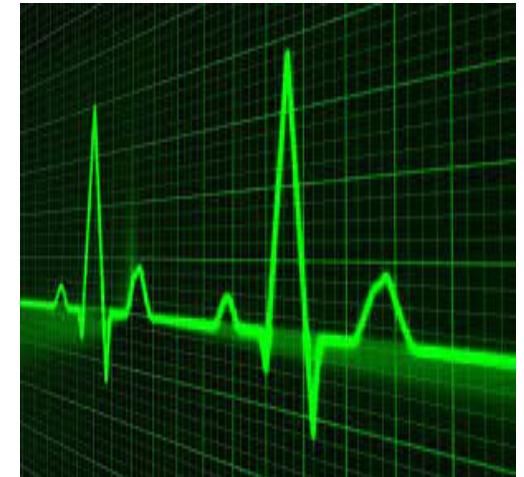
Published in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)

Date of Conference: 14-16 February 2019 **INSPEC Accession Number:** 19046522

Link: <https://ieeexplore.ieee.org/abstract/document/8862258>

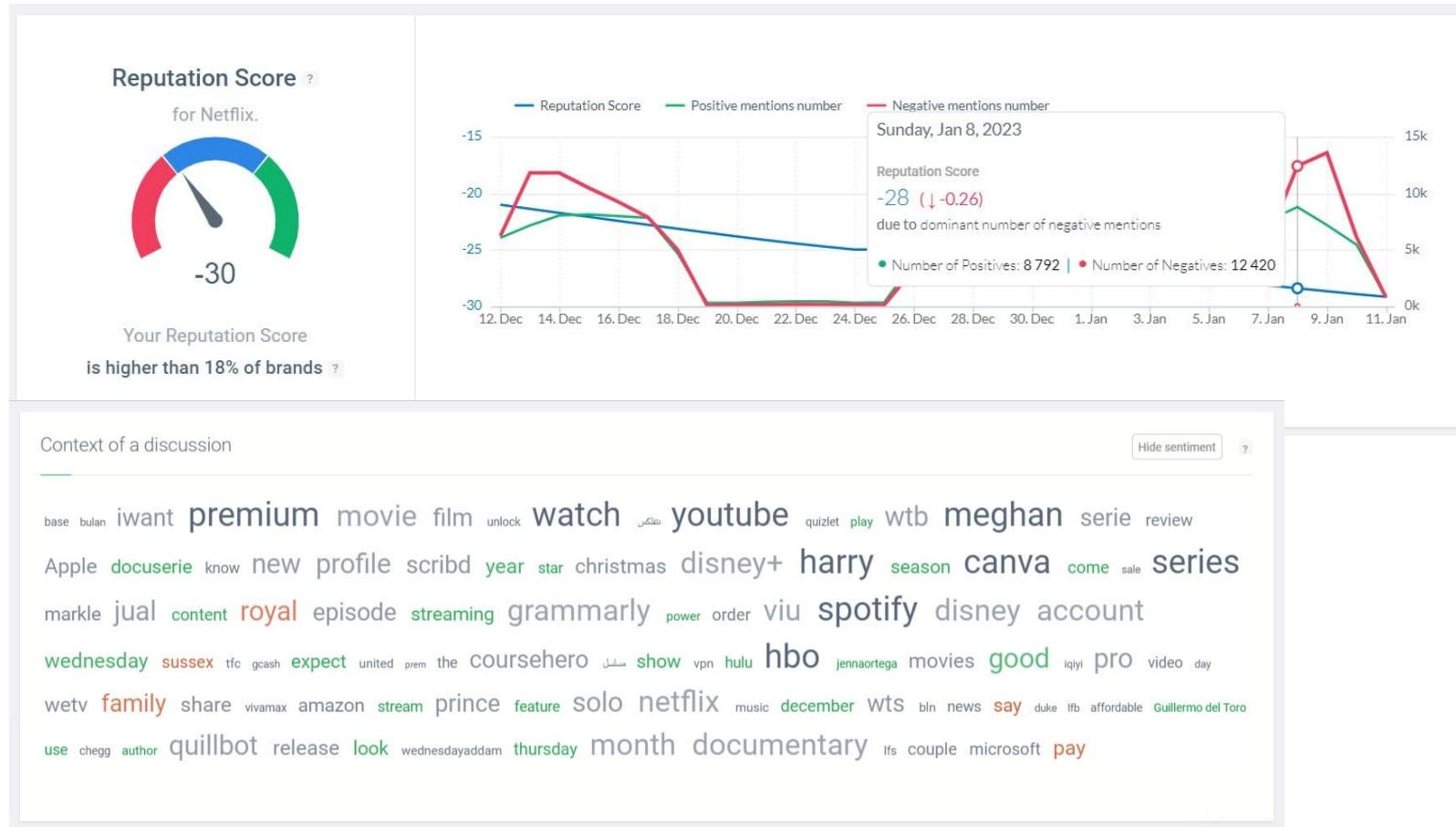
2. Social Media Monitoring

- Social listening, for brand monitoring, public opinion monitoring
 - Track brand mentions: Stay ahead of the curve by monitoring what people are saying about the brand on social media.
 - Uncover sentiment: Go beyond just mentions – understand the emotional tone and overall feeling behind the conversations.
 - Identify trends: Spot emerging issues, positive feedback, and competitor activity to create informed brand strategy.
 - Engage proactively: Respond to concerns, address criticism, and build stronger customer relationships.
 - Measure impact: Track sentiment over time to see the effectiveness of your social media campaigns.



"It takes many good deeds to build a reputation, and only one bad one to lose it." – Benjamin Franklin

Online Reputation Score





Zignal Labs – Social Listening Tools

- Sentiment Analysis pipeline using Amazon SageMaker

COVID-19 – Public Sentiment using Reddit Data

General public's attitude toward governments implementing digital contact tracing to curb COVID-19 – a study based on natural language processing

Praveen S.V., Rajesh Ittamalla ▾

International Journal of Pervasive Computing and Communications

ISSN: 1742-7371

Article publication date: 30 September 2020 

Issue publication date: 25 November 2022

DOWNLOADS ALTMETRICS



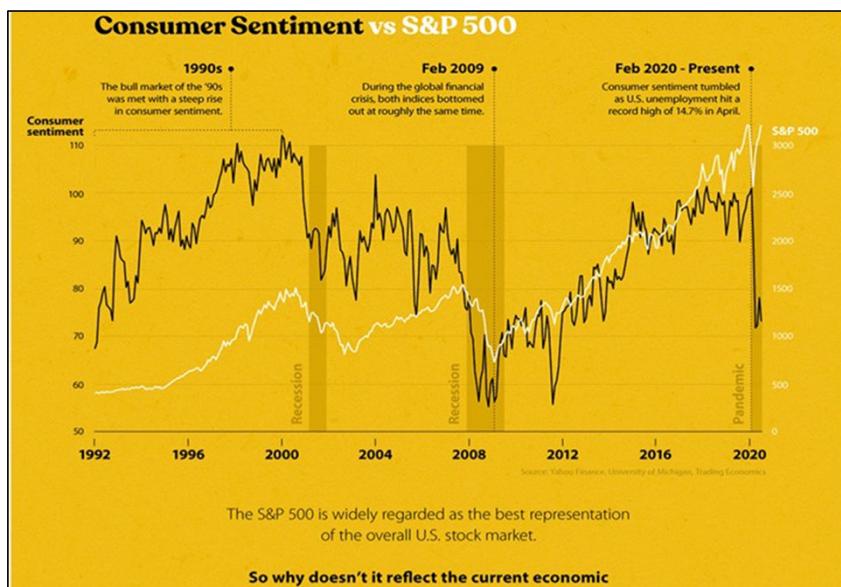
Problem: Governments worldwide are taking various measures to prevent the spreading of COVID virus. One such effort is digital contact tracing. However, many critics view digital contact tracing as an attempt of the government to control people and a fundamental breach of privacy.

Method: Textblob to generate sentiment scores and Latent Dirichlet Allocation to derive a set of topics from the Reddit text posts

Source: SV, P., & Ittamalla, R. (2022). General public's attitude toward governments implementing digital contact tracing to curb COVID-19 – a study based on natural language processing. Link: <https://www.sciencedirect.com/science/article/pii/S2667096821000148>

3. Financial Prediction

- Sentiment drives the stock market. It changes people's consumption habits which in turn drives their investment and savings tendencies



Crypto 'Fear and Greed' Gauge Improves a Bit After 60% Selloff

- Measure climbing toward 'fear' from 'extreme fear' level
- Bitcoin briefly breaks above \$20,000 before coming back down

Source: <https://www.bloomberg.com/news/articles/2022-07-05/crypto-fear-and-greed-gauge-improves-a-bit-after-60-selloff>

BONDS

10-year yield slides to two-week low as weak data, ECB move dampen sentiment

PUBLISHED FRI, JUL 22 2022 5:35 AM EDT | UPDATED FRI, JUL 22 2022 4:18 PM EDT

Source: <https://www.visualcapitalist.com/understanding-the-disconnect-between-consumers-and-the-stock-market/>
<https://www.cnbc.com/2022/07/22/10-year-yield-slides-to-two-week-low-as-weak-data/ecb-move-dampen-sentiment.html>

Financial prediction with sentiment analysis

- Enhanced by sentiment, but not sentiment alone
 - **Analyze public sentiment:** Gauge market sentiment by analyzing news articles, social media posts, and financial reports.
 - **Identify trends and patterns:** Recognize emotional shifts that might precede market movements.
 - **Predict asset performance :** Combine sentiment analysis with other data sources to create more informed predictions.

Analyzing Stock Market Movements Using Twitter Sentiment Analysis

Tushar Rao
NSIT, Delhi, India
Email: rao.tushar@nsitonline.in

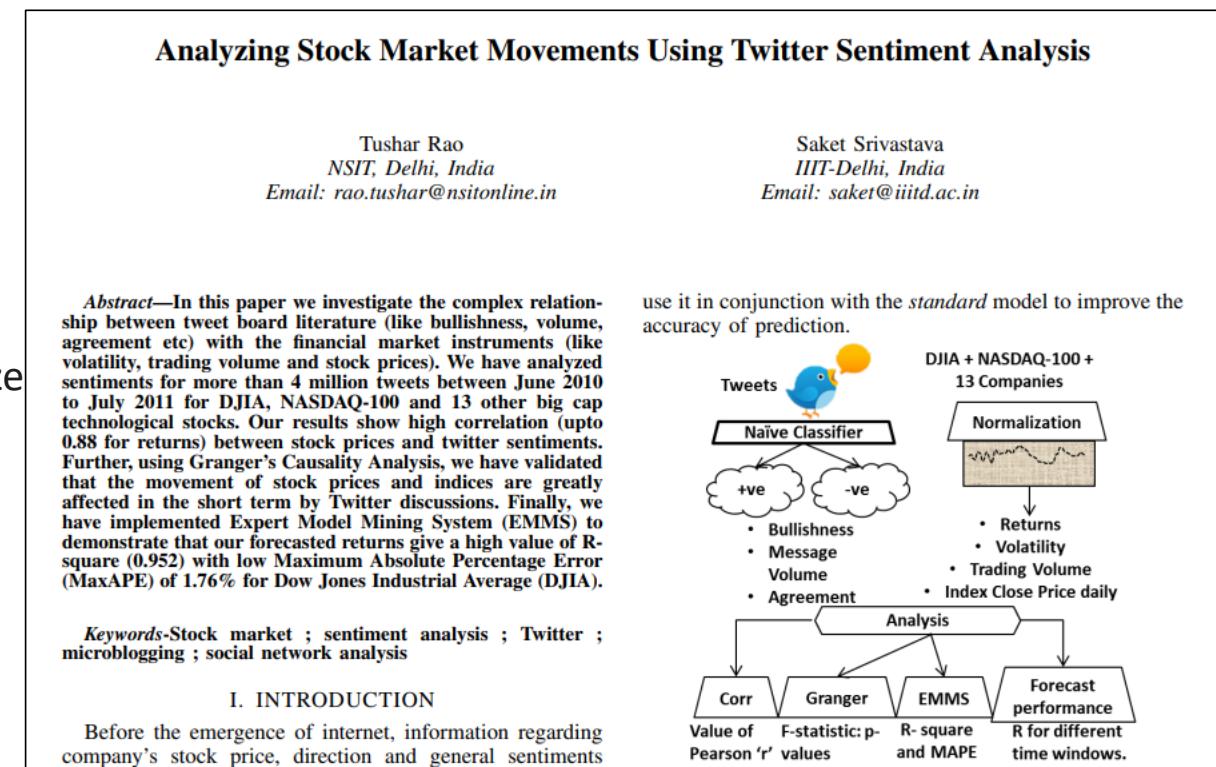
Saket Srivastava
IIT-Delhi, India
Email: saket@iitd.ac.in

Abstract—In this paper we investigate the complex relationship between tweet board literature (like bullishness, volume, agreement etc) with the financial market instruments (like volatility, trading volume and stock prices). We have analyzed sentiments for more than 4 million tweets between June 2010 to July 2011 for DJIA, NASDAQ-100 and 13 other big cap technological stocks. Our results show high correlation (upto 0.88 for returns) between stock prices and twitter sentiments. Further, using Granger's Causality Analysis, we have validated that the movement of stock prices and indices are greatly affected in the short term by Twitter discussions. Finally, we have implemented Expert Model Mining System (EMMS) to demonstrate that our forecasted returns give a high value of R-square (0.952) with low Maximum Absolute Percentage Error (MaxAPE) of 1.76% for Dow Jones Industrial Average (DJIA).

Keywords-Stock market ; sentiment analysis ; Twitter ; microblogging ; social network analysis

I. INTRODUCTION

Before the emergence of internet, information regarding company's stock price, direction and general sentiments



```

graph TD
    Tweets[Tweets] --> NaiveClassifier[Naïve Classifier]
    NaiveClassifier --> Plus[+ve]
    NaiveClassifier --> Minus[-ve]
    Plus --> Bullishness[Bullishness]
    Plus --> MessageVolume[Message Volume]
    Plus --> Agreement[Agreement]
    Minus --> Bullishness
    Minus --> MessageVolume
    Minus --> Agreement
    Bullishness --> Returns[Returns]
    MessageVolume --> Returns
    Agreement --> Returns
    Returns --> Normalization[Normalization]
    Normalization --> DJIA[DJIA + NASDAQ-100 + 13 Companies]
    DJIA --> Analysis[Analysis]
    Analysis --> Corr[Corr]
    Analysis --> Granger[Granger]
    Analysis --> EMMS[EMMS]
    Corr --> ValuePearsonR[Value of Pearson 'r' values]
    Granger --> FstatisticP[F-statistic: p-values]
    EMMS --> RSquareMAPE[R-square and MAPE]
    ValuePearsonR --> ForecastPerformance[Forecast performance R for different time windows.]
    FstatisticP --> ForecastPerformance
    RSquareMAPE --> ForecastPerformance
  
```

The flowchart illustrates the methodology. It starts with 'Tweets' leading to a 'Naïve Classifier' which outputs '+ve' and '-ve' sentiment categories. These categories lead to 'Bullishness', 'Message Volume', and 'Agreement'. These three factors are combined and used in a 'Normalization' step, which then applies to 'DJIA + NASDAQ-100 + 13 Companies'. This leads to an 'Analysis' step, which includes 'Corr', 'Granger', and 'EMMS'. The results from 'Analysis' are then used to calculate 'Value of Pearson 'r' values', 'F-statistic: p-values', and 'R-square and MAPE', which finally lead to 'Forecast performance R for different time windows.'

Link: <http://eprints.lincoln.ac.uk/id/eprint/11274/1/ASONAM%202012.pdf>

Project: Gold Price Trend Prediction

Reuters Eikon Interface

MARKETS NEWS CHARTING COMPANY RESEARCH FILINGS & EVENTS PORTFOLIOS & PORTFOLIOS

NEWS MONITOR v2 Click here to try the new version of News Monitor

Topics Q XAU= AND English AND Significant News Enter RIC, Topic, Portfolio...

Most Recent All

RTRS 04:34:53 AM GLOBAL MARKETS-Stocks slide, dollar gains on hot U.S. inflation data
 RTRS 04:09:20 AM LIVE MARKETS-Wall Street ends week with a steep drop after CPI scare
 GURU 04:08:42 AM Gold intraday: the upside prevails as long as 1864 is support
 RTRS 03:03:36 AM GLOBAL MARKETS-Stocks slide, dollar gains on hot U.S. inflation data
 RTRS 03:01:28 AM PRECIOUS-Gold rebounds in roller coaster trade as economic fears grow
 RTRS 01:53:42 AM LIVE MARKETS-Oil: the new gold?
 RTRS 01:11:31 AM REFILE-LIVE MARKETS-Cresset's models suggest S&P 500 still needs a haircut
 RTRS 01:03:20 AM REFILE-LIVE MARKETS-Cresset's models suggest S&P 500 still needs a haircut
 RTRS 12:12:41 AM LIVE MARKETS-Semis drop as investors eye higher rates, PC slowdown

10 June 2022

RTRS 11:57:24 PM LIVE MARKETS-Banks raise Fed rate calls after hot CPI
 RTRS 11:39:32 PM LIVE MARKETS-Stoxx 600 ends the worst week in one month
 RTRS 11:09:29 PM GLOBAL MARKETS-Stocks slide, dollar gains on hot U.S. inflation data
 RTRS 11:04:39 PM LIVE MARKETS-Summertime blues: CPI scorches, consumer sentiment plunges to record low

2022 EBAC Project By: Xiaohong, Haoxin, Yansun, Chenchen & Youwen

Raw news data table

	Date	Headline	raw_news
0	2021-02-01 01:00:00	PRECIOUS-Silver surges to near 6-month peak on...	Feb 1 (Reuters) - Silver soared as much as 7.4...
1	2021-02-01 08:00:00	PRECIOUS-Retail mania propels silver prices to...	Physical silver demand more than doubled since...
2	2021-02-01 11:00:00	UPDATE 1-India cuts import tax on gold, indust...	Slashes gold, silver import duty to 7.5% from ...
3	2021-02-01 13:00:00	GRAPHIC-Silver fund inflows surge in January o...	By Paturaja Murugaboopathy Feb 1 (Reuters) - ...
4	2021-02-01 14:00:00	PRECIOUS-Silver scales 8-year peak as investor...	Excessive price rise can hamper physical deman...

Price data table

	Date	HIGH	LOW	OPEN	CLOSE	COUNT
0	2021-01-25 00:00:00	1859.0900	1853.4900	1854.2900	1856.6100	1480
1	2021-01-25 01:00:00	1859.6969	1855.0419	1856.6600	1858.0819	1550
2	2021-01-25 02:00:00	1859.6334	1852.4263	1858.0425	1854.6100	2620
3	2021-01-25 03:00:00	1855.8600	1853.3058	1854.4380	1854.1600	1925
4	2021-01-25 04:00:00	1855.4700	1852.9384	1853.9886	1853.6100	1762

4. Fraud Detection

- Identify suspicious language by analyzing emails, messages, or social media posts for words and phrases commonly associated with fraud.
- Analyze customer reviews to identify fake reviews that praise a product or service unrealistically to boost its credibility.
- Monitor social media to track mentions of specific brands or products to identify potential scams or phishing attempts.
- Enhance fraud scoring by integrating sentiment analysis with other fraud detection models to improve their accuracy and identify more complex patterns.
- Investigate suspicious activity by helping investigators understand the motivations and methods of fraudsters by analyzing their communication patterns and sentiment.



Fraud Detection for Cryptocurrency Scheme

KaRuNa: A Blockchain-based Sentiment Analysis Framework for Fraud Cryptocurrency Schemes

Patel Nikunjkumar Sureshbhai*, Pronaya Bhattacharya†, Sudeep Tanwar‡

*†‡Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India
 Email: *17bce083@nirmauni.ac.in, †pronoya.bhattacharya@nirmauni.ac.in, ‡sudeep.tanwar@nirmauni.ac.in

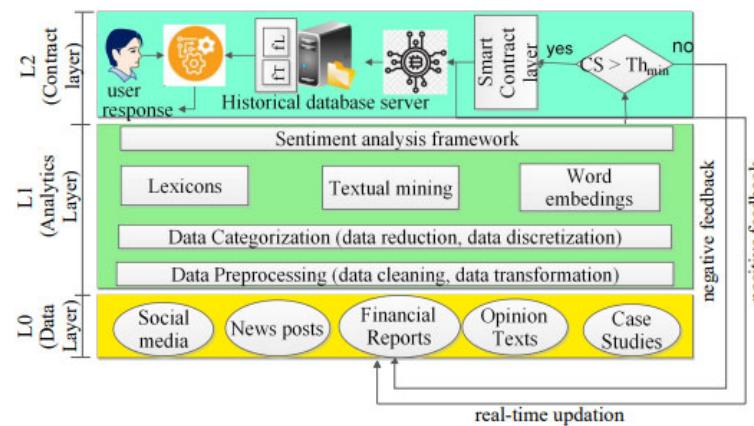


Fig. 1: *KaRuNa: System Model*

Problem: Prospective investors (PI) are susceptible to monetary losses in the open market due to factors such as the pseudo-anonymity of cryptocurrency transactions, non-transparency of money generation and distribution among peers, and untraceable scams.

Solution: KaRuNa – a framework to help PI evaluate investment risks accurately and build trust in the cryptocurrency market

Link: Sureshbhai, P. N., Bhattacharya, P., & Tanwar, S. (2020, June). KaRuNa: A blockchain-based sentiment analysis framework for fraud cryptocurrency schemes. Link: <https://nirmawebiste.s3.ap-south-1.amazonaws.com/wp-content/uploads/sites/22/2022/03/KaRuNa.pdf>

NLP for Fraud Detection in Derivatives Market

Objective



▪ Background

- Derivatives are becoming increasingly popular instruments for financial frauds.
- Current market surveillance system can't leverage textual information.

▪ Objective

Extract financial event analytics from textual data that can be combined with market & trading data, to detect frauds especially **illegal Insider Trading*** in the derivatives market.

* Trading based on material nonpublic information.

▪ Added value for regulator (vs. rule-based system)

- ✓ Increase alerts precision (Currently false alerts are overwhelming)
- ✓ Discover hidden suspicious trading patterns
- ✓ Provide organized contextual information for investigation

▪ Challenges

... 3

Market events info buried in tremendous amount of unstructured textual data

Overload of Data

NLP models that can "understand" and "analyze" the business news events

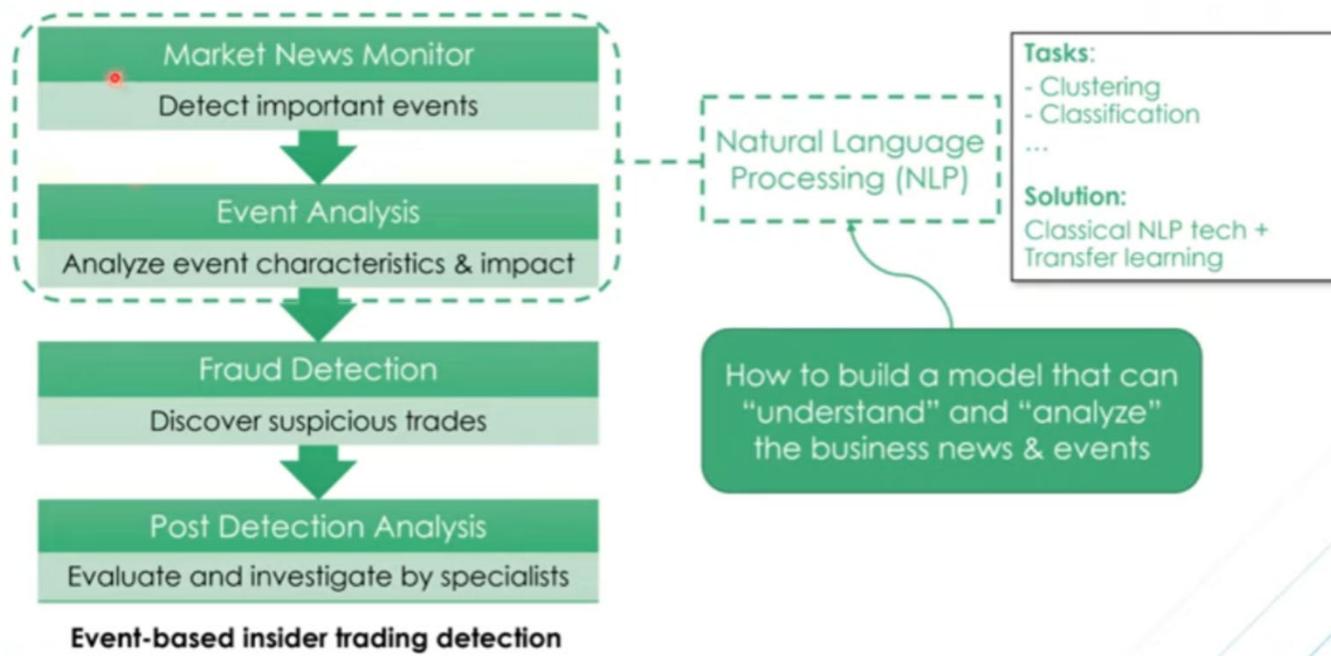
Scarcity of Insights

Scarcity of labeled (annotated) financial textual data

Source: Liu Pan, Natural Language Processing for fraud detection in derivatives market - 17 June 2021. Link: <https://www.youtube.com/watch?v=CK6L8Y5zxM>

Event-based Insider Trading Detection

Methodology



Source: Liu Pan, Natural Language Processing for fraud detection in derivatives market - 17 June 2021. Link: <https://www.youtube.com/watch?v=CK6L8Y5zxM>



www.iss.nus.edu.sg



www.facebook.com/ISS.NUS



twitter.com/ISSNUS



@iss.nus



www.linkedin.com/company/iss_nus



youtube.com/user/TheISSNUS/



EB5204 : NEW MEDIA AND SENTIMENT MINING

MODULE 2.2: LEXICON-BASED & SUPERVISED METHODS

Dr. Wang Aobo

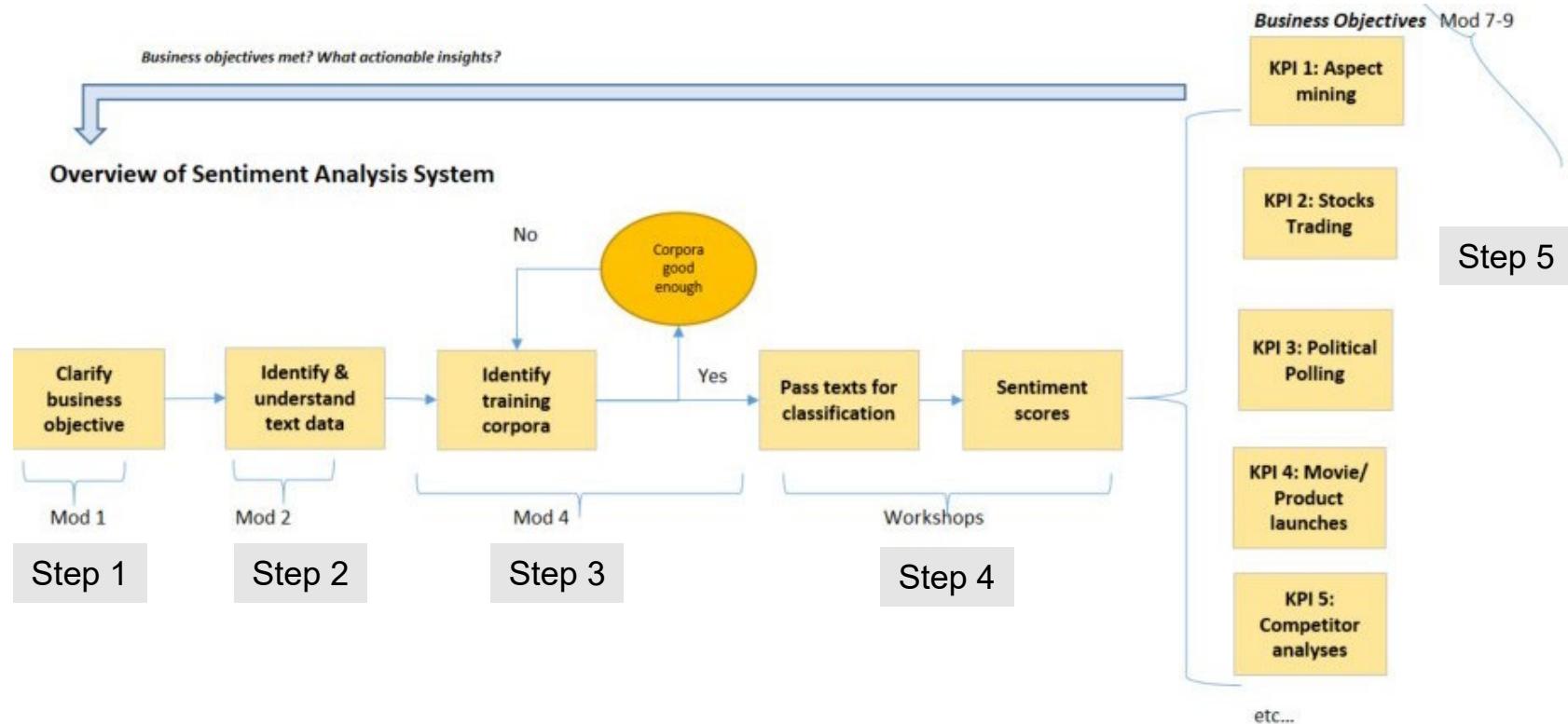
isswan@nus.edu.sg



Agenda

- Considerations for Sentiment Analysis System Design
- Lexicon-based approach for sentiment mining
- Training **data** set generation for sentiment mining
- **Feature Engineering** for sentiment mining
- Word Embedding Vectors as Features

0. Overview of Typical Sentiment Analysis System

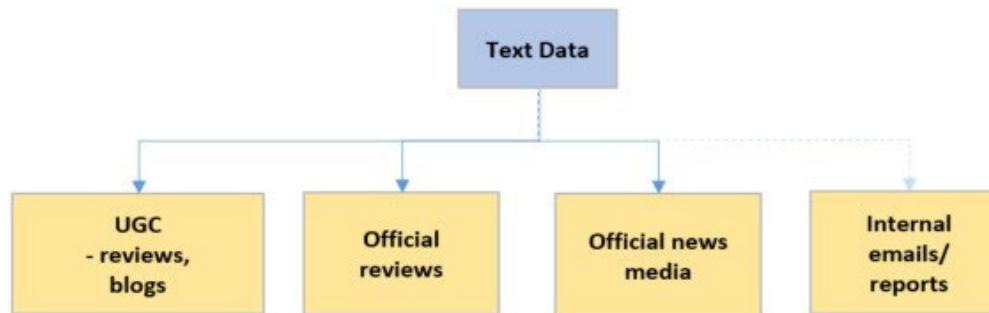


Step 1: Clarify Business Objectives

- In any business analytics system, the **business objectives need to be clearly defined.**
- Example of typical business objectives include:
 1. To reduce customer churn by focusing on their key complaints
 2. To anticipate stock market reactions based on public sentiments
 3. To detect fake news and alert the appropriate authority to take actions
 4. To understand the public sentiment towards a new public policy after public consultations

Step 2: Identify and Understand Text Data

- The next step is to identify the **data source** and **type of text data**.
- The pre-processing pipeline will be very much dependent on type of text data we are working with.
- Social media from media like WSJ, Reuters are usually well-structured and can be extracted relatively easily.



Step 3: Identifying Training Corpora

- Before training the sentiment analysis models, we need to ensure that the training corpora is of sufficiently high quality. **Garbage in garbage out** phenomenon is very real.
- There are two ways of improving the quality of the training corpora:
 1. Having **domain experts** to perform the data annotations, or a systematic data annotation scheme
 2. Perform tasks such as Subjectivity Classification and Sarcasm Detection to ensure correct computation of sentiment scores

Data Set	Description of Tweets	Quantity of Tweets			
		Overall	Positive	Negative	Neutral
Telco	Related to telecommunications company Telus' products, and services. Include general discussion of experiences, news, and specific events.	5281	20.9%	8.9%	70.2%
Pharma	Related to users' experiences with pharmaceutical drugs. Include mentions of adverse events, positive interactions, etc.	5009	15.6%	11.1%	73.3%
Security	Related to major security companies' products and services, including security incidents and new software releases and/or security patches.	5086	24%	11.1%	64.9%
Tech	Related to four major tech firms. Include discussion of companies' products, services, policies, and general user experiences.	3502	15.1%	16.9%	68.0%
Retail	Include discussion of a specific category of retail products (household paint) and user experiences related to those products.	3750	42.7%	9.0%	48.3%

Table 1: Evaluation Test Bed Overview

Source: <https://www.semanticscholar.org/paper/Benchmarking-Twitter-Sentiment-Analysis-Tools-Abbas-Hassan/d0a521c8cc0508f1003f3e1d1fbf49780d9062f7>

Step 4: Sentiment Classification and Scoring

- Different levels of sentiment classification (e.g document-level and sentence-level)
- Examples of Supervised sentiment classification methods
 - 1. Naïve Bayes methodology
 - 2. K Nearest Neighbour (kNN) Classification
 - 3. Maximum Entropy Method
 - 4. Support vector machine (SVM)
- Examples of Unsupervised sentiment classification methods
 - 1. Syntactic Patterns and Web Search
 - 2. Sentiment Lexicons



Evaluating Model Performance can be an issue!

Step 5: KPIs based on Business Objectives

- Key performance indicators (KPIs) refer to a set of quantifiable measurements used to gauge a business value of the system, and is related to the concept of Return on Investment (ROI) .
- Since resources (e.g. manpower, time, budget) are expended in the development and maintenance of such system, the ROI to the business must be clear
- For example, there must be **actionable insights** that can be distilled from the use of the system. These insights must also be well communicated to stakeholders via **Sentiment Visualization** (ex: dashboards)

1. Lexicon-based semi-supervised approach

Lexicon-based approach

- Requires a **sentiment lexicon** (patterns - words, phrases, idioms), composite expression, rules of opinions
- Essentially compute the sentiment polarity of a target in a sentence by using **sentiment aggregation** taking into account the **distances** of the sentiment expressions and the target entity/aspect.
- Also consider sentiment **shifters**, *but*-clauses, and the syntactic relationships of sentiment expressions and opinion targets.



- **Opinion Lexicon**: list of words, phrases, expressions used to express sentiment
 - **+ve terms**: good, excellent, cool, wow!, ...
 - **-ve terms**: stinks, bad, old-fashioned, yuck!,...
 - **Phrases**: “cost an arm and a leg”
 - **Slangs**: “referee kayu!”, “kiasu”
 - **Idioms**: “soft hearted”, “hard nosed”
 - There are flaws – note due to the context.
 - *“The bag is a little big for her”*
 - *“That’s pretty ugly”*



Creating the Opinion Lexicon

- **Combination** of manual, dictionary and corpus based approaches
 - **Manual**: by inspection
 - **Dictionary**: own source, WordNet, SenitWordNet for words
 - Often cannot get context dependent words/phrases
- The process is **iterative**
 - Knowing some sentiment words helps to find more
 - E.g., “The dress is **elegant** and **refined** but **provocative**”
 - If “elegant” is +ve, then “refined” is +ve; “provocative” is –ve

Apply Opinion Lexicon

“The phone’s call quality is not good, but its battery life is long.”

- Mark sentiment words and phrases

The phone’s call quality is not good [+1], but its battery life is long.

- Apply sentiment shifters

The phone’s call quality is not good [-1], but its battery life is long.

- Handle but-clauses

*The phone’s call quality is not good [-1], **but** its battery life is long [+1].*

- Aggregate opinions

- *Call quality – negative, battery life - positive*

A simple method as example

- Assumption: the target entities and aspects are known (as NER in a sentence, or in a specific review)
- Main steps (Ding, Liu and Yu, 2008):
 - Mark sentiment words and phrases
 - Apply sentiment Rules (such as negation words)
 - Handle *but*-clauses (*but, however, except for, except that, with the exception of*)
 - Indicating contrary, but not always ("x is great, but y is better")
 - Aggregate opinions
 - Across sentences and time, weighing by its importance.

Sentiment Rules – Sentiment Reverse

- Sentiment shifters
- Negation words like *not*, *never*, *none*, *nobody*, *nowhere*, *neither*, *cannot*...(identify with the adjective if less than 'n' words?)
 - “*The taste is not good.*”
- Modal auxiliary verbs (e.g., *would*, *should*, *could*, *might*, *must*, *ought*) may change sentiment orientation, but not always.
 - “*The service could be improved.*”
- Adverbs like *barely*, *hardly*
 - “*It hardly works.*”
- Words like *fail*, *omit*, *neglect*, etc.
 - “*The car failed to start.*”

POS + shifter => NEG

NEG + shifter => POS

Sentiment reverse- opinionated item

- Decreased (or removal, disappearance) and increased quantity of an opinionated item (NEG and POS, often nouns) can change its orientation.

“This drug reduced my pain significantly.”

“My pain disappeared after taking the drug.”

“The earphone can isolate noise.”

POS + decreased => NEG

NEG + decreased => POS

POS + increased => POS (intensification)

NEG + increased => NEG (intensification)

Sentiment composition rules

- The quantity or change of quantity of *potential positive or negative items* (PositivePolarityItem or NegativePolarityItem, like “*battery life*”(1), “*price*”(2), “*memory*”(3), etc.)

“*The camera memory went up a lot.*”

“*The battery life is short.*”

“*Sony reduced the price of the camera.*”

NPI + no/low/less/decreased => POS

NPI + large/larger/increased => NEG

PPI + no/low/less/decreased => NEG

PPI + large/larger/increased => POS

Sentiment conflicts

- When multiple sentiment words occur together
“*terribly(-) good(+)*”, “*pretty (+) ugly (-)*”
- Resolution is to **rank** the constituents on the basis of relative **weights** (importance)
- E.g. if two sentiment words have the opposite polarity
 - if ADV+**ADJ**, the first adjective gets higher weight
 - if **V**+OBJ, the verb polarity prevails
 - If in two clauses connected by “but”, the one after the connector is dominant.



Other rules

- **Intensification** rule: “*extremely happy*” (for granular scale of sentiment)
- Consume resource and produce waste:
“*This computer uses a lot of electricity.*”
- Add other rule for deviation from the norm or a desired value range
“*After taking the drug, my blood pressure went up to 200.*”
“*I got 45 marks for my score.*”



Sentiment words in non-opinion contexts

Sometimes need to be careful with such words like...

- Entity names containing sentiment words
“Best Denki”
- Function names containing sentiment words
“fast forward” “I’m feeling lucky”
- Greetings and good wishes
“Good morning” “hope you get well soon”
- Author’s self-description
“I know Samsung products very well.”

Senses of sentiment words

Words with different senses can have different sentiment. (SentiWordNet)

- Many sentiment words have multiple meanings or senses and they do not always express sentiment in all contexts.

“*great grandfather*”

“*This is clearly a bad phone.*”

“*Well, I do not think that this is a good car.*”

- Some express different sentiments in different context

“*This car smells.*”

“*This perfume smells good.*”

“*This room smells bad.*”

“*This room has a smell.*”

“*This room has a foul smell.*”

“*This room has a nice smell.*”



Lexicon semi-supervised approach



- Uses **observations** from words/ their patterns to construct **rules** for sentiment scores.
- The sentiment scores from individual ‘patterns’ are modified through rules then aggregated.
- **Most** used in industry practice actually. Especially finance because of its **flexibility**.
- Don’t under-estimate its overall effectiveness

<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x>
2011 paper sets off trend in finance.

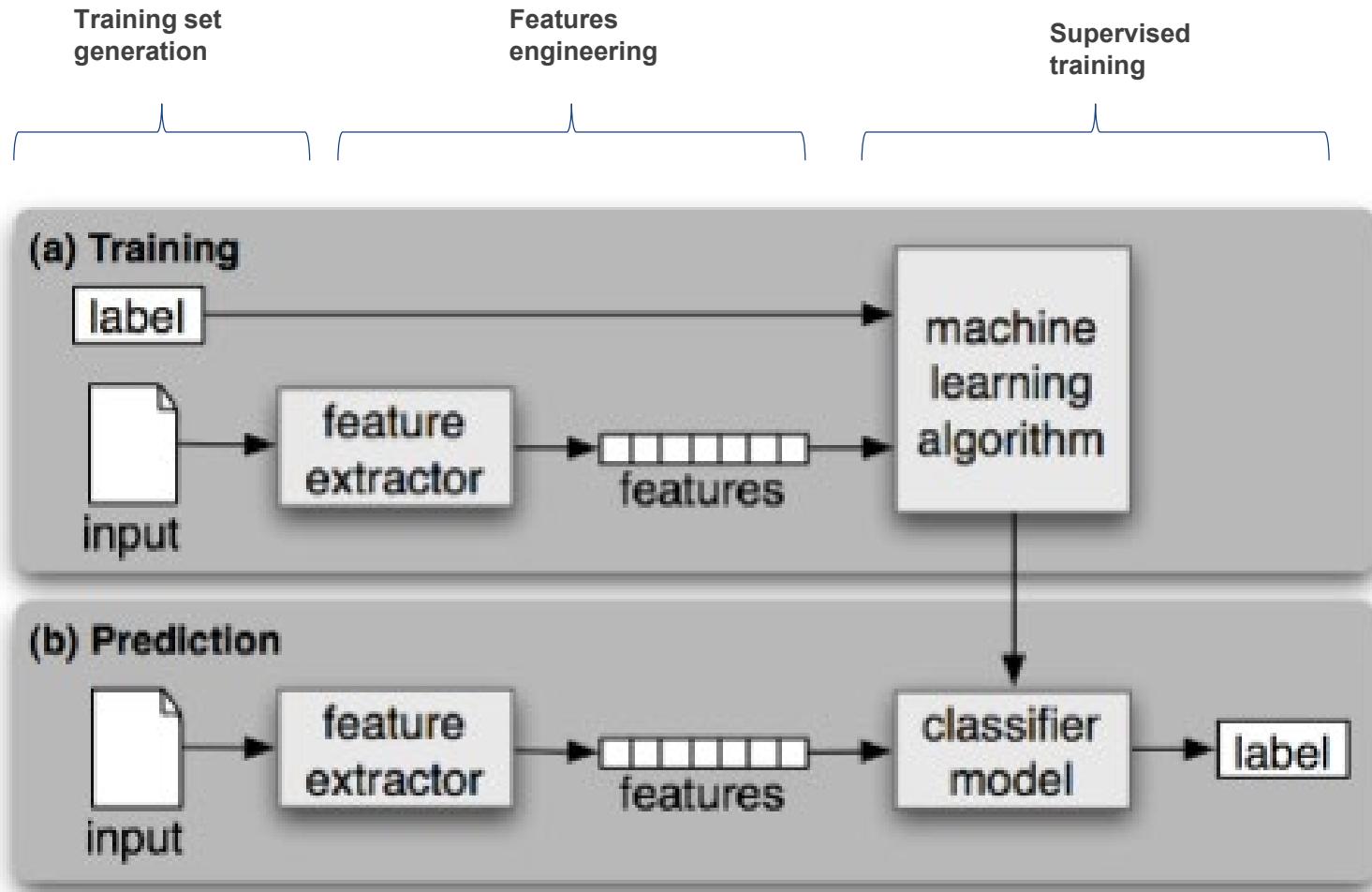
Pros and Cons of Training Methods

- **Unsupervised**
 - Pros:
 - Domain independence method, shown to perform well in a large number of applications.
 - Flexible, can be easily extended and improved.
 - Cons:
 - Heavy investment in time and effort to build the initial **knowledge base of lexicon, patterns and rules**
 - Still need to handle domain differences, mainly domain- or context-dependent sentiment expressions.
- **Supervised**
 - Pros: powerful learning of all kinds of features
 - Cons:
 - Dependent on **training data** which needs to be manually labelled for each domain.
 - Models trained in one domain often perform poorly in another domain.



2. Supervised approach

Overview of Supervised Classifier System



Sentiment Analysis Classifiers:

- **Classic Algorithms:**
 - **Naïve Bayes**
 - K-Nearest neighbor
 - Max Entropy
 - **Support Vector machine**
 - Deep learning (detailed in next class)



1. Training data for sentiment analysis

Training data set I

- The key point is to use the training data as **similar** to the test set, which applies generally for all supervised training models.
- The training and test data sets should be used from the **same domain** as far as possible. It solves problems of domain-specific terms. In most cases, best to generate a training data set **for your specific objective**.
- In generating training set, go for “**high-precision**” and “**low recall**”.
 - *High precision* means be sure those you say are positive are indeed positive. those you say are negative are indeed negative. Normally happens if you **set a ‘high bar’**.
 - *Low recall* means a lot of the actual positives or actual negatives are actually “ignored” as they cannot clear the ‘high bar’.
- **Balance** your dataset

Training data set II

- Ways to create training reference data:
 - dictionary corpus
 - user-generated means
 - manual (by inspection – tedious)

The training data set is usually ***not static*** but requires fine-tuning even after production.

This helps to account for ***changing fads*** in expressions, languages slangs etc as well.

Dictionary corpus

- Using existing dictionary corpus, egs are:
 - i. SentWordNet
 - ii. Public sources eg. Liu Bing,
<https://www.w3.org/community/sentiment/wiki/Datasets>
 - iii. nltk corpus

Then **expand** and **modify** the dictionary corpus.

First, a revision over **synsets** (revision primer from Text Mining)



Wordnet synsets

- WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
<https://wordnet.princeton.edu/>
- The training corpus can be **expanded** using bootstrapping. through WordNet **synsets**, or related words.
- SentiWordnet adds on to WordNet by assigning sentiment polarity to these senses

More in the workshop today on **WordNet** and **SentiWordNet**...

Bootstrapping synsets

- The **bootstrapping** of wordnet synsets can be understood in 2 steps.
 - Use a **seed** set of positive and negative words with their sentiment. Iterate through one by one
 - Search for the seed word's **synset** of records. These words then takes on the original seedset's sentiment.

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show

Display options for sense: (gloss) "an example sentence"

Verb

- S: (v) [decelerate](#), [slow](#), [slow down](#), [slow up](#), [retard](#) (
"The car decelerated")
- S: (v) [slow](#), [slow down](#), [slow up](#), [slack](#), [slacken](#) (bec
slowed")
- S: (v) [slow](#), [slow down](#), [slow up](#) (cause to proceed r
him down")

Adjective

- S: (adj) [slow](#) (not moving quickly; taking a comparal
"the slow lane of traffic"; "her steps were slow"; "he i
news"; "slow but steady growth")
- S: (adj) [slow](#) (at a slow tempo) "the band played a s
met anyone quite so dim"; "although dull at classical
was uncommonly quick"- Thackeray; "dumb officials
decisions"; "he was either normally stupid or being c
the slow students")
- S: (adj) [slow](#) ((used of timepieces) indicating a time
"the clock is slow")

User-generated ratings

- Use the **meta-data** in social media to assign positive or negative ratings to the comment posts.



Benny053
Brisbane,
Australia

267 62

Reviewed 25 March 2018

Delicious Dinner

Visited this restaurant with extended family. The restaurant had great ambience, good food but service was patchy. They served one of the best Sweet Sour Pork dish I have tasted. The Deep Fried Garoupa with Soy Sauce was also well done. Steamed Minced Pork over Soft Tofu was delightful. The Chinese Vegetable was delicious and so was the Fried Pork Collar Butt over Lettuce. We also had a Platter of 3 Roasted meats and they were well prepared. We also had the Minced Pork with Stir Fried Long Beans that was well cooked. The experience was memorable albeit a little pricey.

Show less

Value

Service
Food

See all 20 reviews by Benny053 for Singapore

Ask Benny053 about Canton Paradise



shuyim1
246 51

Reviewed 9 November 2017 □ via mobile

Delicious food with great service!

Our family had a birthday lunch at Canton Paradise and thoroughly enjoyed ourselves. The tim sum was delicious and we added some Chinese dishes. Everyone ate our fill and it's very reasonably price. We will be back for more!

Thank shuyim1



Simply use high ratings this as positive labels; and low ratings as negative labels

User-generated ratings

- Mind the biased reviews.

回忆
nova青春版 2020/2/7
版本号 4.7.27

这个软件真的太太太太太棒了，本来以为在家写完该死的作业后就可以愉快的玩耍了，没想到还有这种好软件，我根本没有被强迫下载钉钉，也没有被强迫加入班级团队，更没有被强迫使用钉钉。我愉快的写着钉钉班级布置的作业，根本没有感到不耐烦，原来钉钉这个软件的出现是为了帮我杀假期中无聊（宝贵）的时间，因为这个软件的出现，我打游戏的时间一下子减少了3/1，真是太棒了，我可以拜托无聊（有趣）的游戏，来自愿（被强迫）使用有趣（无聊）钉钉，这真是太棒了!!!这种软件一定要一星好评的啦！

17,182 556 45

アプリは悪く無いんですけど... 2月18日
★★☆☆☆ こんな世界と嘆く誰かの生き...

アプリは使いやすいんですけどね...もう嫌なんですよ無理なんです(´;`;) 最近の通知音は孫悟空の緊箍咒に思えてきました(今日も頭痛が絶えないんです...) 宿題の通知はまるで取り立て...お代官様 あっしに納められる年貢はもうさらに表示



2. Features for sentiment analysis

Features used in sentiment mining

- From Wikipedia:
 - **Feature engineering** is the process of using domain knowledge of the data to create features that make machine learning algorithms work.



- Represent [sentences] with a vector of numbers, which can better/best distinguish the [polarity] among all the [sentences]

Feature Engineering

- Some common features used in sentiment analysis are
 - Part of speech (POS) tags (adjectives or nouns)
 - Opinion lexicons and phrases (n-grams)
 - Negations
 - Syntactic dependency (more about this on Day 3)
 - Sentiment-aware tokens (recall 1st day)
 - Word vectors
 - Terms frequency and different information retrieval *weighting* schemes – tf-idf

What are other word features do you think will matter?

In an actual project, it is wise to **look through some data** sets in some detail, and identify what sets positive or negative polarity statements apart.

Feature Selection (or extraction)

- To select **relevant features** and reduce the number of features used in the matrix
- Various ways (**trial-and-error**):
 - Remove **features that appear rarely** in the documents
 - Select **top K** number of most **frequent** features
 - Leverage on the labels to pick **K most useful** features

More about it in workshop.

- Two key steps before building a sentiment analysis are:
 - i. *Training data (corpus)* selection/ generation
 - ii. *Features selection*

These pre-steps are key to the success of a sentiment analysis and usually **more important than the training algorithms** themselves.

Training data selection needs to be as similar as possible to the production data. The features selection requires domain expertise.



Features from WordToVec

- Feature generation and selection could be tedious
- How might we generate “*universal*” features **automatically?**
- Zooming into **word level** vector representation

Features from **Word Vectors**

- **Count from Data**
 - Word Co-occurrence + SVD
 - Count-based model
- **Learn from Data**
 - CBOW and SKIPGRAM
 - NN Methods
 - Predictive Model
- **Count and Learn from Data**
 - GLOVE: Global Vectors for Word Representation
 - Count + SGD

Count from Data

- Word-level representation
- Counting context-words within a `window_size`

Sent_1: *I like deep learning*

Sent_2: *I like NLP*

Sent_3: *I enjoy flying*

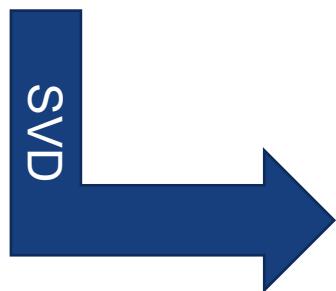
Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

Count From Data

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

(7,7)



	S1	S2
I	1.5	.1
like	3.14	.23
enjoy	2.7	-.98
Deep	.55	.1
learning	.8	2.5
NLP	-2.5	3
flying	4.5	4.9

(7,N)

Sorted Singular Values
12.29
6.2
...

(N,N)

	I	like
S1	.1	2	3	4	6	7
S2	.5	6	7	3	1	8

(N,7)

Count From Data

$\text{vec}(I) =$
$\text{vec}(\text{like}) =$
$\text{vec}(\text{enjoy}) =$
$\text{vec}(\text{deep}) =$
$\text{vec}(\text{learning}) =$
$\text{vec}(\text{NLP}) =$
$\text{vec}(\text{flying}) =$

	S1	S2
I	1.5	.1
like	3.14	.23
enjoy	2.7	-.98
Deep	.55	.1
learning	.8	2.5
NLP	-2.5	3
flying	4.5	4.9

Sorted Singular Values
12.29
6.2
...

(N,N)

(7,N)

Features from Word Vectors

- **Count from Data**
 - Word Co-occurrence + SVD
 - Count-based model
- **Learn from Data**
 - Word2Vec
 - NN Methods
 - Predictive Model
- **Count and Learn from Data**
 - GLOVE: Global Vectors for Word Representation
 - Count + SGD

Learn From Data

One-Hot Encoding (Sparse Representation)

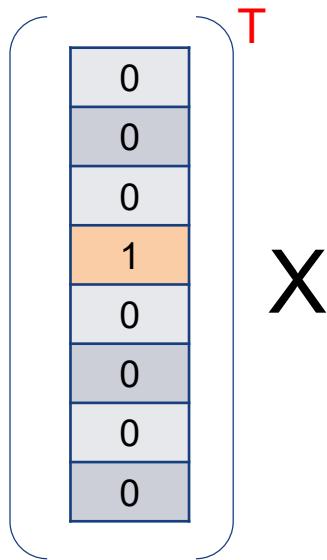
Vocabulary of the corpus (big enough)



	he	she	eats	drinks	sushi	ramen	hungry	coffee
he	1	0	0	0	0	0	0	0
drinks	0	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0
coffee	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	0
	0	0	1	0	0	0	0	0

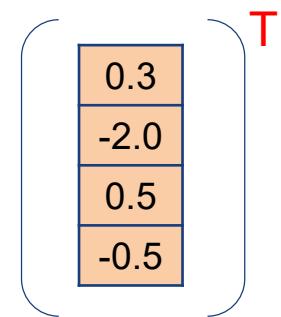
$v('drinks')$

Word2Vec



0.1	0.2	-0.4	0.9
0.2	0.1	-0.3	0.9
0.2	-1.4	0.3	-0.1
0.3	-2.0	0.5	-0.5
0.2	-1.1	0.3	-0.7
0.9	-1.3	0.4	-0.9
0.3	-3.0	0.5	-0.2
0.5	-0.1	0.2	0.1

Word Embeddings



One-Hot Encoding

$1 \times |V|$

$v('drinks')$

$|V| \times d$



Input

$1 \times d$

$v('drinks')$

Features from WordToVec

- **Count from Data**
 - Word Co-occurrence + SVD
 - Count-based model
- **Learn from Data**
 - Word2Vec
 - NN Methods
 - Predictive Model
- **Count and Learn from Data**
 - **GLOVE**: Global Vectors for Word Representation
 - Count + SGD

GLOVE-Global Vectors for Word Representation

- Word-level representation
- Counting context-words within a **window_size**

Sent_1: *I like deep learning*

Sent_2: *I like NLP*

Sent_3: *I enjoy flying*

Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

(# of “*like*” as “*I*’s” context-words) = 2

(# of “*I*” as “*like*’s” context-words) = 2

GLOVE-Global Vectors for Word Representation

- Word-level representation
- Counting context-words within a window_size

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

$$P(I, \text{like}) = P(\text{like} \mid I) = \frac{\text{Count}(\# \text{ of "like" as "I's" context-words})}{\text{Count}(\text{ total # of "I's" context-words})}$$

$$= C_{I,\text{like}} / C_I = 2/3$$

$$\log(P(I, \text{like})) = \log(C_{I,\text{like}} / C_I) = \log(C_{I,\text{like}}) - \log(C_I) = \log 2 - \log 3$$

GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

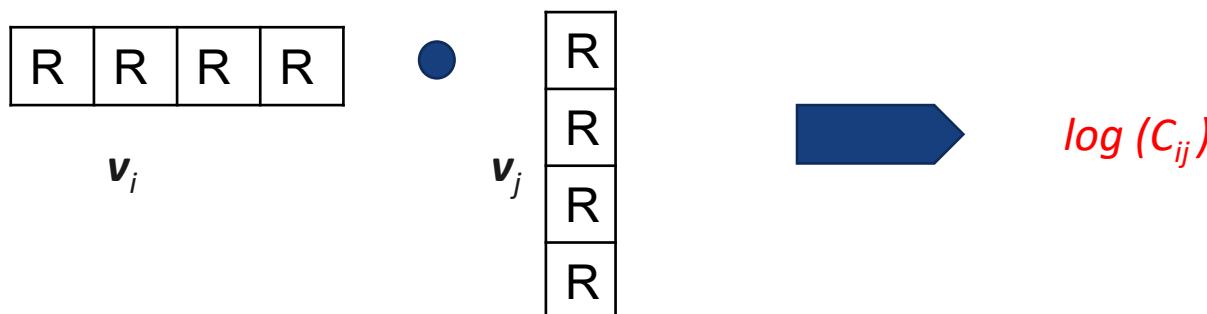
counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

$$\log(P(I, \text{like})) = \log(C_{I, \text{like}} / C_I) = \log(C_{I, \text{like}}) - \log(C_I) = \log 2 - \log 3$$

Let v_i = the vector representing "I" i refers to "I"

v_j = the vector representing "like" j refers to "like"

Then we Expect : mapping $v_i \bullet v_j$ to $\log(C_{ij})$



GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

Let v_i = the vector representing "I" i refers to "I"

v_j = the vector representing "like" j refers to "like"

Then we Expect : mapping $v_i \bullet v_j$ to log (C_{ij})

Thus we Define: Least Square Loss Function : $L = \sum_{ij} [\log(C_{ij}) - (v_i \bullet v_j + v_{bias})]^2$

$$L2LossFunction = \sum_{i=1}^n (y_{true} - y_{predicted})^2$$

GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

Moreover we Define: weighted least square Loss Function :

$$L = \sum_{i,j} [\log(C_{ij}) - (\mathbf{v}_i \bullet \mathbf{v}_j + v_{bias})]^2 \bullet \text{Weight_Func}(C_{ij})$$

Constrains:

$$\text{Weight_Func}(0) = 0$$

Bigger C_{ij} leads to Bigger Weight_Func(C_{ij})

Weight_Func(C_{ij}) should have a upper bound as C_{ij} can be a big number

GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

$$L = \sum_{i,j} [\log(C_{ij}) - (\mathbf{v}_i \bullet \mathbf{v}_j + v_{bias})]^2 \bullet \text{Weight_Func}(C_{ij})$$

$$\text{Weight_Func}(C_{ij}) = \begin{cases} 1, & \text{when } C_{ij} \geq 100 \\ (C_{ij}/100)^{0.75}, & \text{otherwise} \end{cases}$$

Constrains:

$$\text{Weight_Func}(0) = 0$$

Bigger C_{ij} leads to Bigger Weight_Func(C_{ij})

Weight_Func(C_{ij}) should have an upper bound as C_{ij} can be a big number

GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

Window_size=1

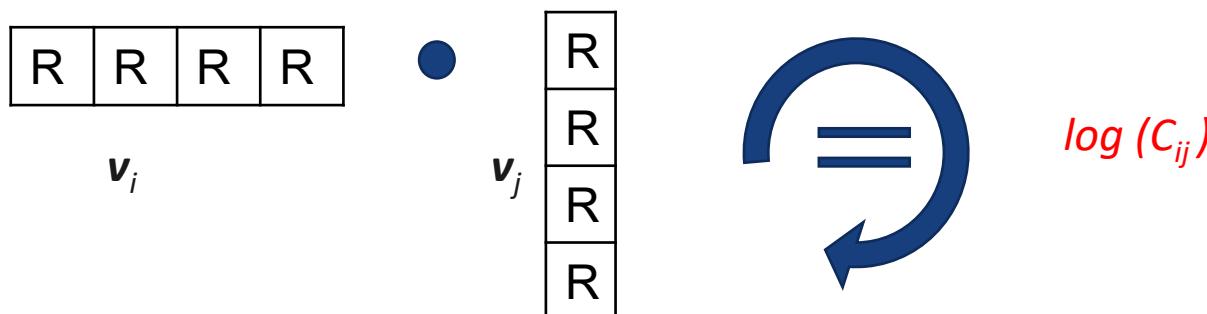
counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

$$\log(P(I, \text{like})) = \log(C_{I, \text{like}} / C_I) = \log(C_{I, \text{like}}) - \log(C_I) = \log 2 - \log 3$$

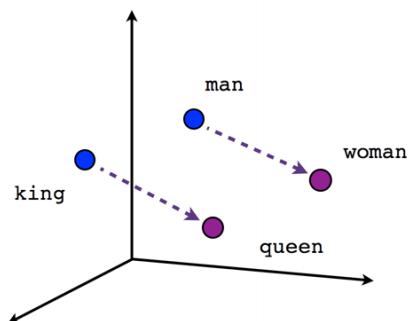
Let v_i = the vector representing "I" i refers to "I"

v_j = the vector representing "like" j refers to "like"

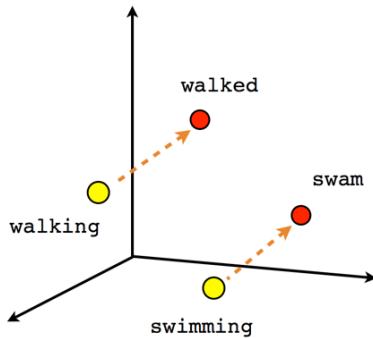
Then we Expect : mapping $v_i \bullet v_j$ to $\log(C_{ij})$



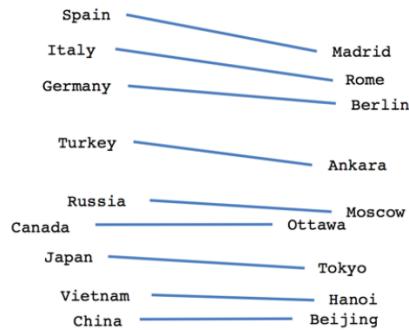
Properties of Word Vectors



Male-Female



Verb tense



Country-Capital

Ingredients

Corpus of text	As large as possible
Annotations	0
Initialize weights (aka Embeddings)	1x per word
Deep Learning Model	1x
Cost Function	Appropriately
GPU	Lotsa of it

Features from WordToVec

- **Count from Data (SVD)**
 - Best for Word-level Similarity
 - significantly outperformed word2vec and GLOVE
 - Computational cost is High
 - Difficult to handle huge matrix
 - Keep window short (window_size = 2)
- **Learn from Data (Word2Vec)**
 - Cheap to train and Scales with corpus size
 - Generate improvements on other predictive tasks
 - Capture complex patterns beyond word level similarity
- **Count and Learn from Data (GLOVE)**
 - Fast to Train
 - Believed to have advantages from both
 - Differences are not obvious
 - Widely used (pre-trained model from Wikipedia and Twitter)

Sentence Embedding

$v('he')$	0.5	-1.3	0.6	1.1	
$v('drinks')$	0.3	-0.2	0.5	0.5	
$v('coffee')$	1.3	2.1	-0.8	1.1	
	$\text{AVG()}/\text{MAX()}/\text{MIN()}/\text{Concat}()$				
Sent0 ("he drinks coffee")	0.7	0.2	0.1	0.9	

Instead of picking K most useful features,
here take N dimensional Word Embedding

Combine Feature Sets

v('he')	0.5	-1.3	0.6	1.1
v('drinks')	0.3	-0.2	0.5	0.5
v('coffee')	1.3	2.1	-0.8	1.1
AVG()				
Sent0 ("he drinks coffee")	0.7	0.2	0.1	0.9

	he	drink	coffee	<i>tfidf</i> drink coffee	<i>tfidf</i> he drink	<i>PPMI</i> he drink	<i>PPMI</i> drink coffee	...	Sent Vector				
sent0	0.01	0.38	0.00	0.87	0.00	4.23	0.00	...	0.7	0.2	0.1	0.9	

```
final_train = np.c_[X_w2v_train,X_glove_train,k_best]  
final_train.shape
```

How to Choose Context?

- **Different contexts lead to different embeddings**
- **Small context window:** more syntax related
- **Large context window:** more semantics related

Limitations

- **Sensitive to “tokens”** (cat vs cats)
- **Inconsistent across space**, embeddings for the same words trained with different data are different
- **Can encode bias** (stereotypical gender roles, racial bias)
- **Not interpretable**

- Key steps before building a sentiment analysis are:
 - i. *Training data (corpus)* selection/ generation
 - ii. *Features selection*
 - iii. *Features from embedding*

These pre-steps are key to the success of a sentiment analysis and usually **more important than the training algorithms** themselves.

Training data selection needs to be as similar as possible to the production data. The features selection requires domain expertise.

Word Embeddings can be retrained with domain data or downloaded from pre-trained data

NMSM Day2

Neural Nets & W2V

Dr Wang Aobo
aobo.wang@nus.edu.sg

Agenda

- Statistical Modelling vs. Deep Neural Nets
- Deep Learning Basics for NLP
 - MLP
 - Deep Learning Training Routine (quiz)
 - Workshop: Basic NN on Colab
- Word2Vec & DL Specific
 - CBOW and SkipGram
 - ActivationFunction/LossFunction
 - Optimiser/Learning Rate
 - Workshop: Word2Vec from Scratch

How does NLP Work

- The whole task here is...

Documents

Lost glamor
R
o
High tea at Raffles!
R
o
2 Not what it was, but still
a
R
o
Amazing service
R
o
Rated 5 by travel-gini
on Feb 26, 2013
th
Great location with a
little bit of history, the
staff make this hotel
though



Doc/Feature Matrix

	amazing	service	lost	glamour	...
Doc1	1.5	2.1	0	0	
Doc2	0	0	3.1	1.5	
Doc3	0	0	0	1.9	
Doc4	0	0	0	0	
...					



ML + CPU
Supervised
Unsupervised

Classic NLP vs. Deep learning

- Frequency TF-IDF and PPMI vectors are
 - long ($|V| > 100,000$)
 - sparse (lots of zero)
 - efficient for simple tasks with reasonably large of dataset
 - interpretable as designed by human intelligence
 - **difficult to capture contextual dependency**

	he	drink	hold	...	<i>tfidf</i> drink apple	<i>tfidf</i> hold apple	<i>tfidf</i> apple juice	...	<i>PPMI</i> drink apple	<i>PPMI</i> hold apple	<i>PPMI</i> apple juice
sent0	0.01	0.38	0.00	...	0.87	0.00	0.92	...	4.23	0.00	8.90
sent1	0.01	0.00	0.28	...	0.00	0.87	0.00	...	0.00	2.45	0.00

Statistical modelling

- **All models are wrong, but some are useful**
 - for Document/Sentence Classification
 - SVM/KNN/Decision Tree/RandomForest/Naïve Bayes/MaxEnt
 - for Sequence Labeling
 - HMM/CRF
 - for Language modeling
 - N-gram/RandomForest/MaxEnt
 - for Machine Translation
 - Phrase/Tree-based Model + beam search

How does NLP Work

- The whole task here is...

Documents

Lost glamor
R
o
High tea at Raffles!
R
o
2 Not what it was, but still
a
R
o
Amazing service
R
o
Rated 5 by travel-gini
on Feb 26, 2013
th
Great location with a
little bit of history, the
staff make this hotel
though



Doc/Feature Matrix

	amazing	service	lost	glamour	...
Doc1	1.5	2.1	0	0	
Doc2	0	0	3.1	1.5	
Doc3	0	0	0	1.9	
Doc4	0	0	0	0	
...					



ML + CPU
Supervised
Unsupervised

- Able to generate this Matrix **automatically** ?
- Able to generate this Matrix **task-independently** ?
- Yes! Learn to generate from data

How does NLP Work

- The whole task here is...

Documents

Lost glamor
R High tea at Raffles!
o Raffles DV
2 Not what it was, but still a...
W Amazing service
R o Rated 5 by travel-gini
on Feb 26, 2013
th Great location with a
little bit of history, the
staff make this hotel
though



Doc/Feature Matrix

	amazing	service	lost	glamour	...
Doc1	1.5	2.1	0	0	
Doc2	0	0	3.1	1.5	
Doc3	0	0	0	1.9	
Doc4	0	0	0	0	
...					



ML + CPU
Supervised
Unsupervised



Word Indexing Matrix

	amazing	service	lost	glamour	...
amazing	1	0	0	0	
service	0	1	0	0	
lost	0	0	1	0	
glamour	0	0	0	1	
...					



DNN + GPUs
Supervised



	?	?	?	?	?
Doc1	11.5	2.1	5.70	-30.2	
Doc2	-3.40	0.34	3.1	1.5	
Doc3	5.8	0.560	5.9	1.9	
...					

Classic NLP vs. Deep learning

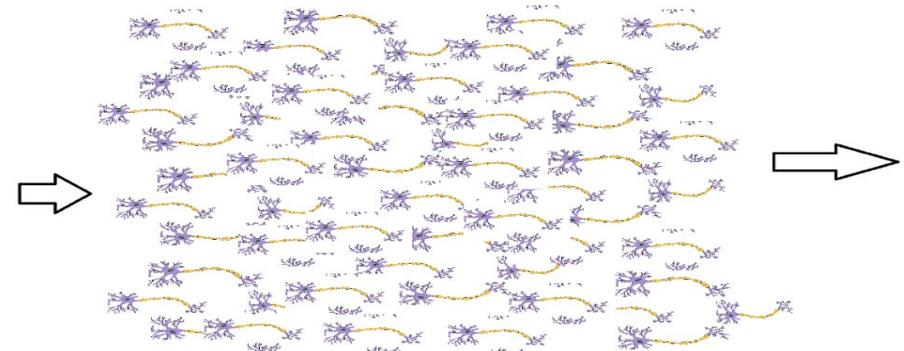
- Deep learning can **create** feature vectors that are
 - short (often fixed-sized <2000, decided empirically)
 - dense (most are non-zeros)
 - **non-Interpretable** as decided empirically without human intelligence
 - able to capture **contextual dependency**
 - beneficial to **all** tasks (classification/sequence labeling/Translation/QA)

Agenda

- Deep Learning Basics for NLP
 - MLP
 - Deep Learning Training Routine (quiz)
- Workshop: Deep Learning from Scratch
- Word2Vec & DL Specific
 - CBOW and SkipGram
 - ActivationFunction/LossFunction
 - Optimiser/Learning Rate
- Workshop: Word2Vec from Scratch

Deep Learning Basics

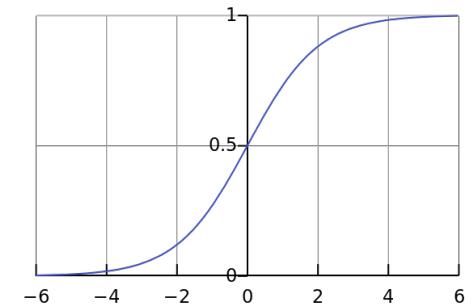
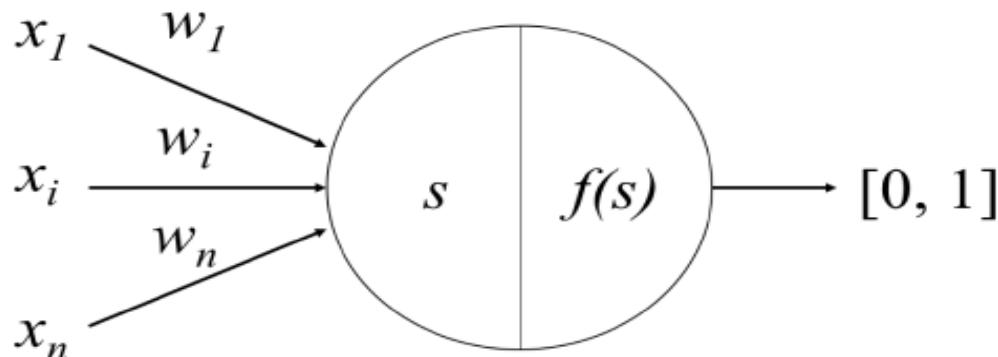
- Perceptron
 - "*system that depends on **probabilistic** rather than deterministic principles* for its operation, gains its reliability from the properties of statistical measurements obtain from a large population of elements"
 - Frank Rosenblatt (1957)
 - 100 billion **perceptron** in our brain
 - BERT ~ 110 million to 17 billion **para**
 - GPT ~ 1.5 billion to 175 billion **para**
 - **LLAMA 7B~**
 - **GPT4 ???**



(*Image from [Akshay Chandra Lagandula's blog](#))

Recap Perceptron

- Given a **set of inputs x** , perceptron
 - learns **w vector** to map the inputs to a real-value output between [0,1]
 - through the summation of the dot product of the $w \cdot x$
 - with a transformation function (aka. **activation function**)



Summation

$$s = \sum w \cdot x$$

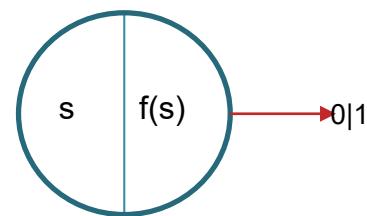
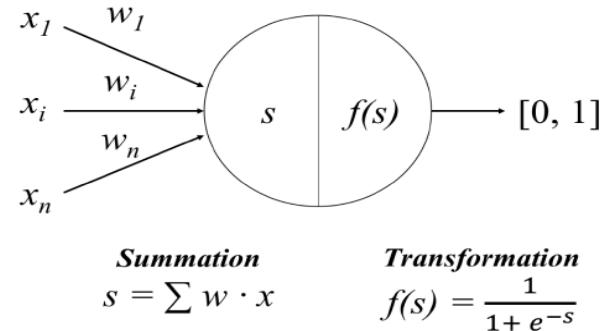
Transformation

$$f(s) = \frac{1}{1+e^{-s}}$$

Perceptron

- Word level classification
 - Positive=1; Negative=0
 - let $n=5$

	Like x1	Hate x2	Good x3	Enjoy x4	Bad x5
Like	1	0	0	0	0
Hate	0	1	0	0	0
Good	0	0	1	0	0
Enjoy	0	0	0	1	0
Bad	0	0	0	0	1



Perceptron

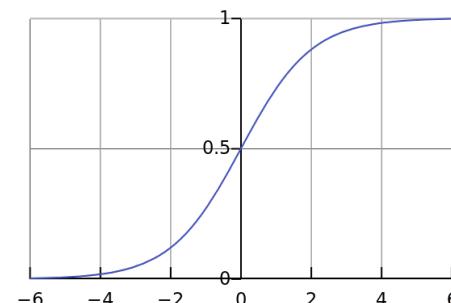
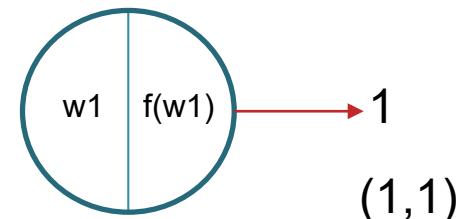
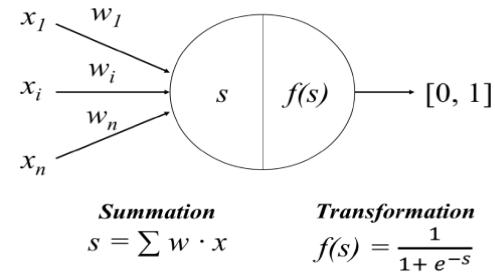
- Word level classification
 - Positive=1; Negative=0
 - let $n=5 = \text{vocabulary_size}$

	Like x1	Hate x2	Good x3	Enjoy x4	Bad x5
Like	1	0	0	0	0
Hate	0	1	0	0	0
Good	0	0	1	0	0
Enjoy	0	0	0	1	0
Bad	0	0	0	0	1

(1,5)

weight
w1
w2
w3
w4
w5

(5,1)

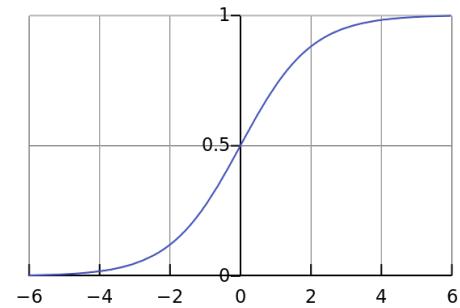
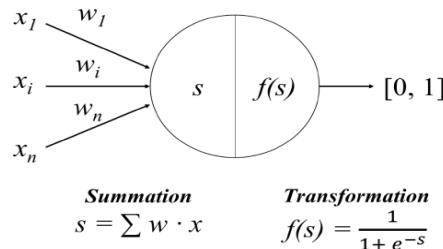


Perceptron

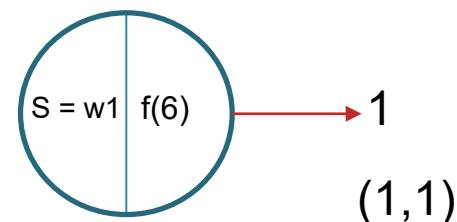
- Word level classification
 - Positive=1; Negative=0
 - let $n=5 = \text{vocabulary_size}$
 - $\text{Batch_size} = 1$

	Like x1	Hate x2	Good x3	Enjoy x4	Bad x5
Like	1	0	0	0	0
Hate	0	1	0	0	0
Good	0	0	1	0	0
Enjoy	0	0	0	1	0
Bad	0	0	0	0	1

(1,5)



weight
w1=6
-6
6
6
-6
(5,1)



- How about unknown words?

Perceptron

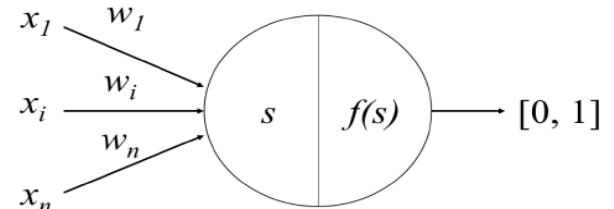
- Word level classification
 - Positive=1; Negative=0
 - let $n=26 \ll \text{vocabulary_size}$

	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	1	1	0	...	0

(1,26)

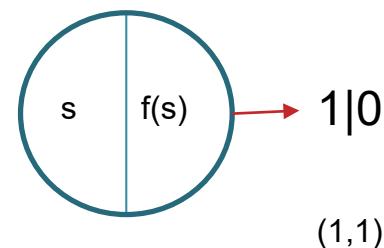
weight
w1
w2
w3
...
w26

(26,1)



$$\text{Summation} \\ s = \sum w \cdot x$$

$$\text{Transformation} \\ f(s) = \frac{1}{1 + e^{-s}}$$



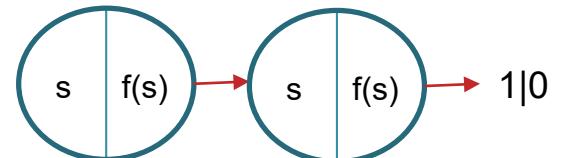
(1,1)

Multi-Layer Perceptron

- Word level classification
 - let $n=26 << \text{vocabulary_size}$
 - More parameters
 - More layers

	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	0	1	0	...	0

weight
w1
w2
w3
...
w26

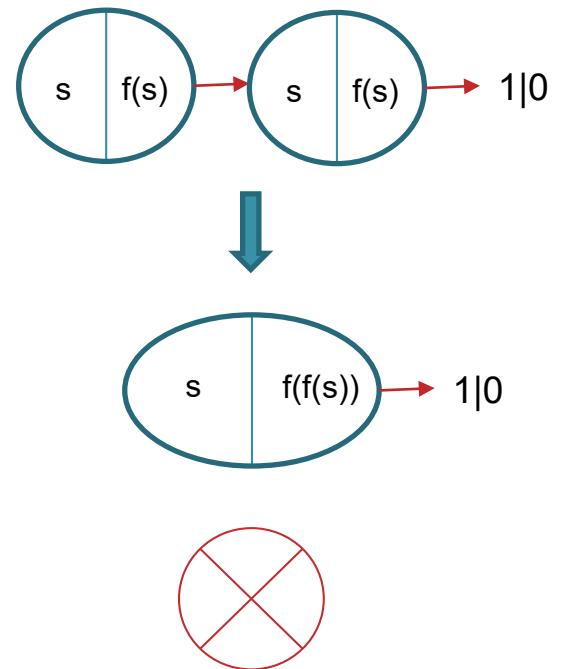


Multi-Layer Perceptron

- Word level classification
 - let $n=26 << \text{vocabulary_size}$
 - More parameters
 - More layers

	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	0	1	0	...	0

weight
w1
w2
w3
...
w26



Multi-Layer Perceptron

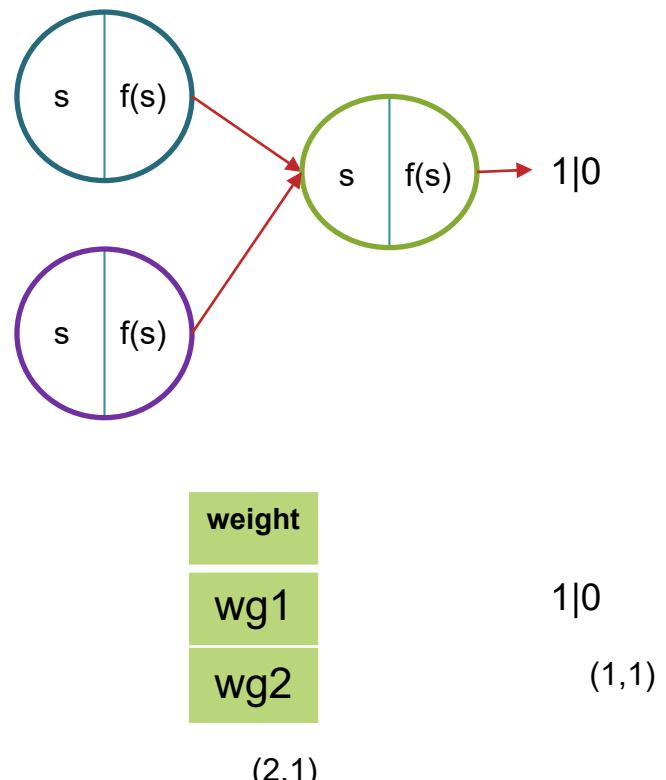
- Word level classification
 - let $n=26 << vocabulary_size$
 - More layers
 - More parameters

	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	0	1	0	...	0

(1,26)

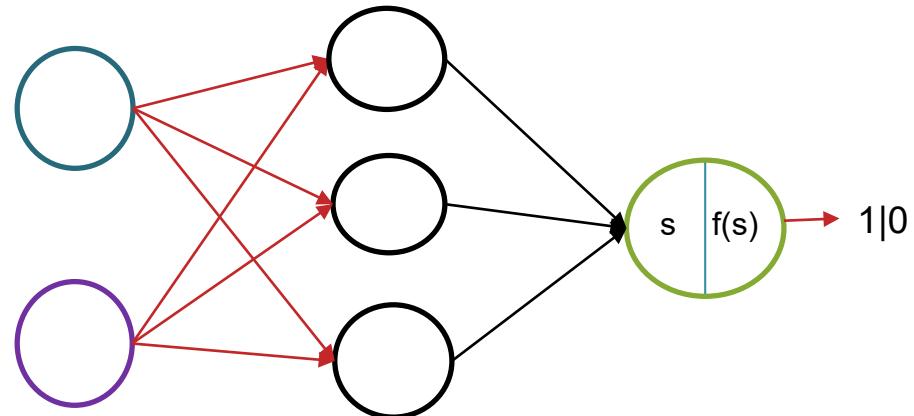
weight	weight
wb1	wp1
wb2	wp2
wb3	wp3
...	...
wb26	wp26

(26,2)



Forward Propagation

- Word level classification
 - let $n=26 << vocabulary_size$
 - More parameters
 - Even More Layers*



	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	0	1	0	...	0

weight	weight
wb1	wp1
wb2	wp2
wb3	wp3
...	...
wb26	wp26

weight	weight	weight
w	w	w
w	w	w

weight
wg1
wg2
wg3

$(1,1)$
 $(2,3)$
 $(3,1)$

Y_{pred}

(1,26)

(26,2)

Deep learning Training Routine

Repeat the following until desired

- Initialize weights vector
 - Random
 - One-hot encoding
- Forward Propagation
- Compute and log the loss
- Back Propagation
- Optimizer

Deep learning Training Routine

“brute force” searching action with **strategy** until desired

- Compute and keep the Cost/Loss
 - Define and compute the $loss(Y_{pred}, Y_{correct})$
- Back Propagation
 - Compute the **partial derivatives rate** of *Loss function wrt W for all layers*
 - Chain rule applies
- Optimizer (“brute force” searching action + **Strategy**)
 - Gradient Descent and Delta rule
$$\text{New weight} = \text{Old weight} - \text{Derivative Rate} * \text{Learning rate}$$
 - learning rate a constant (usually very small)
 - to avoid big steps

Optimization

- Example
 - $y = wx$ with $n=5$ training examples (forward)
 - Loss function: $L = \frac{\sum(y_p - y_t)}{n} = \frac{\sum(wx - y_t)}{n}$
 - when w randomly initialized as 3 (guessing)

x	y_{pred} ($w=3$) $y=3x$	y_{true}	$y_{pred} - y_{true}$
0	0	0	0
1	3	2	1
2	6	4	2
3	9	6	3
4	12	8	4
Loss	-	-	2

1st iter

$$\frac{dL}{dw} = \sum_{i=0}^n x/n = 2$$

$$w^{new} = w^{old} - \eta \frac{dL}{dw}$$

$$\eta = 0.5 \quad w_{init} = 3$$

$$w^{new} = 3 - 0.5 * 2 = 2$$

$$y = 2x \quad \text{Loss function}=0$$

Gradient Descent

- Minimize a “fake” lost function

$$L = w^2 - 4w + 6$$

$$\frac{dL}{dw} = 2w - 4$$

- Iterations

$$w^{new} = 3.00 - 0.3(2 * 3.00 - 4) = 2.40$$

$$w^{new} = 2.40 - 0.3(2 * 2.40 - 4) = 2.16$$

- Apply Delta rule

$$w^{new} = w^{old} - \eta \frac{dL}{dw}$$

$$\eta = 0.3 \quad w_{init} = 3$$

Chain Rule

$$\text{loss}(Y_{pred}, Y_{correct}) = Y_{pred} - Y_{correct} = \underbrace{x * w_{bp}}_{f_1(x, W_{bp}) = y_1} * \underbrace{w}_{f_2(y_1, W) = y_2} * \underbrace{w_g}_{f_3(y_2, w_g) = y_3} - 1$$

$$\frac{\partial \text{Loss}}{\partial w_{bp}} = \frac{\partial(y_3)}{\partial y_2} \frac{\partial(y_2)}{\partial y_1} \frac{\partial(y_1)}{\partial w_{bp}}$$

	a x1	b x2	c x3	...	z x26
like	0	0	0	...	0
hate	1	0	0	...	0
good	0	0	0	...	0
enjoy	0	0	0	...	0
bad	0	1	0	...	0

weight	weight
wb1	wp1
wb2	wp2
wb3	wp3
...	...
wb26	wp26

weight	weight	weight
w	w	w
w	w	w

weight
wg1
wg2
wg3

y
(1,1)

(2,3) (3,1)

$$W_{bp}^{new} = W_{bp}^{old} - \eta \frac{\partial \text{Loss}}{\partial W_{bp}}$$

(1,26)

(26,2)

Deep learning Training Routine

Repeat the following until desired

- Initialize weights vector W for all layers
- **Forward Propagation**
 - *reaching the final layer to get Y_{pred}*
- **Compute and keep the cost/loss**
 - Define and compute the $loss(Y_{pred}, Y_{correct})$
- Back Propagation
 - **partial derivatives rate** of *Loss function wrt W for all layers*
- Optimizer
 - *New weight = Old weight - Derivative Rate * Learning rate*



Agenda

- **Workshop: NN Basics on Colab**
- Word2Vec & DL Specifics
 - CBOW and SkipGram
 - ActivationFunction/LossFunction
 - Optimiser/Learning Rate
- Workshop: Word2Vec from Scratch

Word2Vec & DL Specifics

Dr Wang Aobo
aobo.wang@nus.edu.sg

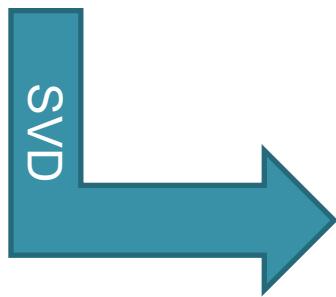
Features from Word Vectors

- Count from Data
 - Word Co-occurrence + SVD
 - Count-based model
- Learn from Data
 - CBOW and SKIPGRAM
 - NN Methods
 - Predictive Model
- Count and Learn from Data
 - GLOVE: Global Vectors for Word Representation
 - Count + SGD

Count From Data

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

(7,7)



	S1	S2
I	1.5	.1
like	3.14	.23
enjoy	2.7	-.98
Deep	.55	.1
learning	.8	2.5
NLP	-2.5	3
flying	4.5	4.9

Sorted Singular Values	
12.29	
6.2	
...	

(N,N)

	I	like
S1	.1	2	3	4	6	7
S2	.5	6	7	3	1	8

(N,7)

(7,N)

GLOVE-Global Vectors for Word Representation

Sent_1: I like deep learning

Sent_2: I like NLP

Sent_3: I enjoy flying

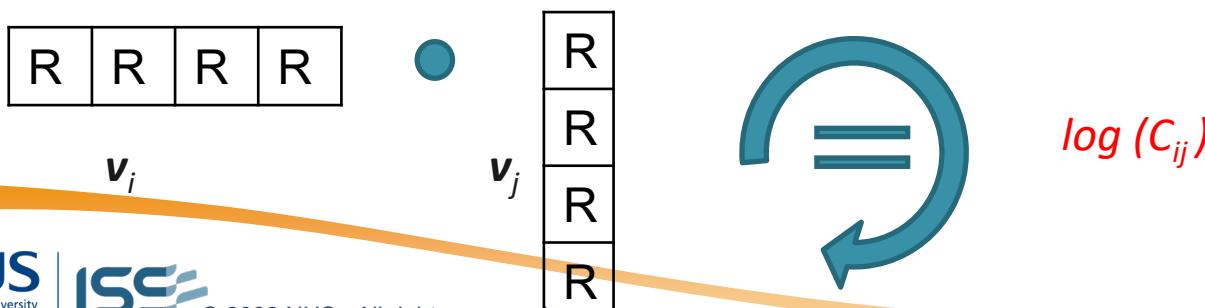
Window_size=1

counts	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

$$\log(P(I, \text{like})) = \log(C_{I, \text{like}} / C_I) = \log(C_{I, \text{like}}) - \log(C_I) = \log 2 - \log 3$$

Let v_i = the vector representing "I" i refers to "I"
 v_j = the vector representing "like" j refers to "like"

Then we Expect : mapping $v_i \bullet v_j$ to $\log(C_{ij})$



One-Hot Encoding (Sparse Representation)

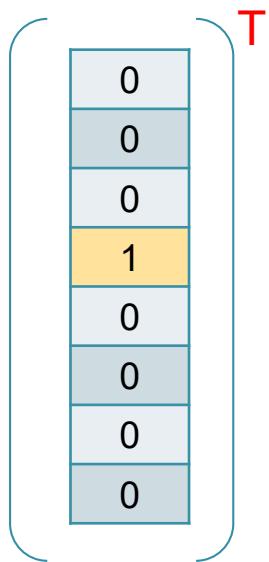
Vocabulary of the corpus (big enough)

	give	she	at	talk	have	ramen	a	drink
give	1	0	0	0	0	0	0	0
talk	0	0	0	1	0	0	0	0
have	0	0	0	0	1	0	0	0
drink	0	0	0	0	0	0	0	1
a	0	0	0	0	0	0	1	0
at	0	0	1	0	0	0	0	0

$v('talk')$

Lookup Function

$v('talk')$



One-Hot Encoding

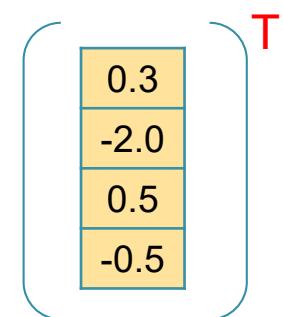
$1 \times |V|$

0.1	0.2	-0.4	0.9
0.2	0.1	-0.3	0.9
0.2	-1.4	0.3	-0.1
0.3	-2.0	0.5	-0.5
0.2	-1.1	0.3	-0.7
0.9	-1.3	0.4	-0.9
0.3	-3.0	0.5	-0.2
0.5	-0.1	0.2	0.1

Word Embeddings

$|V| \times d$

$v('talk')$
Embedded



Input

$1 \times d$

Learn the Matrix through Making Prediction

34

Word2Vec (CBOW)

Learn the Matrix through “classification” task

Sentence: the bulk of linguistic questions concern the distinction between a and m. a linguistic account of phenomenon ...

of	the bulk _____ linguistic questions
linguistic	bulk of _____ questions concern
questions	of linguistic _____ concern the
concern	linguistic questions _____ the dis-
the	questions concern _____ dis- tinction
dis-	concern the _____ tinction between
tinction	the dis- _____ between a
between	dis- tinction _____ a and
a	tinction between _____ and m.
and	between a _____ m. a
m.	a and _____ a linguistic
a	and m. _____ linguistic account
linguistic	m. a _____ account of
account	a linguistic _____ of a
of	linguistic account _____ a phenomenon
a	account of _____ phenomenon gen-
phenomenon	of a _____ gen- erally

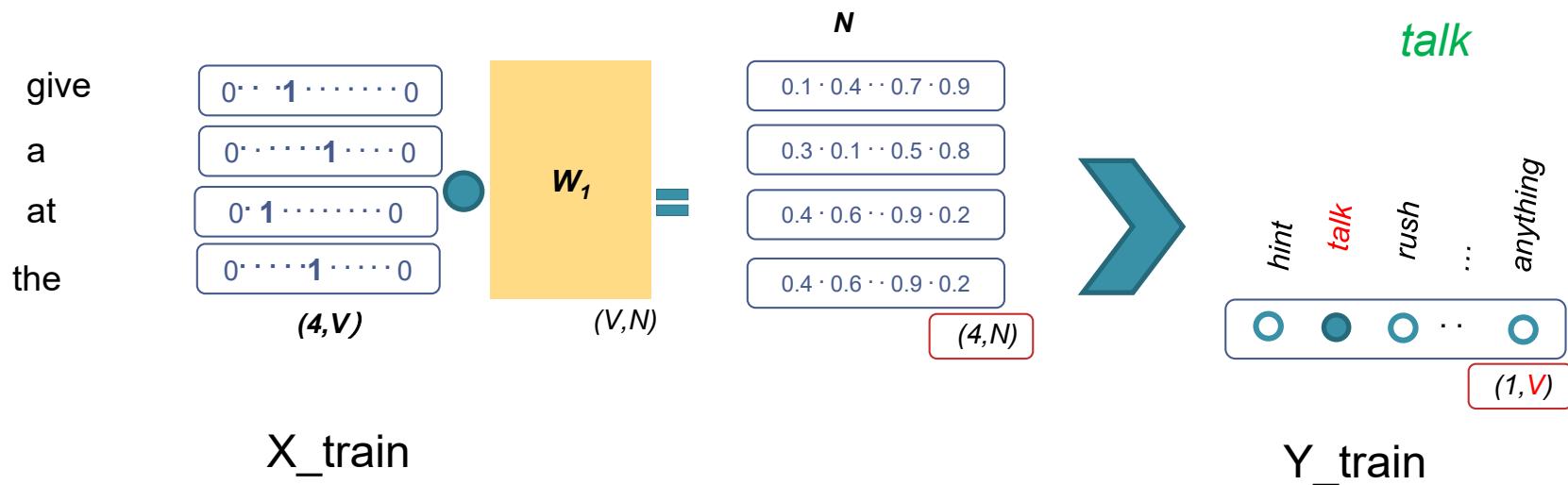
window_size = 2



Word2Vec (CBOW)

Task: Iterate through every word with a given window; learn W such the models can predict what's the word given only the context words as inputs.

give a _____ at the



Word2Vec (CBOW)

give a talk at the

give $\begin{matrix} 0 \cdots 1 \cdots \cdots \cdots 0 \end{matrix}$
a $\begin{matrix} 0 \cdots \cdots 1 \cdots \cdots 0 \end{matrix}$
at $\begin{matrix} 0 \cdot 1 \cdots \cdots \cdots 0 \end{matrix}$
the $\begin{matrix} 0 \cdots \cdots 1 \cdots \cdots 0 \end{matrix}$
 $(2 * \text{window_size}, V)$

w_1 (V, N)

N
 $0.1 \cdot 0.4 \cdots 0.7 \cdot 0.9$
 $0.3 \cdot 0.1 \cdots 0.5 \cdot 0.8$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$
 $(4, N)$

AVG () + Relu ()

Re-shaping
the values

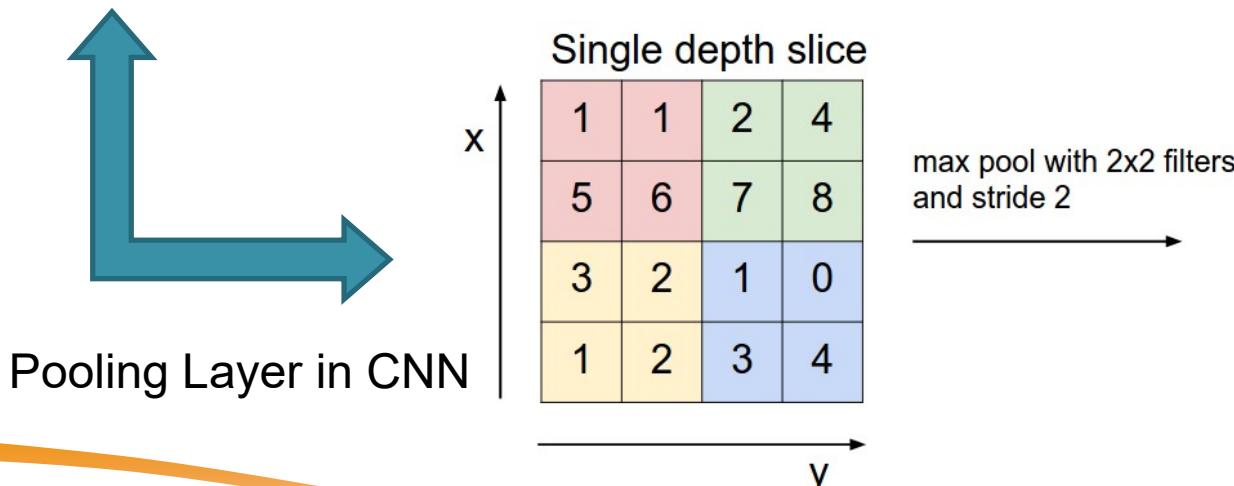
$0.1 \cdot 0.1 \cdots 7 \cdot 0.4$
 $(1, N)$

Re-shaping
the rows

hint talk rush ... anything
 $(1, V)$

Naïve Sentence Embedding

$v('give')$	0.5	-1.3	0.6	1.1
$v('a')$	0.3	-0.2	0.7	0.5
$v('at')$	0.3	2.3	-0.8	1.0
$v('the')$	1.7	-0.2	-0.1	1.0
$\text{AVG()}/\text{MAX()}/\text{MIN()}/\text{Concat}()$				
Sent0 ("give a at the")	0.7	0.2	0.1	0.9



Word2Vec (CBOW)

give a talk at the

give $\begin{matrix} 0 & \cdots & 1 & \cdots & \cdots & 0 \end{matrix}$
a $\begin{matrix} 0 & \cdots & \cdots & 1 & \cdots & 0 \end{matrix}$
at $\begin{matrix} 0 & \cdot & 1 & \cdots & \cdots & 0 \end{matrix}$
the $\begin{matrix} 0 & \cdots & \cdots & 1 & \cdots & 0 \end{matrix}$

$(2 * \text{window_size}, V)$

w_1
 (V, N)

N
 $0.1 \cdot 0.4 \cdots 0.7 \cdot 0.9$
 $0.3 \cdot 0.1 \cdots 0.5 \cdot 0.8$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$
 $(4, N)$

AVG () + Relu ()

$\overline{\quad}$

Re-shaping
the values

$0.1 \cdot 0.1 \cdots 7 \cdot 0.4$

$(1, N)$

Re-shaping
the rows

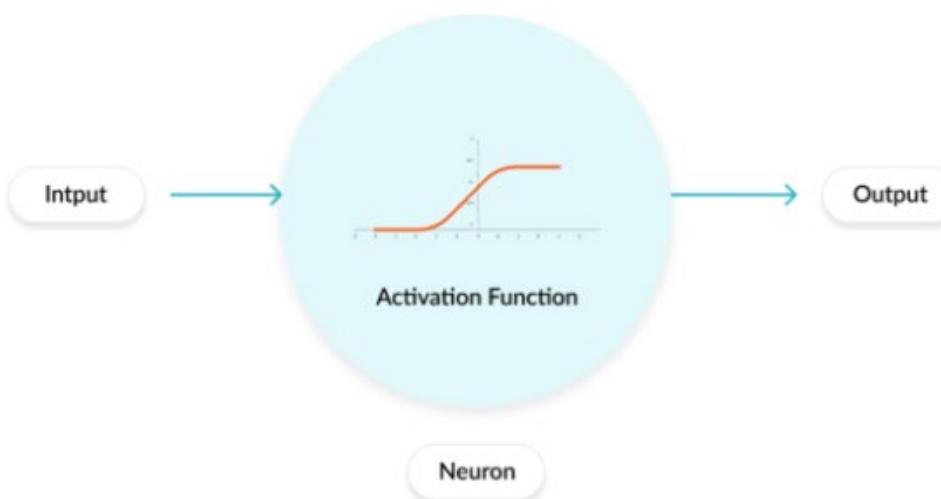
hint **talk** rush ... anything
 \circ \bullet \circ \cdots \circ
 $(1, V)$



Activation Function

- **Activation Function**

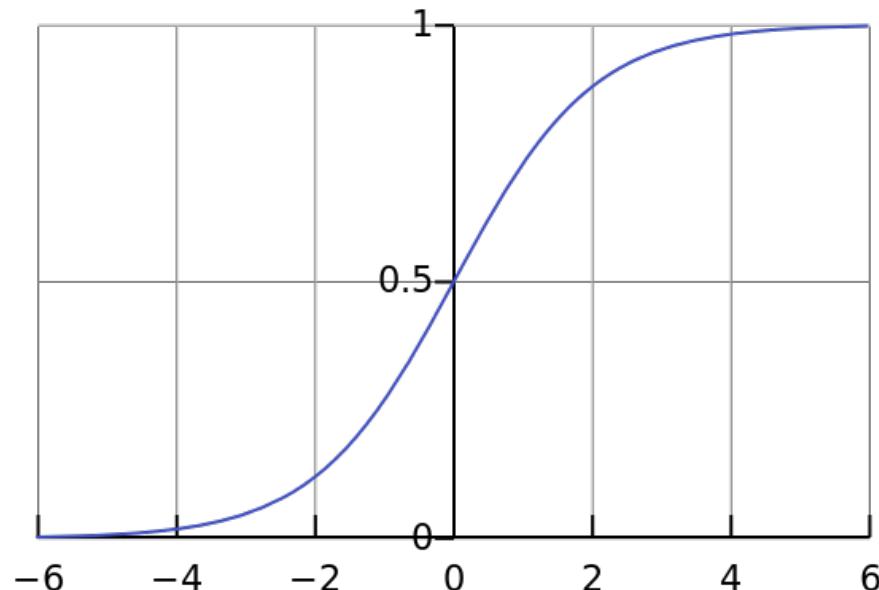
- Smoother decision function is expected
- Activations bound in $(0,1)$
- Support backpropagation
- Need to be non-linear



Activation Function

- **Sigmoid** function
 - smooth output between 0 and 1
 - interpreted as a probability of “Yes”

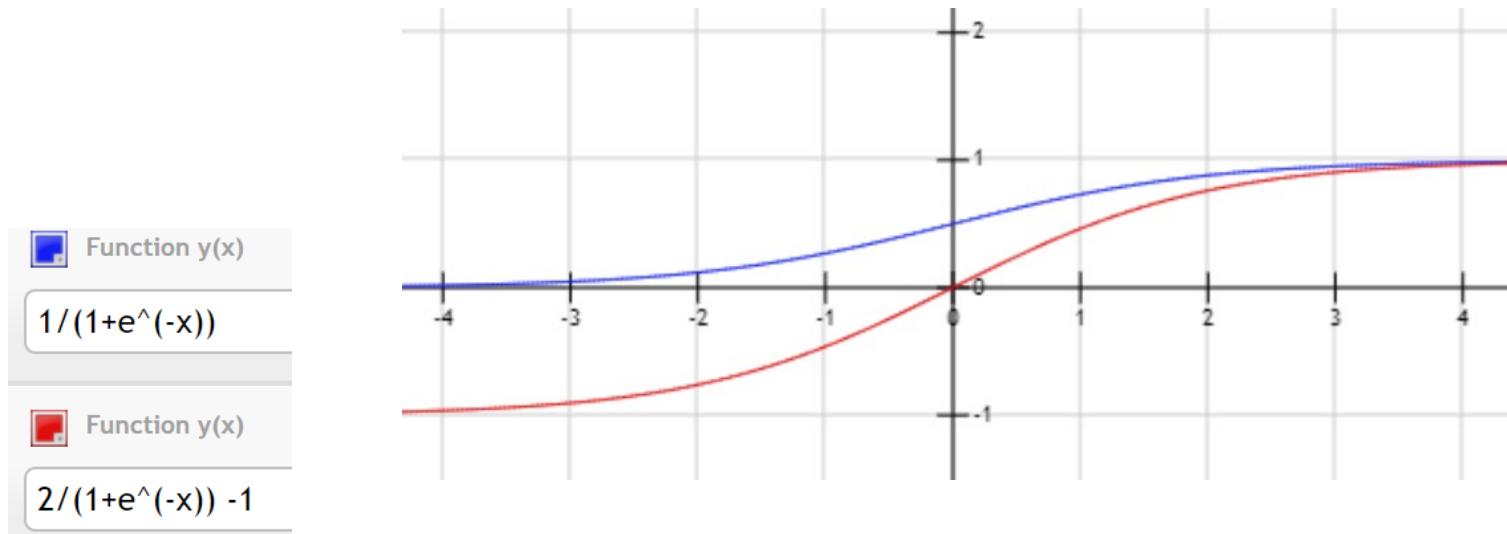
$$S(x) = \frac{1}{1 + e^{-x}}$$



(Image from Wikipedia)

Activation Function

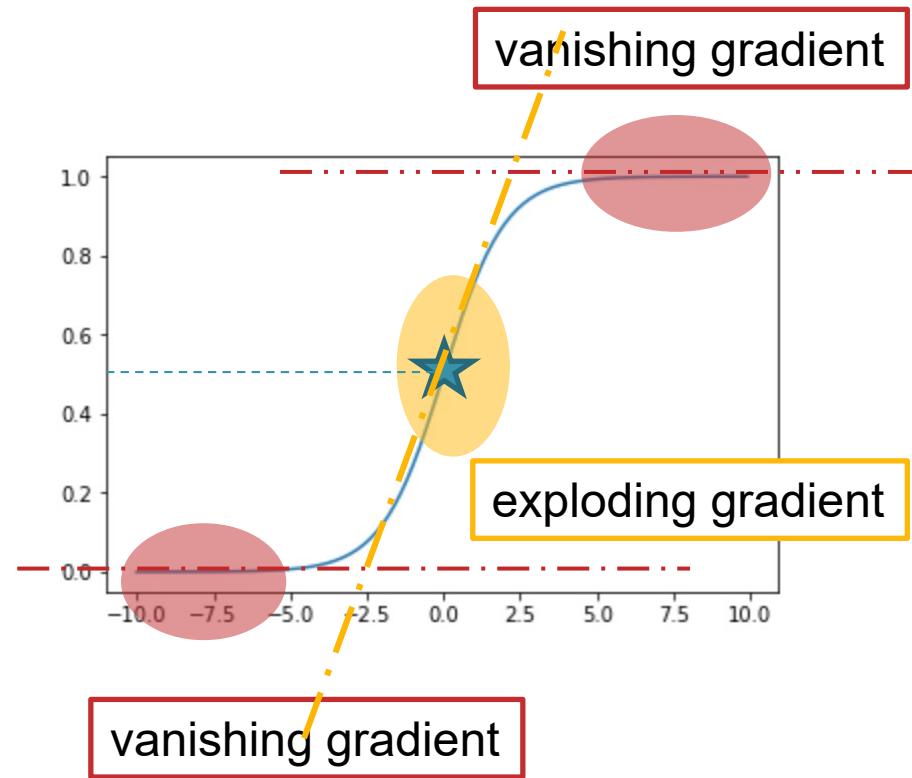
- Tanh Function
 - smooth output between -1 and 1
 - 0 centroid
 - interpreted as a set of “weights”



$$\tanh(x) = 2 \text{ sigmoid}(2x) - 1$$

Activation Function (Sigmoid)

```
3 import numpy as np  
4 import matplotlib.pyplot as plt  
5  
6 def sigmoid(x):  
7     return 1/(1+np.exp(-x))  
8  
9 # Generate points from -10 to +10,  
10 # in steps of 0.1  
11 x = np.arange(-10, 10, 0.1)  
12 y = sigmoid(x)  
13  
14 # Plot the graph.  
15 plt.plot(x, y)  
16 plt.show()
```



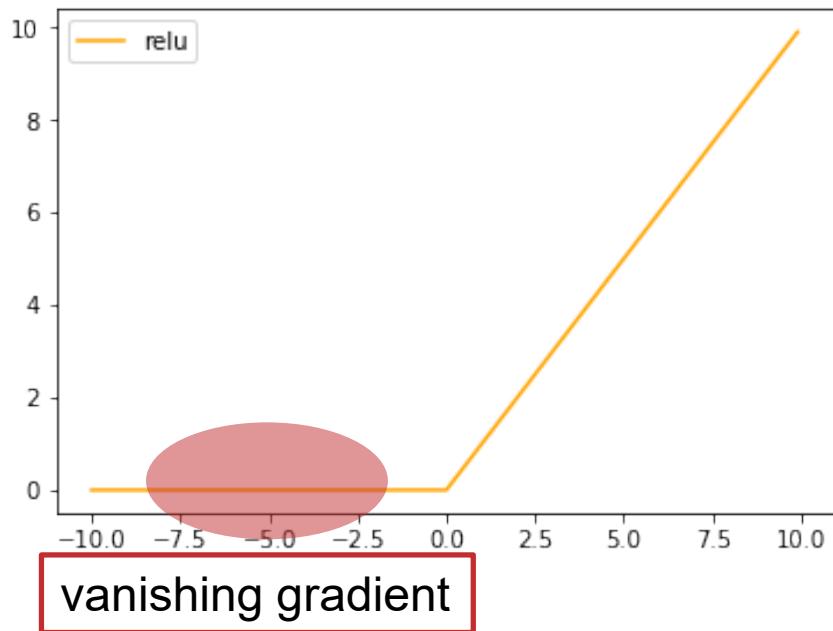
Activation Function (ReLU)

- **Intuitions can be useless for NN**
 - Smoother decision function is expected
 - Activations bound in a range
 - “0” centroid



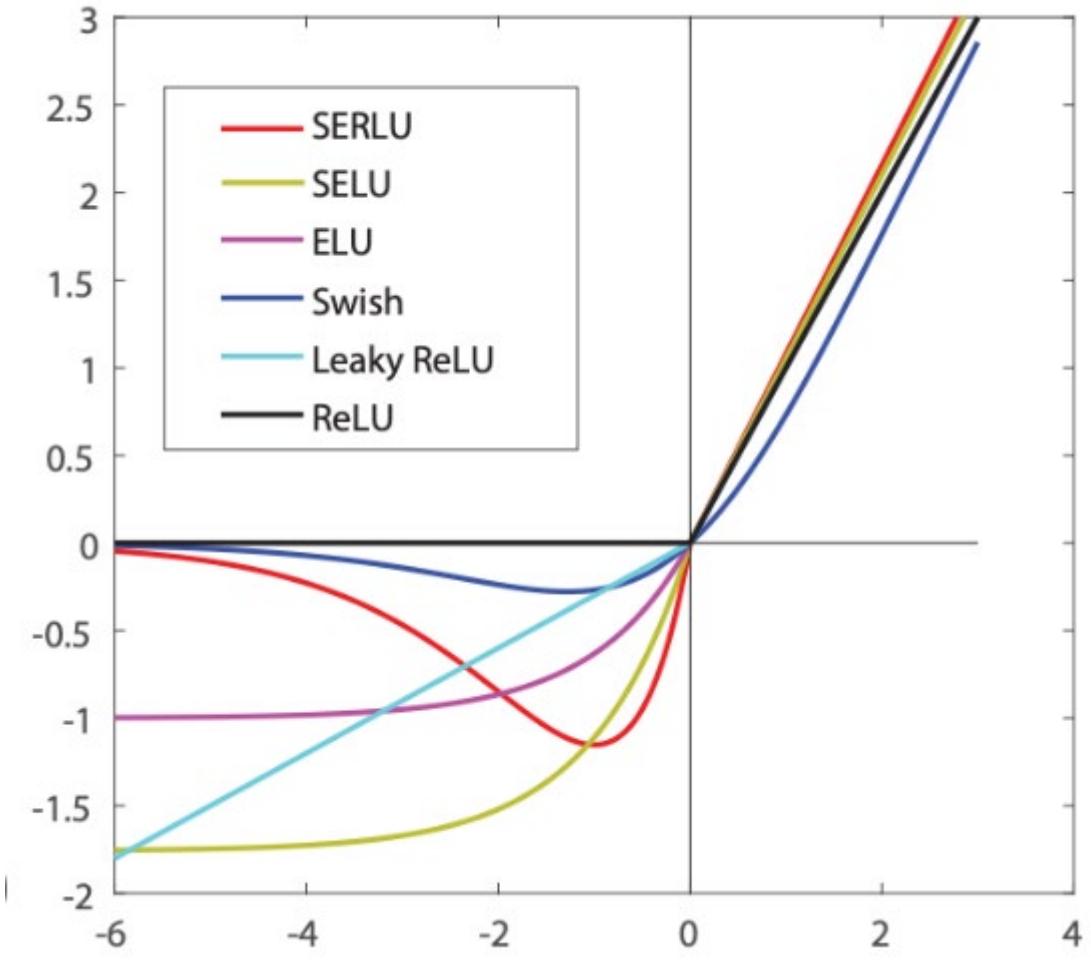
- **All you need is just**
 - Simple
 - Fast

$$A(x) = \max(0, x)$$



Activation Function (ReLU)

- ReLU's Family



Word2Vec (CBOW)

give a talk at the

give
a
at
the

0 ··· 1 ··· 0
0 ··· 1 ··· 0
0 · 1 ··· 0
0 ··· 1 ··· 0

($2 * \text{window_size}, V$)

w_1

N

0.1 · 0.4 · 0.7 · 0.9
0.3 · 0.1 · 0.5 · 0.8
0.4 · 0.6 · 0.9 · 0.2
0.4 · 0.6 · 0.9 · 0.2

($4, N$)

Dense Layer

Re-shaping
the rows

AVG () + Relu ()

Re-shaping
the values

0.1 · 0.1 · 7 · 0.4

(1, N)

Re-shaping
the columns

w_2

(N, V)

Re-shaping
the Probs

Softmax ()

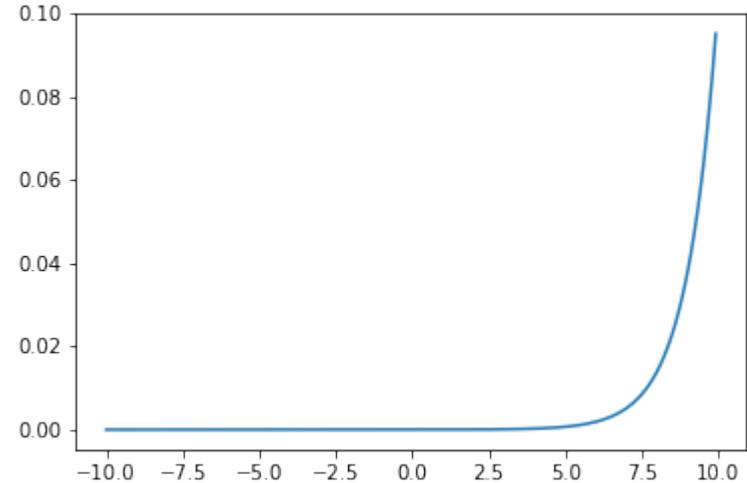
hint
talk
rush
...
anything

(1, V)

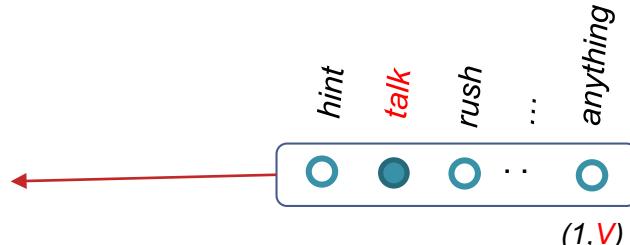
Last Layer Activation Function

- SoftMax

```
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 def softmax(x):
7     return np.exp(x) / np.sum(np.exp(x), axis=0)
8
9 x = np.arange(-10, 10, 0.1)
10 y = softmax(x)
11
12 plt.plot(x,y)
13 plt.show()
```



Vector of x_i as Input
Never be a single x

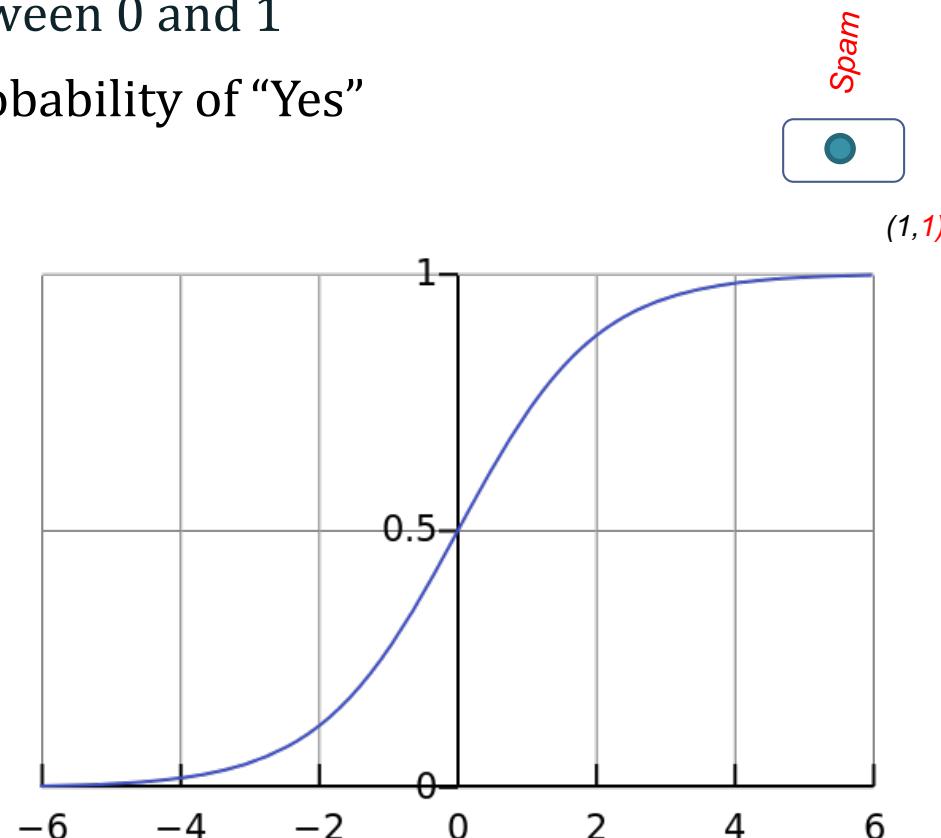


Last Layer Activation Function

- **Sigmoid** function
 - smooth output between 0 and 1
 - interpreted as a probability of “Yes”

A single x as input
Never be a vector of x_i

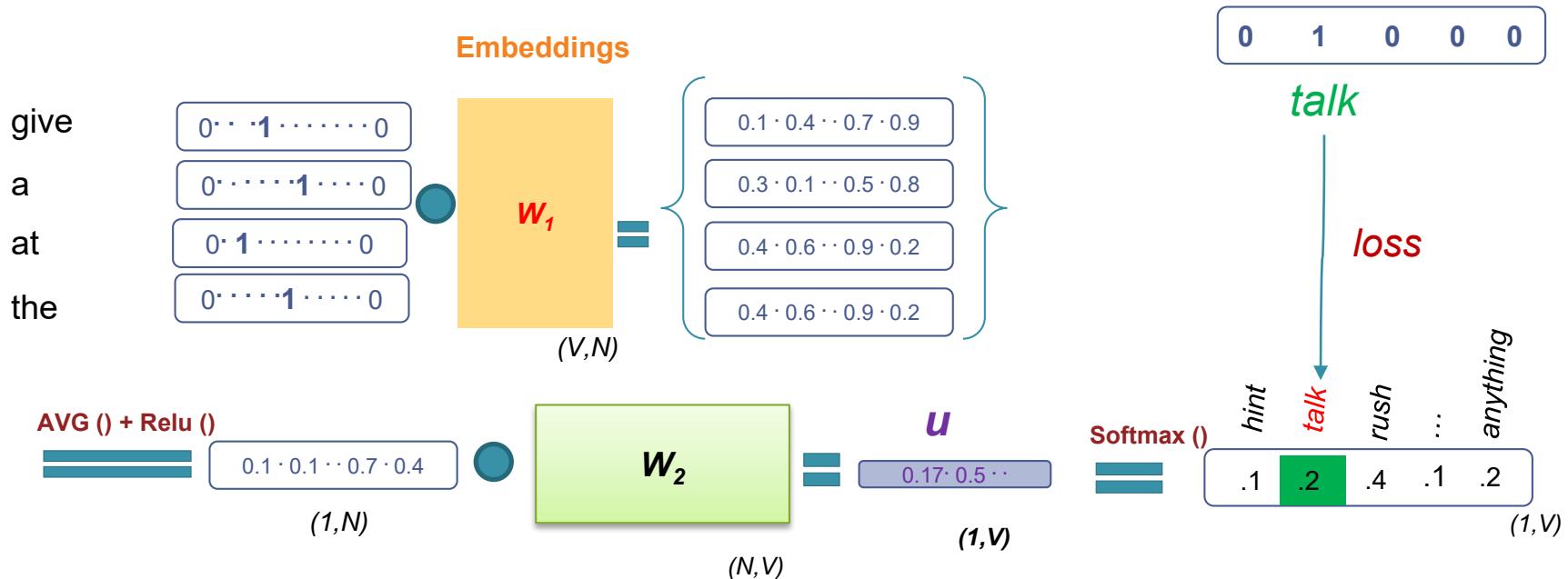
$$S(x) = \frac{1}{1 + e^{-x}}$$



(Image from Wikipedia)

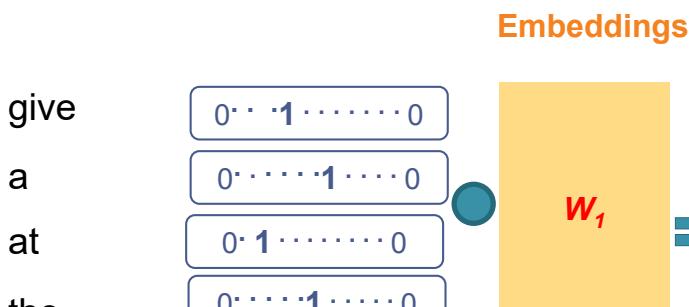
Word2Vec (CBOW)

Loss Function



Loss Function

(Categorical) Cross Entropy Loss Function



$$\text{AVG () + Relu ()} = \frac{0.1 \cdot 0.1 \cdots 0.7 \cdot 0.4}{4} = 0.1$$

$$\mathbf{W}_2 = \mathbf{u} = \begin{matrix} 0.17 & 0.5 \cdots \end{matrix}$$

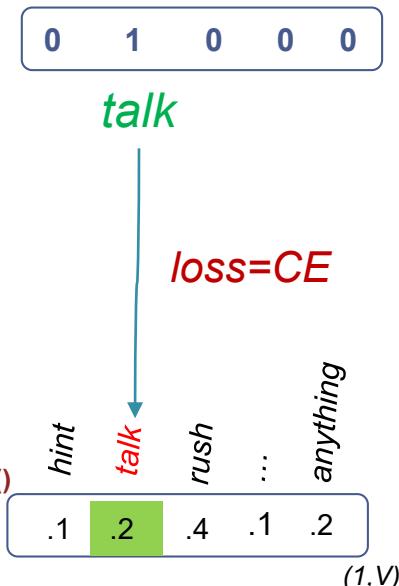
$$\text{Loss} = - \sum_{c=1}^M y_{o,c} \log(p_{o,c})$$

! Note

- M - number of classes (dog, cat, fish)
- log - the natural log
- y - binary indicator (0 or 1) if class label c is the correct classification for observation o
- p - predicted probability observation o is of class c

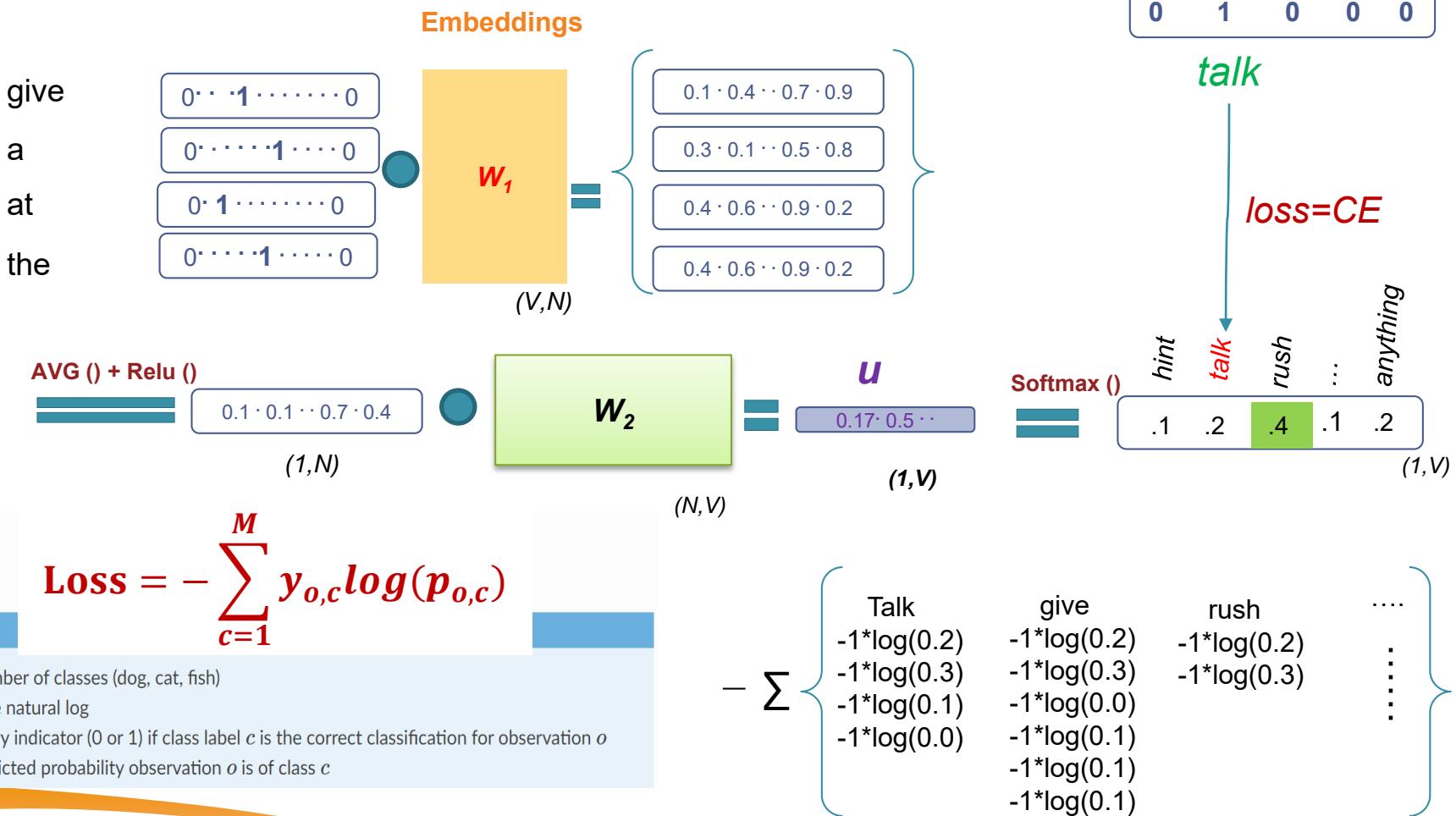
Talk

$$- 0 * \log(0.1) - 1 * \log(0.2) - 0 * \log(0.4) - 0 * \log(0.1) - 0 * \log(0.2)$$



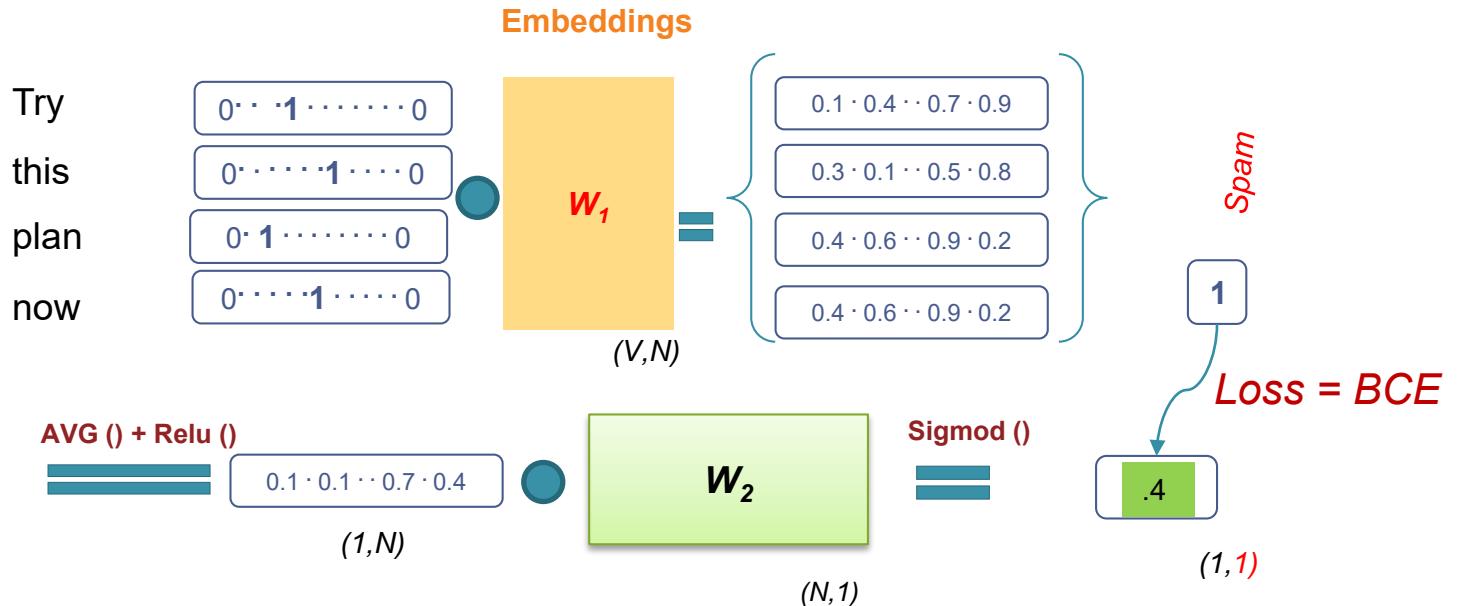
Loss Function

(Categorical) Cross Entropy Loss Function



Not For CBOW

Binary Cross Entropy Loss Function

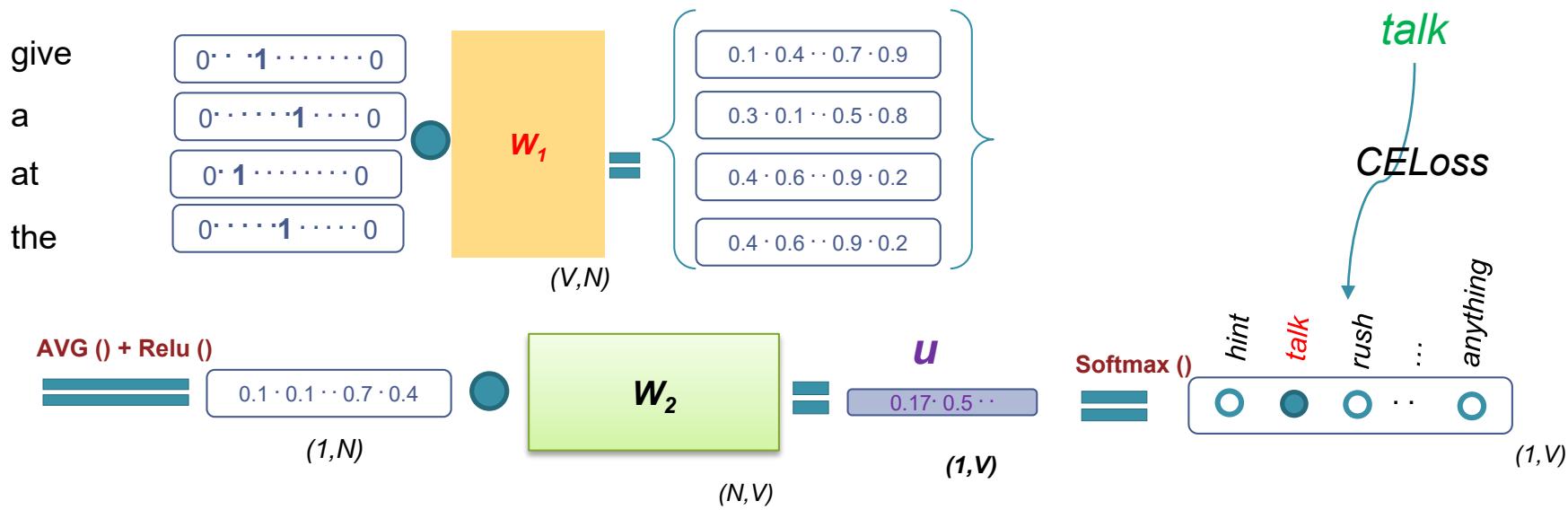


$$BCE = - \sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1-t_1) \log(1-s_1)$$

$$= - \sum \left\{ \begin{array}{ll} \text{spam} & \text{not spam} \\ -1 * \log(0.4) & -(1-1) * \log(1-0.4) \\ -1 * \log(0.3) & -(1-1) * \log(1-0.3) \\ -1 * \log(0.1) & -(1-1) * \log(1-0.1) \\ -1 * \log(0.0) & -(1-1) * \log(1-0.0) \end{array} \right\}$$

Word2Vec (CBOW)

Backpropagation SGD



$$W_1(\text{new}) = W_1(\text{old}) - \frac{\partial \text{lost}}{\partial w_1} * lr$$

$$W_2(\text{new}) = W_2(\text{old}) - \frac{\partial \text{lost}}{\partial w_2} * lr$$

Backpropagation

SGD vs Adam

```
# Vanilla SGD  
x += - learning_rate * dx
```

x is a vector of parameters and
dx is the gradient

Well generalised
but slower
Used together with
momentum

Faster
but sometimes
not converging

```
# Adam  
m = beta1*m + (1-beta1)*dx  
v = beta2*v + (1-beta2)*(dx**2)  
x += - learning_rate * m /  
(np.sqrt(v) + eps)
```

x is a vector of parameters and
dx is the gradient
m is the smoothen gradient
v is the ‘cache’ used to normalize **x**
eps is smoothing term (1e-4 to 1e-8)
beta1, beta2 are hypers (0.9, 0.999)

Backpropagation

- **Learning Rate**
 - Fixed Learning Rate based on Experience
 - Vanilla model: MLP/RNN/CNN $10^{-2} \sim 10^{-3}$
 - Typical Models: LSTM/CNN 10^{-3}
 - Complex Models: BERT 10^{-5}

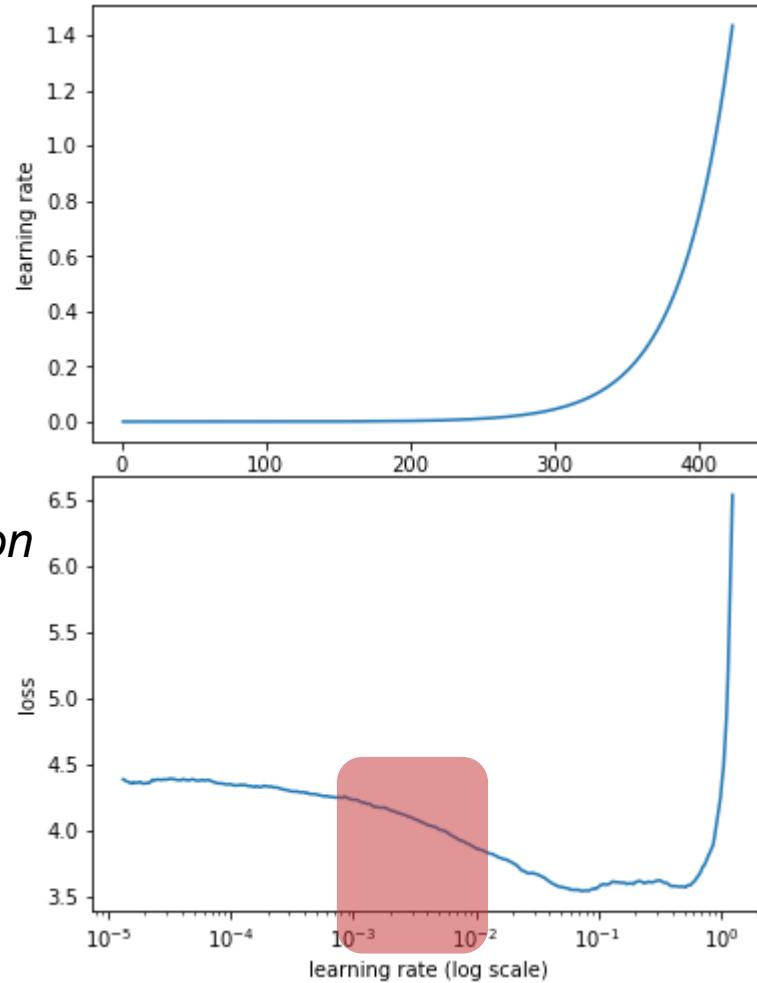
Backpropagation

- Fixed Learning Rate
 - Estimate Fixed Learning Rate

Increase lr as iterations going up

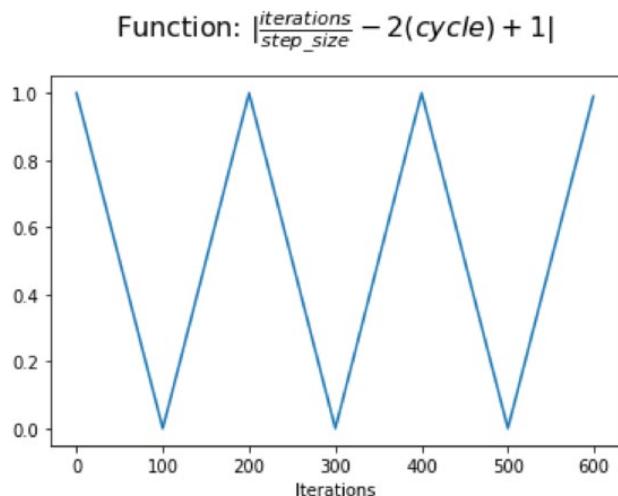
save and plot the *loss*
as per $lr == \text{as per iteration}$

get the lr
fastest decrease in the loss

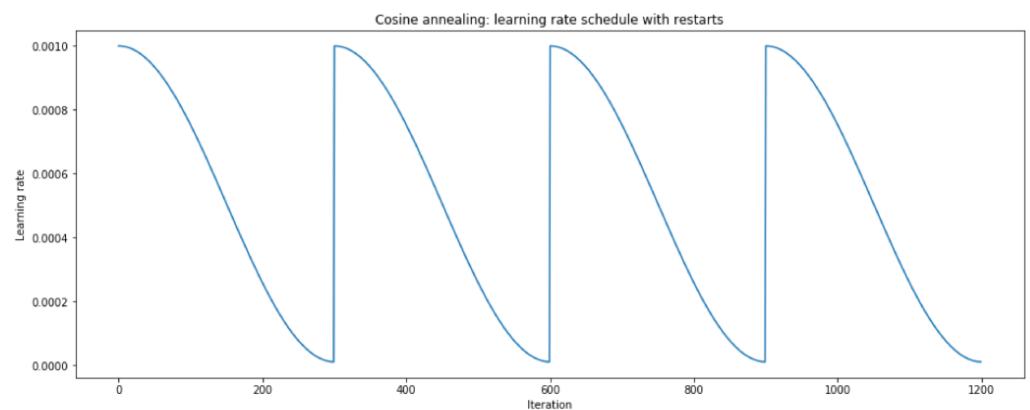
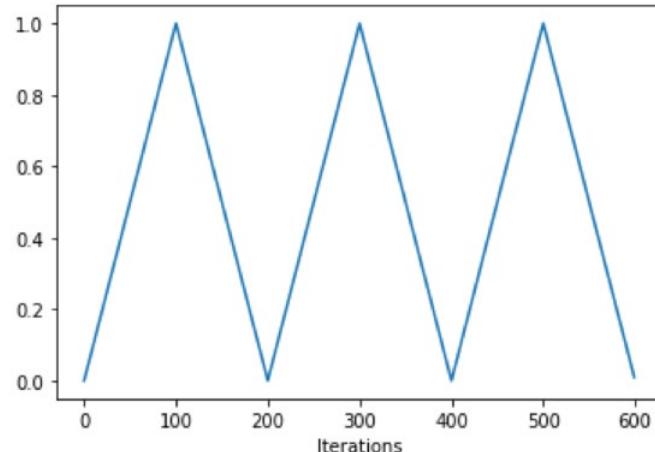


Backpropagation

- **Dynamic Learning Rate**
 - Estimate Initial Learning Rate
 - Program Dynamic Learning Rate



Function: $1 - |\frac{\text{iterations}}{\text{step_size}} - 2(\text{cycle}) + 1|$



$$\eta_t = \eta_{\min}^i + \frac{1}{2} (\eta_{\max}^i - \eta_{\min}^i) \left(1 + \cos\left(\frac{T_{\text{current}}}{T_i} \pi\right) \right)$$

Annealing,
i.e. taking a
partial step

Word2Vec (CBOW)

give a talk at the

give $\begin{matrix} 0 \cdots 1 \cdots \cdots \cdots 0 \end{matrix}$
 a $\begin{matrix} 0 \cdots \cdots 1 \cdots \cdots 0 \end{matrix}$
 at $\begin{matrix} 0 \cdot 1 \cdots \cdots \cdots 0 \end{matrix}$
 the $\begin{matrix} 0 \cdots \cdots 1 \cdots \cdots 0 \end{matrix}$

($2 * \text{window_size}, V$)

w_1 (V, N)

N
 $0.1 \cdot 0.4 \cdots 0.7 \cdot 0.9$
 $0.3 \cdot 0.1 \cdots 0.5 \cdot 0.8$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$
 $0.4 \cdot 0.6 \cdots 0.9 \cdot 0.2$

($4, N$)

talk

CELoss

Re-shaping
the rows

Re-shaping
the values

Re-shaping
the columns

Re-shaping
the Probs

AVG () + Relu ()

$0.3 \cdot 0.43 \cdots 0.75 \cdot 0.5$

($1, N$)

w_2

(N, V)

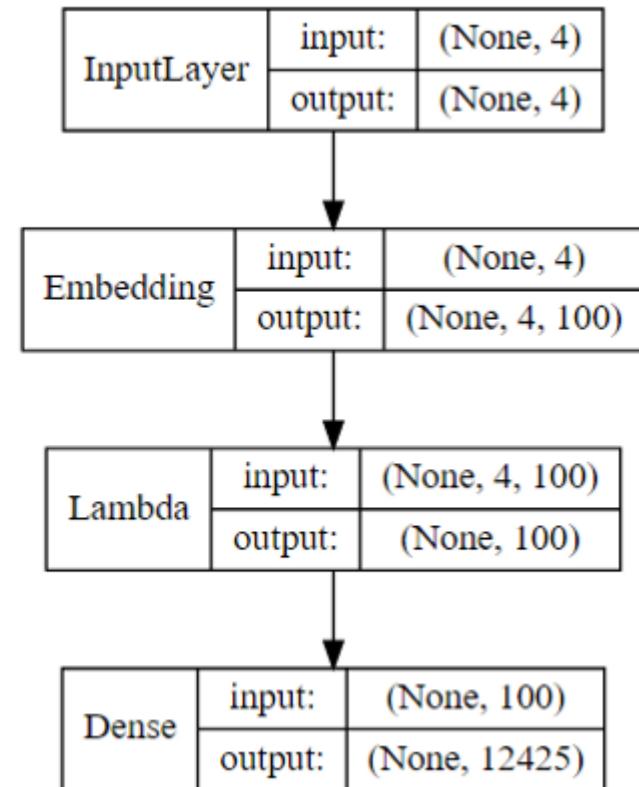
Softmax ()

hint talk rush ... anything

($1, V$)

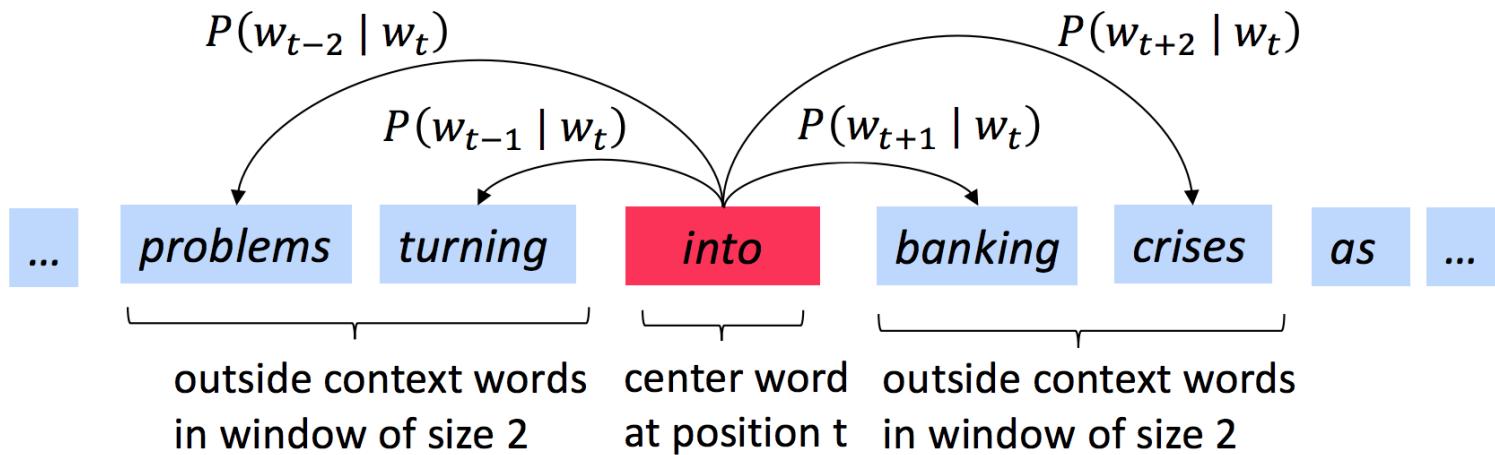
CBOW Model Summary

Layer (type)	Output Shape	Param #
=====		
embedding_1 (Embedding)	(None, 4, 100)	1242500
lambda_1 (Lambda)	(None, 100)	0
dense_1 (Dense)	(None, 12425)	1254925
=====		
Total params:	2,497,425	
Trainable params:	2,497,425	
Non-trainable params:	0	



Word2Vec (Skipgram)

Task: Iterate through each word with a given window; for each word predict the context words within the window



(E.g. from Manning (2018) Stanford cs224n course)

Word2Vec (Skipgram)

Sentence: language users never choose words randomly , and language is essentially non-random .

In-/Outputs:

```
[  
    ('never',      ['language', 'users', 'choose', 'words']),  
    ('choose',     ['users', 'never', 'words', 'randomly']),  
    ('words',      ['choose', 'words', ',', 'and']),  
    ('randomly',   ['words', 'randomly', 'and', 'language'])  
]
```

Word2Vec (Skipgram)

Sentence: language users never choose words randomly , and language is essentially non-random .

Windows:

```
['language', 'users', 'never', 'choose', 'words']
```

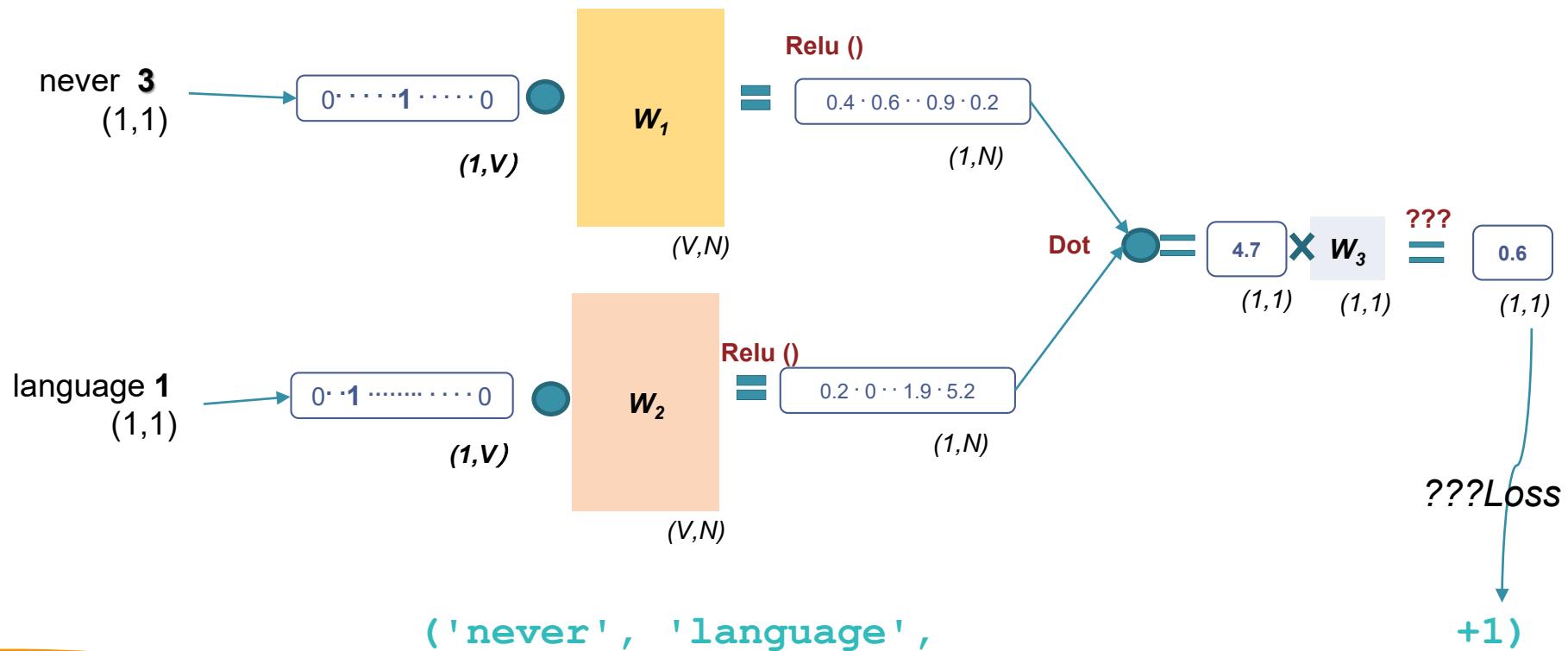
```
('never', 'language', 1),  
('never', 'users', 1),  
('never', 'choose', 1),  
('never', 'words', 1),  
('never', ',', 0),  
('never', 'non-random', 0),  
('never', 'is', 0),  
('never', 'is', 0)
```

aka.
**negative
sampling**

Word2Vec (Skipgram)

Sentence: language (1) users (2) never (3) choose (4) words
 (5) randomly (6), (7) and (8) language (1) is (10)
 essentially (11) non-random (12). (13)

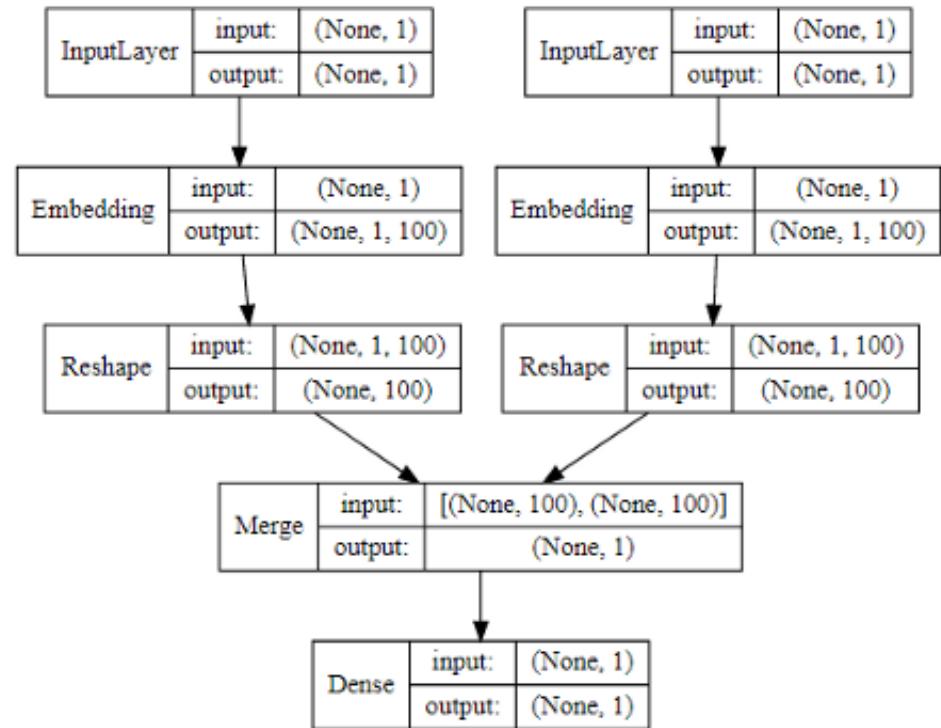
['language' , 'users' , 'never' , 'choose' , 'words']



Word2Vec (Skipgram)

- Skipgram Model

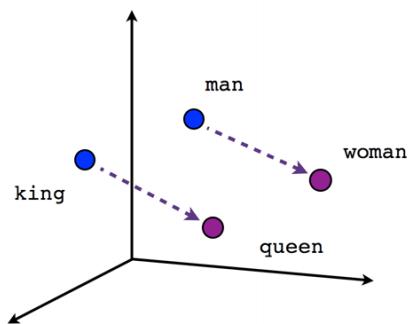
Layer (type)	Output Shape	Param #
merge_2 (Merge)	(None, 1)	0
dense_3 (Dense)	(None, 1)	2
Total params:	2,485,002	
Trainable params:	2,485,002	
Non-trainable params:	0	



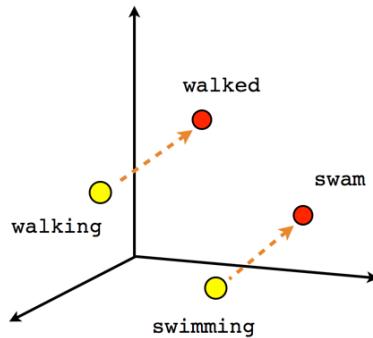
How to Choose Context?

- **Different contexts lead to different embeddings**
- **Small context window:** more syntax related
 - *I like...*
 - *She likes ...*
- **Large context window:** more semantics related
 - *stackoverflow great website for programmers*

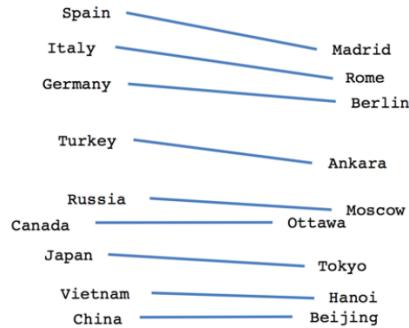
Properties of Word Embeddings



Male-Female



Verb tense



Country-Capital

Ingredients

Corpus of text	As large as possible
Annotations	0
Initialize weights (aka Embeddings)	1x per word
Deep Learning Model	1x
Cost Function	Appropriately
GPU	Lotsa of it

When to use pre-trained embeddings?

- Generally, when you don't have much training/annotated data
- **Useful:** Use as inputs to model for classification task, e.g. tagging, parsing, ranking (based on similarity)
- **Less Useful:** Machine Translation / Sequence generating tasks
- **Not Useful:** Generic Language Modeling, for those, we have sentence embeddings...

Embedding Bias

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{computer programmer}} - \overrightarrow{\text{homemaker}}.$$

	“He” Occupations	“She” Occupations
Cosine Similarity	[“retired”, “doctor”, “teacher”, “student”, “miller”, “assistant”, “lawyer”, “baker”, “judge”, “governor”, “butler”]	[“doctor”, “ teacher ”, “ nurse ”, “actress”, “student”, “miller”, “reporter”, “retired”, “lawyer”, “actor”, “artist”]
Inner Product Similarity	[“cleric”, “photographer”, “skipper”, “chaplain”, “accountant”, “inspector”, “rector”, “investigator”, “psychologist”, “treasurer”, “supervisor”]	[“librarian”, “housekeeper”, “nanny”, “accountant”, “sheriff”, “envoy”, “tutor”, “salesman”, “butler”, “footballer”, “solicitor”]

Summary Word2Vec

Steps

1. Define task that we want to predict
2. Go through each sentence and create the task's in-/outputs
3. Iterate through task's I/O, put the inputs through the embeddings and models to create predictions
4. Measure cost of the predicted and expected output
5. Update embedding weights accordingly (*backprop)
6. Repeat Step 3-5 until desired.

Neither **GloVe** or **Word2Vec** has been shown to provide definitively better results rather they should both be evaluated for a given dataset.

Train Word2Vec from “scratch”

CBOW AND SKIPGRAM

SENTIMENT SUMMARIZATION & VISUALIZATION

Sentiment Summarization

(a) Extractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

(b) Abstractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

Extractive vs Abstractive Summary

Source: <https://devopedia.org/text-summarization>

Contrastive Opinion Summarization

Table 3: Sample contrastive sentence pairs

No	Positive	Negative
1	oh ... and file transfers are fast & easy .	you need the software to actually transfer files
2	i noticed that the micro adjustment knob and collet are well made and work well too.	the adjustment knob seemed ok, but when lowering the router, i have to practically pull it down while turning the knob.
3	the navigation is nice enough , but scrolling and searching through thousands of tracks , hundreds of albums or artists , or even dozens of genres is not conducive to save driving .	difficult navigation - i wo n't necessarily say " difficult ," but i do n't enjoy the scrollwheel to navigate .
4	i imagine if i left my player untouched (no backlight) it could play for considerably more than 12 hours at a low volume level.	there are 2 things that need fixing first is the battery life. it will run for 6 hrs without problems with medium usage of the buttons.

- a set of contrastive sentence pairs

Examples of Contrastive sentence pairs

Source: Kim, H. D., & Zhai, C. (2009, November). Generating comparative summaries of contradictory opinions in text. In Proceedings of the 18th ACM conference on Information and knowledge management (pp. 385-394).

<https://dl.acm.org/doi/pdf/10.1145/1645953.1646004>

Sentiment Visualization

Information visualization techniques take advantage of our visual information processing ability by creating visual representations of the large dataset. As a result, various interesting patterns, trends, and outliers can be much more easily observed. A fundamental challenge of designing any information visualization system arises from the human and display limitations. Our limited perceptual and cognitive abilities must be taken into account in the design of an effective visualization. Furthermore, the limited display size means we have to be selective in showing visualizations. There are also some design challenges that are very specific to sentiment visualization for social media text data. One particular challenge arises from the noisy nature of social media text. As a consequence of noisy text, the results of text mining and summarization methods can be inaccurate.

Is it tough to read all these words without visualizations? 😊

Sentiment Visualization

- **[Why Visualization?]** Information visualization techniques take advantage of our visual information processing ability by creating visual representations of the large dataset. As a result, various interesting patterns, trends, and outliers can be much more **easily observed**.
- **[Human and Display Limitations]** A fundamental challenge of designing any information visualization system arises from the human and display limitations. Our limited perceptual and cognitive abilities must be taken into account in the design of an effective visualization. Furthermore, the limited display size means we have to be selective in showing visualizations.
- **[Social Media-specific Design Issues]** There are also some design challenges that are very specific to sentiment visualization for social media text data. One particular challenge arises from the noisy nature of social media text. As a consequence of noisy text, the results of text mining and summarization methods can be inaccurate.



Sentiment Visualization - Word Clouds

- Visualizes the single word or short phrases within social media text data based on frequency of occurrence.
- Quick and low-cost overview of the common words and phrases within the data
- However, there are issues such as co-reference resolution & word sense disambiguation (especially when there are multiple entities in the data)

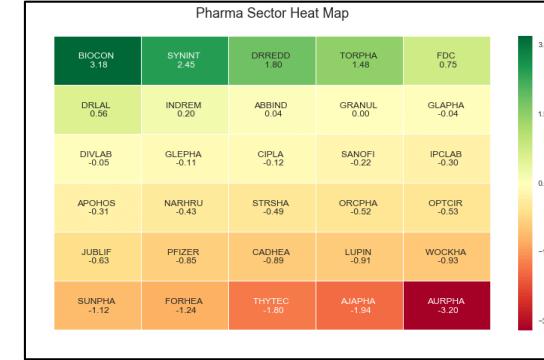
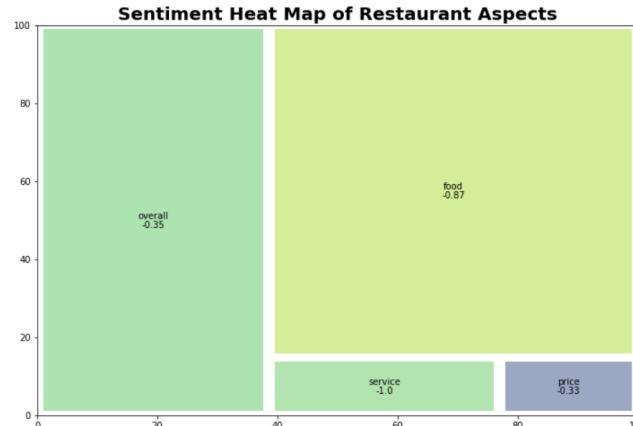


Quick Word Cloud Generation in Python

Link: <https://www.geeksforgeeks.org/generating-word-cloud-python/>

Sentiment Heat Map

- Sentiment Heat Map is a graphical way to visualize social media data in the form of a color scheme (e.g. a gradient color ranging from green to red)
- Commonly used to visualize different aspects of an entity and the associated sentiment score
- It can be cognitively challenging for humans to interpret a heat map, especially given that social media data is inherently hard to cluster to demonstrate an obvious trend



Quick Heat Map Generation in Python

Link: <https://blog.quantinsti.com/creating-heatmap-using-python-seaborn/>

Sentiment Visualization - Time Series of Sentiment

- The visual interface of the Time Series can be used to show a sentiment timeline that aggregates sentiment responses into four different categories: positive, negative, controversial, or neutral
- In addition, the system shows the volume of tweets over time, the keywords over time, and the detailed twitter messages
- This allows the analysts to monitor the variations in sentiments when different policies are being mentioned in real time

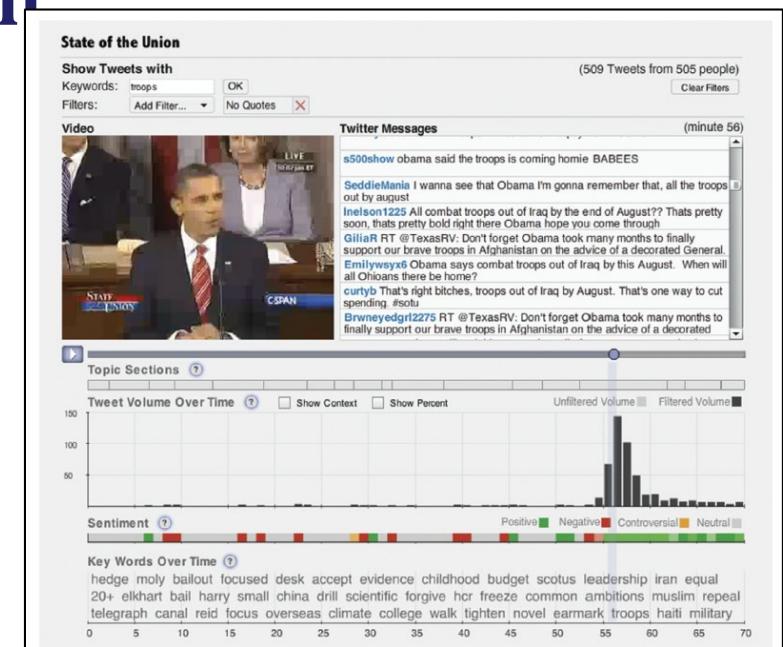
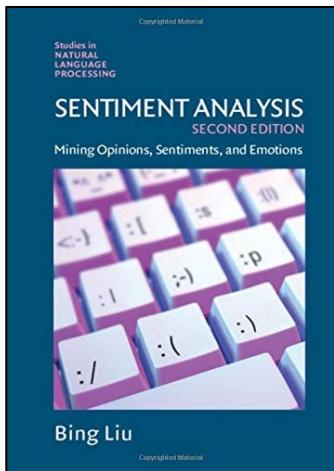


FIG. 11.2

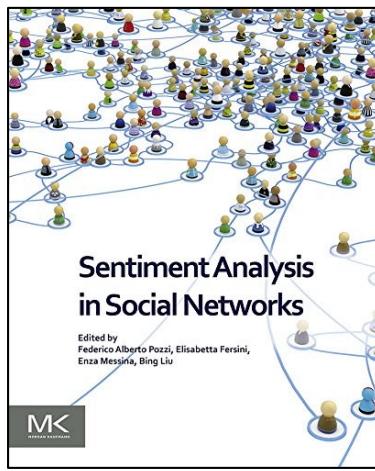
The Vox Civitas user interface [42], showing the Twitter messages along with the volume of tweets, the trend of overall sentiment, and keywords over time.

Resources



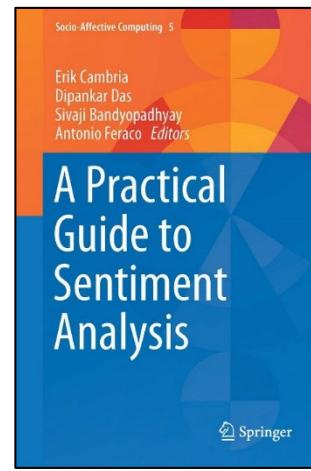
Sentiment Analysis: Mining Opinions, Sentiments, and Emotions (Studies in Natural Language Processing) 2nd Edition (2020)

Note: Read Chapter 1, 2 and 9



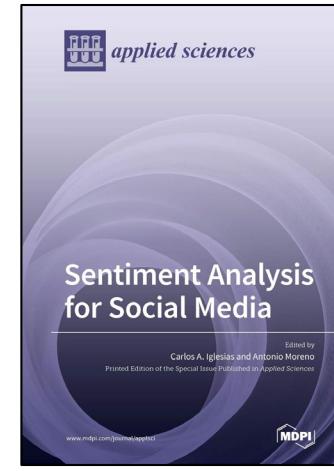
**Sentiment Analysis in Social Networks
1st Edition (2016)**

Note: Read Chapter 1, 7, 9 and 11



**A Practical Guide to Sentiment Analysis (Socio-Affective Computing, 5)
1st ed. 2017 Edition**

Note: Read Chapter 2, 4 and 7



**Sentiment Analysis for Social Media
(2020)**

Note: Read the case study "Using Social Media to Identify Consumers' Sentiments towards Attributes of Health Insurance during Enrollment Season"

References

1. Role, Franois, and Mohamed Nadif. "Handling the impact of low frequency events on co-occurrence based measures of word similarity." Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR-2011). Scitepress. 2011.
2. Yogan, Jaya Kumar, et al. "A review on automatic text summarization approaches." Journal of Computer Science 12.4 (2016): 178-190
3. Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018)
4. Radford, Alec, et al. "Improving language understanding by generative pre-training." URL <https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language-understanding-paper.pdf> (2018).
5. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
6. Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." arXiv preprint arXiv:1801.06146 (2018).
7. Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

EBA5004 PLP – New Media and Sentiment Mining

Entity & Aspect Mining

Dr. Fan Zhenzhen
NUS-ISS
National University of Singapore



Module Objectives

At the end of this module, you will be able to

- Identify common tasks in aspect-based sentiment analysis
- Extract key entities and aspects from opinions for closer analysis of sentiments and their targets
- Use common and LM-based approaches to perform entity/aspect extraction

Agenda

- The opinion target - entity and its aspects
- Tasks in ABSA
- Approaches in entity extraction
- Approaches in aspect extraction
 - Frequency-based approach
 - Lexical-syntactic approach
 - Supervised-learning approach
 - Deep-learning approach
 - Zero-shot approach with LLMs

Entities, Aspects & ABSA

The Opinion Target and its Aspects

- An *opinion* consists of a *sentiment* (positive or negative) and a *target* (of opinion).
- Detect subjective opinion and determine its polarity is often the first step in sentiment mining.
- A common follow-up question is : what exactly do people like or not like?
- Getting the right answer requires **finer-grained** analysis, identifying opinion *target*, which can be an *entity*, or its *aspects*.
- Crucial for applications like product review analysis.

Definition of Opinion

- Recall that opinion is formally defined as a quadruple (g, s, h, t) , where
 - g is the sentiment **target**,
 - s is the **sentiment** of the opinion about the target g ,
 - h is the **opinion holder** (the person or organization who holds the opinion),
 - and t is the **time** when the opinion is expressed..
- In the case of entity and aspect mining, there is an additional dimension – **aspect/ feature a** .
- Extricate quintuple (e, a, s, h, t) with entity e and aspect a together representing the opinion target.

Opinion Target Extraction

- Examples:
 - *Although the **service** is not that great, I still love this **restaurant**.*
 - *The **iPhone's call quality** is good, but its **battery life** is short.*
- **Entities** - names of products, services, individuals, events, and organizations
- **Aspects** – the attributes and components of entities
- Identifying them from text data is called *opinion target extraction*.
- Information Extraction (IE) tasks.

Typical tasks for ABSA

- Identify subjective/opinionated sentences
- Find and extract entities and aspects that have been commented on by an opinion holder
- Determine whether the opinions are positive, negative or neutral
- Group entity/aspect synonyms
- Produce an aspect-based opinion summary of multiple reviews

ABSA subtasks

S: The **drinks** are always **well made** and **wine selection** is **fairly priced**.

a_1	o_1	a_2	o_2
-------	-------	-------	-------

Subtask	Input	Output	Task Type
Aspect Term Extraction(AE)	S	a_1, a_2	Extraction
Opinion Term Extraction(OE)	S	o_1, o_2	Extraction
Aspect-level	S + a_1	s_1	Classification
Sentiment Classification(ALSC)		s_2	
Aspect-oriented	S + a_1	o_1	Extraction
Opinion Extraction(AOE)		o_2	
Aspect Term Extraction and Sentiment Classification(AESC)	S	(a_1, s_1), (a_2, s_2)	Extraction & Classification
Pair Extraction(Pair)	S	(a_1, o_1), (a_2, o_2)	Extraction
Triplet Extraction(Triplet)	S	(a_1, o_1, s_1), (a_2, o_2, s_2)	Extraction & Classification

How?

- The methods and features used for recognizing **entities** and **aspects** are usually different due to their individual specific characteristics.
- Main idea:
 - An opinion always has a target.
 - The target is an aspect or an entity.
 - It's usually a noun or noun phrase (but not always).
 - => often leverage on syntactic structures to discover opinion and target relationships

Entity Extraction

Entity Extraction

- Similar to the classic problem of **Named Entity Recognition (NER)** in NLP.
- Main approaches
 - Rule-based
 - Supervised statistical machine learning (e.g., HMMs, CRFs, etc.)
 - Semi-supervised approaches that only require some unambiguous seed entity names to find all entities of the same type (e.g., PU learning, Bayesian Sets, etc)
- Mature tools are available to perform NER (GATE, NLTK, Stanford NER, etc.), but not for entity extraction.
- For sentiment analysis, an additional step is usually needed to group the entities into synonyms, as the entity can be mentioned in various ways.

Differences from traditional NER

- In traditional NER, the objective is to recognize all named entities of certain types in a corpus, e.g., names of people, names of organizations, etc.
- In sentiment mining, the interest is often to find the mentions of a set of desired entities, e.g. the company's own products, its competitors' products, etc.
- Solved in two steps:
 - Identify all **entity mentions** or **entity expressions** of the interested set of entities.
 - For each **entity mention/expression**, determine to which **entity** it belongs, aka *entity linking* or *entity disambiguation/resolution*

Entity Resolution

- Two name ambiguity problems:
 - Polysemy:
 - *Apple* for *Apple Inc.* (the maker of iPhone and iPad), or *Apple Daily* (a Hong Kong newspaper)?
 - Synonymy:
 - *National University of Singapore* vs. *NUS*,
 - *Volkswagen* vs. *Vwagen*,
 - *Singapore* vs. *the little red dot*,
 - *sound/voice/sound quality* etc.
- When the set of target entities is available, the task is to cluster entity mentions, and map them to target entities.

Scope of entities in different types of text data

- Entity-focused corpora – online reviews of products and services.
 - Entity information can be obtained from meta-data
 - Mentions of other entities might need to be identified for comparison opinion mining
- Domain-focused corpora – forum discussions.
 - Normally focusing on discussions of a specific type of products or topics.
 - Need to perform entity extraction and linking
- Open domain corpora – like Twitter.
 - Can contain documents of **any** entity or topic, with little or no meta-data (like hashtags)
 - Obviously entity extraction and linking is needed. Most challenging.

Search helps

- In a very large corpus, keyword search is often applied first to retrieve relevant posts, using name variations of the desired entities.
- Thus, it's better to have a clear mining objective, from which you'll derive a comprehensive list of name variations to search the large corpus, like Twitter.
- Due to polysemy, a filtering step may be required to get posts that truly contain the desired entities.

Another complexity

- In sentiment analysis applications involving consumer products, the entities are products, which typically have brands and models, forming a hierarchical relationship.
- Brand – product – model
 - e.g. Apple – iPhone – iPhone Pro 14
- Usually by separating brands and models



How: Supervised Entity Linking

- To identify if a mention refers to some known entity, or none.
 - But instead of PER(person), ORG(organization), GPE(geopolitical entity), UKN(unknown), we are typically looking for **products, services, and brands**.
- What's required: the set of desired entities, each with its type and disambiguating text (KB); and entity expressions to be linked
- Example solution, supervised:
 - Candidate generation: for each entity expression, generate possible entities from KB using heuristic rules -> multiple (expression, entity) pairs
 - Candidate ranking: learning to rank the pairs (features such as similarity of entity name strings, similarity of context, entity type, etc.)
 - Or classification: learn a classifier to predict if the (expression, entity) pair is positive/negative
 - Case of NIL (no applicable entity from KB): when no pair is predicted to be positive.
- Main challenges: feature engineering, and labeled examples

How: Semi-Supervised Extraction

- Supervised entity extraction using HMM or CRF is still applicable here, but it requires labeling of training data, which may not always be possible.
- Semi-supervised approaches try to tap on unlabeled examples; e.g. PU learning (learning from positive and unlabeled examples), Bayesian sets
- Given a set of seed entity names, identify all entities of the same type as the seeds from a given corpus
- General idea: if they appear in similar context, they are likely to be the same kind!

PU Learning

- For example, to identify entities of phones
 - Given a set of positive examples (of seed entities), and a corpus
 - Find candidate entities from the corpus
 - Using sequences of specific POS tags: E.g. “[Samsung/NNP Galaxy/NNP S5/CD](#)” as a candidate:
NNP (proper noun), NNPS (plural proper noun), and CD (cardinal number)
 - For each seed entity, create a TF vector for each mention of it representing a positive example, using the surrounding words context of the seed mention.
 - Do the same for each candidate entity
 - Use a PU learning algorithm like S-EM (Liu et al., 2002) to learn a classification model to label the candidates.

Li, X., L. Zhang, B. Liu, and S. Ng. Distributional similarity vs. PU learning for entity set expansion. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010b.

Opinion Holder and Time Extraction

- NER task!
- For social media data
 - the opinion holder is usually the author of the review, blog, post, etc.
 - It's trivial when the ID and date/time of the post are usually known.
- For other data, like news articles, they may need to be extracted out of text, esp. for quoted text.
 - E.g., consider person and organization entities, and score them using Maximum Entropy model.

Aspect Extraction

Main approaches

1. By finding frequent nouns and noun phrases.
2. By exploiting syntactic relations
 - Syntactic dependencies depicting opinion and target relations
 - Lexico-syntactic patterns encoding entity and part/attribute relations
3. Using traditional supervised learning
4. Using Deep Neural Networks (DNN)
5. Zero-shot IE with large language models

Frequency-Based Approach

Good old frequencies...

- Assumption: a reasonable number of reviews about the same product or at least about the same type of products.
 - Find nouns and noun phrases using a POS tagger
 - Count their occurrence frequencies
 - Keep only the frequent ones above a threshold experimentally determined
- **It works.**
 - Aspects are usually expressed as nouns and noun phrases.
 - Vocabulary **converges** when people comment on the same (type of) product.
 - Irrelevant contents tend to be infrequent.

E.g. Popular Mentions from TripAdvisor

Club Med Bintan Island [Visit hotel website](#)

- About
- Location
- Reviews**

Traveller rating	Time of year	Traveller type	Language
<input type="checkbox"/> Excellent 3,976	<input type="checkbox"/> Mar-May	<input type="checkbox"/> Families	<input type="radio"/> All languages (5,640)
<input type="checkbox"/> Very Good 1,147	<input type="checkbox"/> Jun-Aug	<input type="checkbox"/> Couples	<input checked="" type="radio"/> English (4,336)
<input type="checkbox"/> Average 289	<input type="checkbox"/> Sep-Nov	<input type="checkbox"/> Solo	<input type="radio"/> French (472)
<input type="checkbox"/> Poor 122	<input type="checkbox"/> Dec-Feb	<input type="checkbox"/> Business	<input type="radio"/> Chinese (Sim.) (471)
<input type="checkbox"/> Terrible 106		<input type="checkbox"/> Friends	More

Popular mentions

All reviews
circus team
terrace restaurant
chef de village
flying trapeze
evening shows
aerial yoga

family holiday
great holiday
special shout
aqua gym
water aerobics
bintan island
meal times

restaurant manager
adults and kids
free flow
main building
land sports
sports activities
great family

Frequency-based aspect extraction

- Collocation detection



noun

1. LINGUISTICS

the habitual juxtaposition of a particular word with another word or words with a frequency greater than chance.
"the words have a similar range of collocation"

- Frequency is the key here.
- Simple and effective. The candidate aspects are almost always the most important aspects of the product.
- Applicable to entity extraction too.
- Caution: Won't work if the corpus has a mixture of very different products and/or if each product has only one or two reviews.

Improve precision with co-occurrence

- A simplified version of Point-wise Mutual Information (PMI) can be used to compute the co-occurrence strength

$$PMI(a, d) = \frac{hits(a \wedge d)}{hits(a)hits(d)}$$

- Use heuristics to filter for noun phrases that are more likely to be aspects of entities.
 - E.g. for camera reviews, phrases indicating *part-of* relations, like “*of camera*”, “*camera has*”, “*camera comes with*”, can be used to find camera components by web search
 - The discovered phrases that often **co-occur** with such *part-of* (*meronymy*) relation indicators are likely to be correct aspects.

Parts or Attributes

- To distinguish components/parts from attributes
 - Can use Morphological cues (e.g., “-*iness*,” “-*ity*” suffixes)
 - WordNet is a great resource
 - Synonyms (“*fast*” and “*quick*”), antonyms (“*fast*” and “*slow*”)
 - Hyponyms and hypernyms (*is-a* hierarchy), meronyms and holonyms (*part-whole* relation)
 - For example, we can query WordNet with the word “*camera*” and get its senses

Noun

- S: (n) **camera**, **photographic camera** (equipment for taking photographs (usually consisting of a lightproof box with a lens at one end and light-sensitive film at the other))
 - *direct hyponym* / *full hyponym*
 - *part meronym*
 - *direct hypernym* / *inherited hypernym* / *sister term*
- S: (n) **television camera**, **tv camera**, **camera** (television equipment consisting of a lens system that focuses an image on a photosensitive mosaic that is scanned by an electron beam)

<https://wordnet.princeton.edu/>

is-a

- Hyponyms (x is-a camera)

- *direct hyponym / full hyponym*

- [S: \(n\) box camera, box Kodak](#) (a simple camera shaped like a rectangular box)
 - [S: \(n\) candid camera](#) (a miniature camera with a fast lens)
 - [S: \(n\) digital camera](#) (a camera that encodes an image digitally and store it for later reproduction)
 - [S: \(n\) webcam](#) (a digital camera designed to take digital photographs and transmit them over the internet)
 - [S: \(n\) flash camera](#) (a camera with a photoflash attachment)
 - [S: \(n\) motion-picture camera, movie camera, cine-camera](#) (a camera that takes a sequence of photographs that can give the illusion of motion when viewed in rapid succession)
 - [S: \(n\) sound camera](#) (a movie camera that records sounds in synchrony with the visual images)
 - [S: \(n\) point-and-shoot camera](#) (a lightweight photographic camera with an autofocus)
 - [S: \(n\) Polaroid camera, Polaroid Land camera](#) (a camera that develops and produces a positive print within seconds)
 - [S: \(n\) portrait camera](#) (a camera with a portrait lens)
 - [S: \(n\) reflex camera](#) (camera that allows the photographer to view and focus the exact scene being photographed)

is-a relation

- Hypernyms (camera is-a x)

- [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - [S: \(n\) photographic equipment](#) (equipment used by a photographer)
 - [S: \(n\) equipment](#) (an instrumentality needed for an undertaking or to perform a service)
 - [S: \(n\) instrumentality, instrumentation](#) (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
 - [S: \(n\) artifact, artefact](#) (a man-made object taken as a whole)
 - [S: \(n\) whole, unit](#) (an assemblage of parts that is regarded as a single entity) "how big is that part compared to the whole?"; "the team is a unit"
 - [S: \(n\) object, physical object](#) (a tangible and visible entity; an entity that can cast a shadow) "it was full of rackets, balls and other objects"
 - [S: \(n\) physical entity](#) (an entity that has physical existence)
 - [S: \(n\) entity](#) (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - [S: \(n\) photographic equipment](#) (equipment used by a photographer)
 - [S: \(n\) camera, photographic camera](#) (equipment for taking photographs (usually consisting of a lightproof box with a lens at one end and light-sensitive film at the other))
 - [S: \(n\) clapperboard](#) (photographic equipment used to synchronize sound and motion picture; boards held in front of a movie camera are banged together)
 - [S: \(n\) developer](#) (photographic equipment consisting of a chemical solution for developing film)
 - [S: \(n\) enlarger](#) (photographic equipment consisting of an optical projector used to enlarge a photograph)
 - [S: \(n\) flash, photoflash, flash lamp, flashgun, flashbulb, flash bulb](#) (a lamp for providing momentary light to take a photograph)
 - [S: \(n\) light meter, exposure meter, photometer](#) (photographic equipment that measures the intensity of light)
 - [S: \(n\) photographic paper, photographic material](#) (light-sensitive paper on which photograph can be printed)
 - [S: \(n\) sensitometer](#) (a measuring instrument for measuring the light sensitivity of film over a range of exposures)

part-of relation

- Meronyms (x is part-of camera)

- part meronym

- S: (n) aperture (a device that controls amount of light admitted)
 - S: (n) camera lens, optical lens (a lens that focuses the image in a camera)
 - S: (n) delayed action (a mechanism that automatically delays the release of a camera shutter for a fixed period of time so that the photographer can appear in the picture)
 - S: (n) diaphragm, stop (a mechanical device in a camera that controls size of aperture of the lens) "*the new cameras adjust the diaphragm automatically*"
 - S: (n) finder, viewfinder, view finder (optical device that helps a user to find the target of interest)
 - S: (n) hood, lens hood (a tubular attachment used to keep stray light out of the lens of a camera)
 - S: (n) magazine, cartridge (a light-tight supply chamber holding the film and supplying it for exposure as required)
 - S: (n) shutter (a mechanical device on a camera that opens and closes to control the time of a photographic exposure)
 - S: (n) sprocket (roller that has teeth on the rims to pull film or paper through)

Further refinements

- Filter by dropping aspects that do not have sufficient mentions alongside known sentiment words (Blair-Goldensohn et al., 2008)
- Collapse aspects at the word stem level
- Pattern-based filter to remove non-aspect expressions (Moghaddam and Ester, 2010)
- Compare the frequencies of the candidates in a review corpus with those in a generic corpus to identify true aspects (Scuffidi et al., 2007)
- First find the core aspect words, then use the information distance to find other related words, e.g. “\$” and “*dollars*” for “*price*” (Long et al., 2007)
- Etc., etc.

Lexico-Syntactic Approaches

Exploiting syntactic relations

- Syntactic relations between sentiment expressions and their sentiment or opinion targets (Needs a parser)
 - E.g. “*This camera takes great photos.*”
 - “*Picture quality and battery life are great.*”
- Linguistic patterns for aspect-entity relations
 - E.g. X of Y: “the *voice quality* of the *iPhone*”
 - Genitives: “the *camera's price*”

Syntactic opinion-target relations

- Can use a dependency parser
 - Identify reliable dependency relation templates from training(labelled) data
 - Then use them to find valid aspect-sentiment pairs in test data
 - Can be used for simultaneous extraction of both sentiment words and opinion targets – Double Propagation (DP) method (Qiu et al. 2009, 2011)
 - E.g. “*The software is amazing.*” -> pattern “NN – nsubj –JJ”

Tagging

```
The/DT software/NN is/VBZ amazing/JJ ./.
```

Parse

```
(ROOT
  (S
    (NP (DT The) (NN software))
    (VP (VBZ is)
      (ADJP (JJ amazing)))
    (. .)))
```



Universal dependencies, enhanced

```
det(software-2, The-1)
nsubj(amazing-4, software-2)
cop(amazing-4, is-3)
root(ROOT-0, amazing-4)
```

Stanford Parser

<http://nlp.stanford.edu:8080/parser/index.jsp>

Dependencies

- Dependencies are binary relations: a grammatical relation between
 - A *governor* (also known as *head*), and
 - A *dependent*
- For example, *amod* (adjectival modifier)
 - An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP.
 - E.g. *I like this amazing software.*

`nsubj(like, I)` - "I" is the nominal subject of "like".

`dobj(like, software)` - "software" is the direct object of "like".

`det(software, this)` - "this" is the determiner of "software".

`amod(software, amazing)` - "amazing" is the adjectival modifier of "software".

Universal Dependencies

	Nominals	Clauses	Modifier words	Function Words
Core arguments	<u>nsubj</u> <u>obj</u> <u>iobj</u>	<u>csubj</u> <u>ccomp</u> <u>xcomp</u>		
Non-core dependents	<u>obl</u> <u>vocative</u> <u>expl</u> <u>dislocated</u>	<u>advcl</u>	<u>advmmod</u> * <u>discourse</u>	<u>aux</u> <u>cop</u> <u>mark</u>
Nominal dependents	<u>nmod</u> <u>appos</u> <u>nummod</u>	<u>acl</u>	<u>amod</u>	<u>det</u> <u>clf</u> <u>case</u>
Coordination	MWE	Loose	Special	Other
<u>conj</u> <u>cc</u>	<u>fixed</u> <u>flat</u> <u>compound</u>	<u>list</u> <u>parataxis</u>	<u>orphan</u> <u>goeswith</u> <u>reparandum</u>	<u>punct</u> <u>root</u> <u>dep</u>

Stanford Typed Dependencies

- *mod* – modifier

amod - adjectival modifier

appos - appositional modifier

advcl - adverbial clause modifier

det - determiner

predet - predeterminer

preconj - preconjunct

vmod - reduced, non-finite verbal modifier

mwe - multi-word expression modifier

mark - marker (word introducing an *advcl* or *ccomp*)

advmod - adverbial modifier

neg - negation modifier

rcmod - relative clause modifier

quantmod - quantifier modifier

nn - noun compound modifier

npadvmod - noun phrase adverbial modifier

tmod - temporal modifier

num - numeric modifier

number - element of compound number

prep - prepositional modifier

poss - possession modifier

possessive - possessive modifier ('s)

prt - phrasal verb particle

Stanford Typed Dependencies

- *arg* - arguments

agent - agent

comp - complement

acomp - adjectival complement

ccomp - clausal complement with internal subject

xcomp - clausal complement with external subject

obj - object

dobj - direct object

iobj - indirect object

pobj - object of preposition

subj - subject

nsubj - nominal subject

nsubjpass - passive nominal subject

csubj - clausal subject

csubjpass - passive clausal subject

Common dependency relations

- Between sentiment words and aspects:
amod, prep, nsubj, csubj, xsubj, dobj and *iobj*
- Between sentiment words or aspects themselves: the conjunction relation *conj*

“*The phone has a nice screen.*”

Tagging

The/DT phone/NN has/VBZ a/DT nice/JJ screen/NN ./.

Universal dependencies, enhanced

```
det(phone-2, The-1)
nsubj(has-3, phone-2)
root(ROOT-0, has-3)
det(screen-6, a-4)
amod(screen-6, nice-5)
dobj(has-3, screen-6)
```

“*I like the color of the phone.*”

Tagging

I/PRP like/VBP the/DT color/NN of/IN the/DT phone/NN ./.

Universal dependencies, enhanced

```
nsubj(like-2, I-1)
root(ROOT-0, like-2)
det(color-4, the-3)
dobj(like-2, color-4)
case(phone-7, of-5)
det(phone-7, the-6)
nmod:of(color-4, phone-7)
```

Example Rules for Aspect and Opinion Word Extraction

Rule ID	Observed Relation (Line 1) and Constraints (Lines 2–4)	Output	Examples
R1 ₁ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow A$ s.t. $O \in \{O\}$, $O\text{-}Dep \in \{MR\}$, $POS(A) \in \{NN\}$	$a = A$	The phone has a <u>good</u> “screen.” <i>good</i> → <i>mod</i> → <i>screen</i>
R1 ₂ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow A\text{-}Dep \leftarrow A$ s.t. $O \in \{O\}$, $O/A\text{-}Dep \in \{MR\}$, $POS(A) \in \{NN\}$	$a = A$	“iPod” is the <u>best</u> MP3 player. <i>best</i> → <i>mod</i> → <i>player</i> ← <i>subj</i> ← <i>iPod</i>
R2 ₁ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow A$ s.t. $A \in \{A\}$, $O\text{-}Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	Same as R1 ₁ with <i>screen</i> as the known word and <i>good</i> as the extracted word
R2 ₂ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow A\text{-}Dep \leftarrow A$ s.t. $A \in \{A\}$, $O/A\text{-}Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	Same as R1 ₂ with <i>iPod</i> as the known word and <i>best</i> as the extract word
R3 ₁ (AA-Rel)	$A_{i(j)} \rightarrow A_{i(j)}\text{-}Dep \rightarrow A_{j(i)}$ s.t. $A_{j(i)} \in \{A\}$, $A_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(A_{i(j)}) \in \{NN\}$	$a = A_{i(j)}$	Does the player play DVDs with <u>audio</u> and “video”? <i>video</i> → <i>conj</i> → <i>audio</i>
R3 ₂ (AA-Rel)	$A_i \rightarrow A_i\text{-}Dep \rightarrow H \leftarrow A_j\text{-}Dep \leftarrow A_j$ s.t. $A_i \in \{A\}$, $A_i\text{-}Dep = A_j\text{-}Dep$ OR ($A_i\text{-}Dep = \text{subj}$ AND $A_j\text{-}Dep = \text{obj}$), $POS(A_j) \in \{NN\}$	$a = A_j$	Canon “G3” has a great <u>lens</u> . <i>len</i> → <i>obj</i> → <i>has</i> ← <i>subj</i> ← <i>G3</i>
R4 ₁ (OO-Rel)	$O_{i(j)} \rightarrow O_{i(j)}\text{-}Dep \rightarrow O_{j(i)}$ s.t. $O_{j(i)} \in \{O\}$, $O_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	The camera is <u>amazing</u> and “easy” to use. <i>easy</i> → <i>conj</i> → <i>amazing</i>
R4 ₂ (OO-Rel)	$O_i \rightarrow O_i\text{-}Dep \rightarrow H \leftarrow O_j\text{-}Dep \leftarrow O_j$ s.t. $O_i \in \{O\}$, $O_i\text{-}Dep = O_j\text{-}Dep$ OR ($O_i / O_j\text{-}Dep \in \{\text{pnmod, mod}\}$), $POS(O_j) \in \{JJ\}$	$o = O_j$	If you want to buy a <u>sexy</u> , “cool,” accessory-available MP3 player, you can choose iPod. <i>sexy</i> → <i>mod</i> → <i>player</i> ← <i>mod</i> ← <i>cool</i>

Lexico-syntactic patterns

- Genitive constructions are frequently used to express part-of and attribute-of relations.
 - “*the battery of the iPhone*” – “NP-head of NP-mod”
 - “*the iPhone's sound quality*” – “NP-mod's NP-head”
- However, the semantic relations of the two nouns can be quite different in different context
 - **Part-of:** “*iPhone's battery*”
 - Possession: “*John's iPhone*”
 - **Attribute-of:** “*iPhone's price*”
 - Kinship: “*John's brother*”
 - Source-from: “*John's birth city*”
 - Make-produce: “*Apple's phone*”
- In sentiment analysis, it's easier to **fix NP-mod to be a specific entity to look for its aspect**

The approach with no parser

- Full parsing is expensive (typically fewer than 20 sentences per second), and it doesn't work well on informal data.
- **Approximate** the dependency using distance:
 - If a sentence does not have a frequent aspect but has some sentiment words, the **nearest** noun or noun phrase to a sentiment word is extracted as an aspect.
- **Approximate** the dependency relations
 - Using linear patterns of words and POS tags, or chunk patterns (from shallow parsing)
 - And extract using a good pattern matching algorithm.
- Very useful in practice.

Ranking the candidates

- Aspect relevance ($r(a)$): how likely it's a genuine aspect
 - If an aspect is modified by multiple sentiment words
“delivery” modified by *“quick”*, *“cumbersome”*, *“timely”*
 - If it's extracted by multiple lexico-syntactic patterns
 - *“The engine of the car is large”* and *“The car has a big engine”*
 - If it's extracted by both a sentiment word modification relation and a lexico-syntactic pattern in the same sentence
 - *“There is a bad hole in the mattress”*
- Aspect frequency ($f(a)$)
 - Rank the frequent aspects higher
- Final ranking score: $S(a) = r(a) \log (f(a))$

Going further

- Use phrase dependency parser to extract noun phrases and verb phrases.
- Adding comparative- and superlative-based relations
 - E.g. “*The iPhone 5 has better voice quality than Moto X.*”
- Adding sentiment composition rules
 - E.g. “*Enbrel has reduced my joint pain.*”
- Resource usage aspect - resource expression, (*usage_verb, quantifier, resource_noun*)
 - E.g. “*This washer uses a lot of water.*”
usage

Implicit aspect

- Explicit aspects
 - “*The picture quality of the camera is great.*”
- Implicit aspects - opinion expressions that do not mention aspect explicitly as a noun
 - “*The camera is expensive*” (“price”)
 - “*This beautiful camera ...*” (“appearance”)
 - “*The camera does not easily fit in a pocket*” (“size”)
- Mapping adjectives or verb phrases to aspects
 - Corpus-based approach: using the co-occurrence of sentiment words and explicit mentions e.g. “*The size of the phone is small*” vs “*the phone is small*”
 - Dictionary-based approach: using dictionary definition of “*expensive*” – “*marked by high prices*”

Grouping aspect expressions into categories

- People use different words or phrases to describe the same aspect or aspect category.
e.g. “*sound quality*”, “*voice quality*”
- Need to group **aspect expressions** into **aspect categories**, each representing one aspect
- Very challenging as it's **subjective** task. Different application or different users may require different categories based on application need or granularity of analysis
- Can use WordNet or other thesaurus to find synonyms, but not sufficient
 - Domain dependent synonyms
 - Multiword phrases, not in WordNet or dictionaries
 - Not synonyms – “expensive” and “cheap”

Some methods

- Aspects sharing common words: “*battery life*” and “*battery power*”
- Aspects that are synonyms in dictionaries: “*movie*” and “*film*”
- Aspects with short lexical distances measured in WordNet: “*movie*” and “*show*”
- Mapping aspect expressions to aspect nodes in an **existing taxonomy**, based on similarity
- Using topic modelling (LDA)
- Other semi-supervised methods

The background features a large, abstract graphic composed of several overlapping, wavy bands of varying shades of blue. The bands curve from the top left towards the bottom right, creating a sense of motion. The darkest band is positioned in the center-left, while lighter bands are on the right and bottom. A thin white vertical line runs through the center of the slide.

Traditional Supervised- Learning Approaches

3. Supervised learning

- Treat aspect extraction as a special IE problem.
- Dominant method: sequential learning or **sequence labeling** like *hidden Markov models* (HMMs) and *conditional random fields* (CRF).
- Many other methods, like sequential rules, tree-structured classification, etc. have been tried as well.
- Requires **labelled data**.

Sequence labelling - example

- “*I love the ease of transferring the pictures to my computer.*”
- Tags: IOB/IO tags, or similar tags

```

<BG>I</BG><OPINION_POS_EXP>love</OPINION_P
OS_EXP><BG>the</BG><PROD_FEAT-
BOE>ease</PROD_FEAT-BOE> <PROD_FEAT-MOE>
of</PROD_FEAT-MOE><PROD_FEAT-
MOE>transferring</PROD_FEAT-MOE>
<PROD_FEAT-MOE>the</PROD_FEAT-MOE>
<PROD_FEAT-EOE>pictures</PROD_FEAT-EOE>
<BG>to</BG><BG>my</BG><BG>computer</BG>
  
```

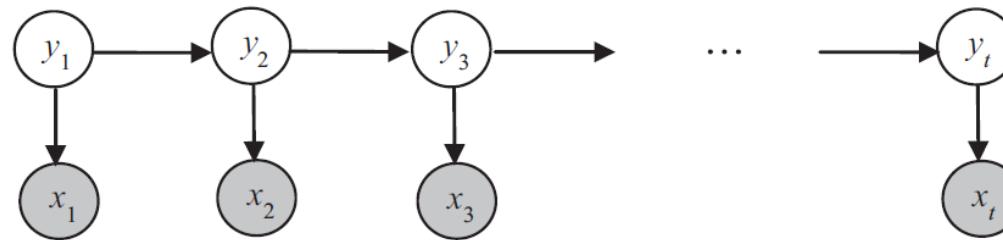
Token	Label
I	O
love	B-POS
the	O
ease	B-ASP
of	I-ASP
transferring	I-ASP
the	I-ASP
pictures	I-ASP
to	O
my	O
computer	O
.	O

HMM

- Directed sequence model, successfully applied to many sequence labeling problems such as NER and POS tagging

$\mathbf{y} = \langle y_0, y_1, \dots, y_t \rangle$: hidden state sequence

$\mathbf{x} = \langle x_0, x_1, \dots, x_t \rangle$: observation sequence



- Assumptions:
 - state y_i only depends on its immediate predecessor state y_{i-1} (Markov Assumption)
 - the observation x_i only depends on the current state y_i .
- For aspect extraction,
 - Observations: words or phrases in a review
 - Underlying/hidden states: aspect or opinion expression tags

HMM

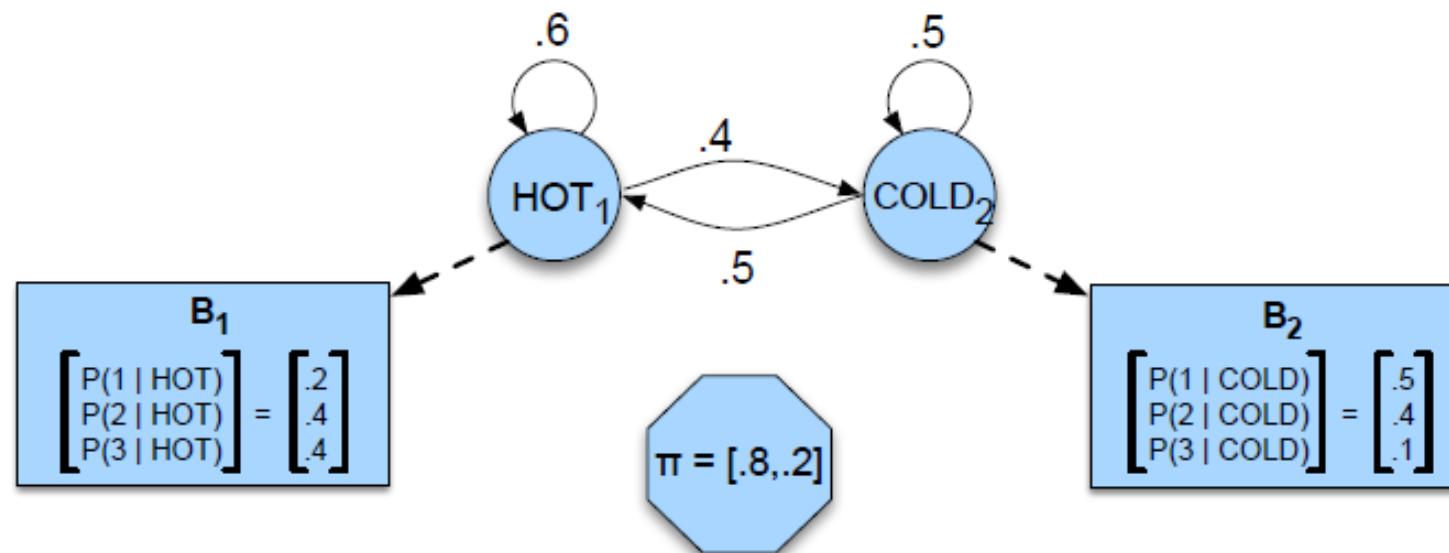
- The **joint probability** of a state sequence \mathbf{y} and an observation sequence \mathbf{x}

$$p(\mathbf{y}, \mathbf{x}) = \prod_{i=1}^t p(y_i|y_{i-1})p(x_i|y_i)$$

- $p(y_0)$ over the initial state,
 - a state transition distribution $p(y_i|y_{i-1})$,
 - an observation distribution $p(x_i|y_i)$
- Learning the model: given some observation sequences, learn the model parameter that maximizes the observation probability, e.g. MLE from counts.
- Applying the model (decoding): given a new observation sequence, find an optimal state sequence for it, e.g. Viterbi algorithm

Understanding HMM

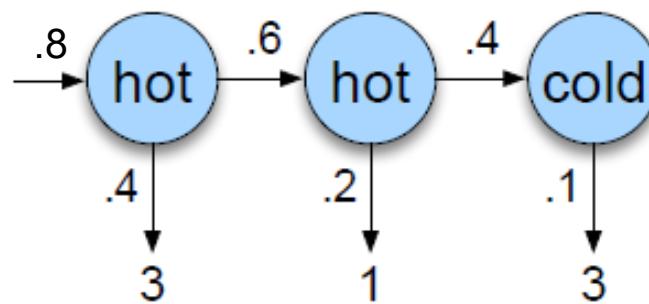
- Hidden states: H(ot) and C(old) weather
- Observations: the number of ice creams eaten on that day – {1, 2, 3}
- Transition probabilities
- Emission probabilities (observation likelihood)



Sequence probability

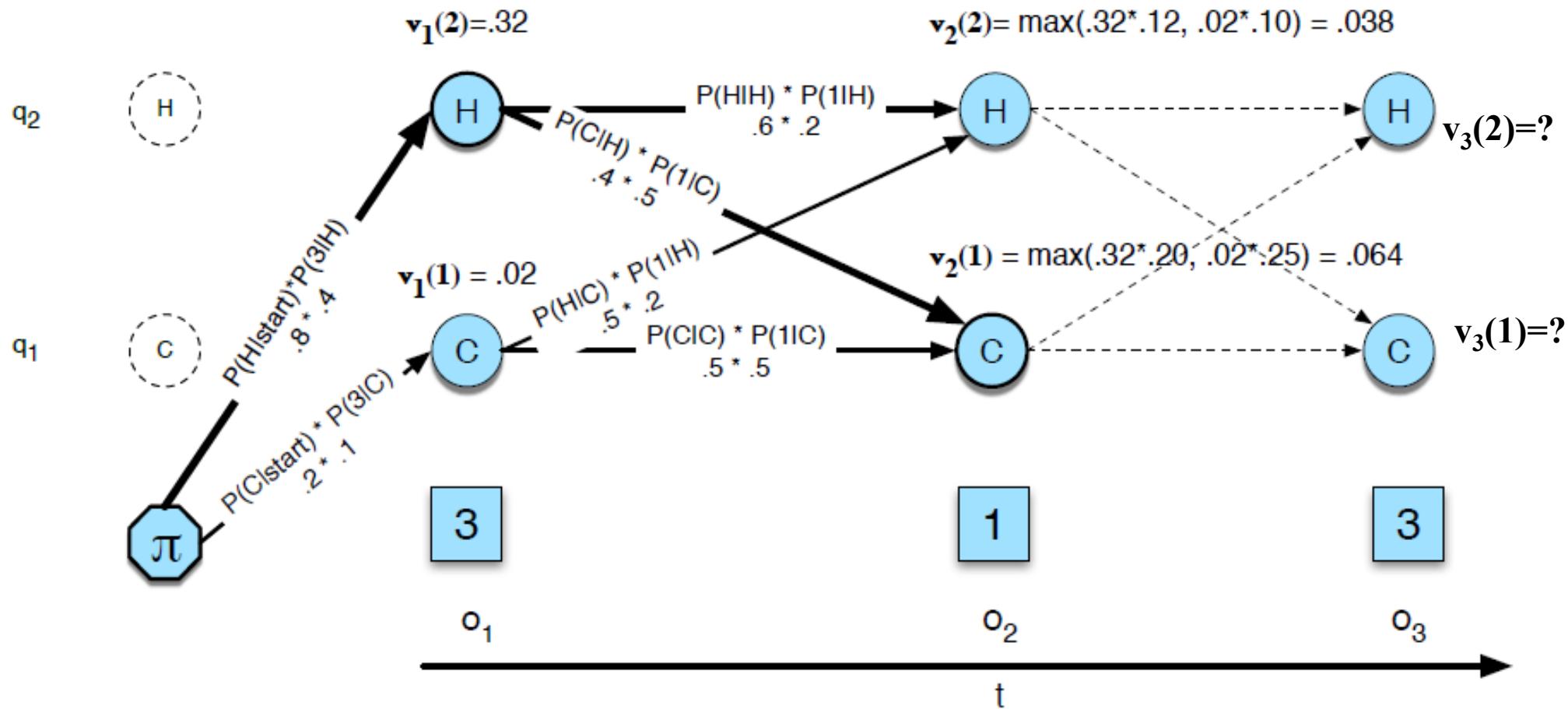
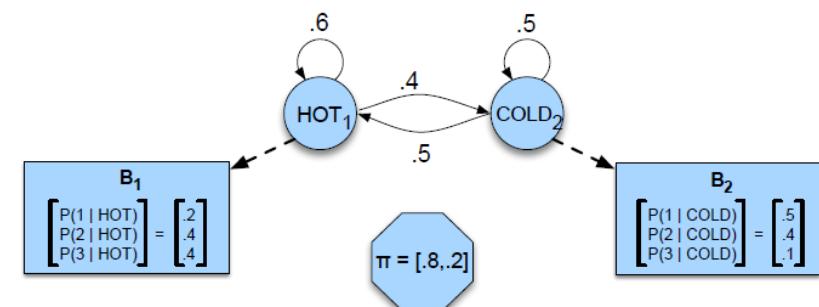
- E.g. to compute the joint probability of observation sequence $\langle 3, 1, 3 \rangle$ and hidden state sequence $\langle \text{hot}, \text{hot}, \text{cold} \rangle$

$$\begin{aligned}
 P(3\ 1\ 3, \text{hot hot cold}) &= P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \\
 &\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold})
 \end{aligned}$$



Viterbi Decoding

Given the observation sequence $<3, 1, 3>$, find the best sequence of hidden states



Exercise

- The previous slide shows the computation of $v_t(j)$ for two states at two time steps: $v_1(1)$ & $v_1(2)$, $v_2(1)$ & $v_2(2)$

$$v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t)$$

- Now try to compute the values for $v_3(1)$ & $v_3(2)$

HMM for text sequence tagging

- Lexicalized HMM, to integrate features such as POS information into HMM
- E.g. Given a sequence of words $W = w_1 w_2 w_3 \dots w_n$ and corresponding parts-of-speech $S = s_1 s_2 s_3 \dots s_n$
- The task is to find an appropriate sequence of tags $T = t_1 t_2 t_3 \dots t_n$ that maximize the conditional probability $P(T|W,S)$

$$\hat{T} = \arg \max_T \prod_{i=1}^n \begin{pmatrix} P(s_i | w_{i-1}, t_i) \times \\ P(w_i | w_{i-1}, s_i, t_i) \times \\ P(t_i | w_{i-1}, t_{i-1}) \end{pmatrix}$$

Conditional Random Fields

- CRF is a probabilistic (**undirected**) graphical model, also good for sequence modeling
- Discriminative model considering dependencies between neighboring nodes, considering global context, **easier to incorporate a lot of features**, but slower in training
- It models a conditional probability $p(\mathbf{Y}|\mathbf{X})$ over hidden sequence \mathbf{Y} given observation sequence \mathbf{X}

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \underbrace{\theta_k}_{\text{Weight}} \underbrace{f_k(y_t, y_{t-1}, \mathbf{x}_t)}_{\text{Feature}} \right\}$$

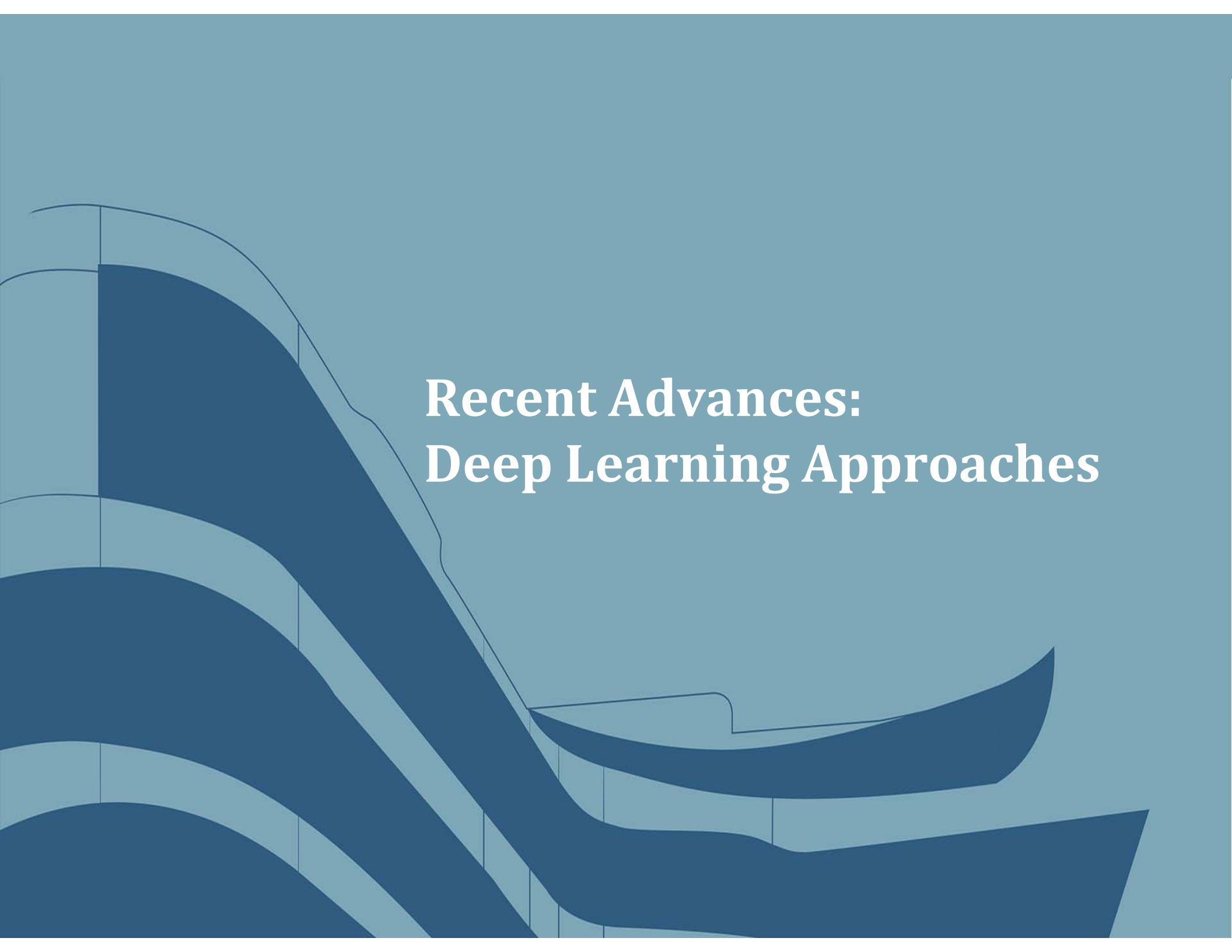
Normalization

- To label an unknown observation sequence \mathbf{x} by selecting the hidden sequence \mathbf{y} that maximizes $p(\mathbf{y}|\mathbf{x})$.

CRF in NLP context

	the	only	redeeming	factor	was	the	food
X	DT	JJ	NN	NN	VBD	DT	NN
Y	O	O	O	O	O	O	B-A

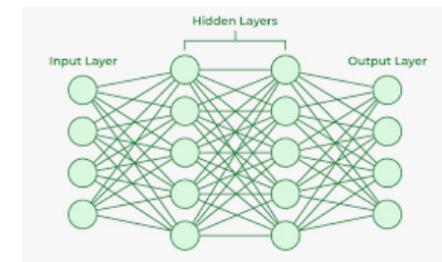
- Output Y can be the EA tags (that you want to learn):
 - B-A: beginning of aspect
 - I-A: part of aspect
 - O: anything outside
- The EA tag of a word can be determined by X features, which can be the word, its POS tag, its neighbouring words (not just preceding words) with their respective POS and EA tags, etc.
- Feature engineering required.

The background of the slide features a large, abstract graphic composed of several overlapping, wavy bands of varying shades of blue. The bands curve from the top left towards the bottom right, creating a sense of motion and depth. The colors range from light cyan at the top to dark navy blue at the bottom.

Recent Advances: Deep Learning Approaches

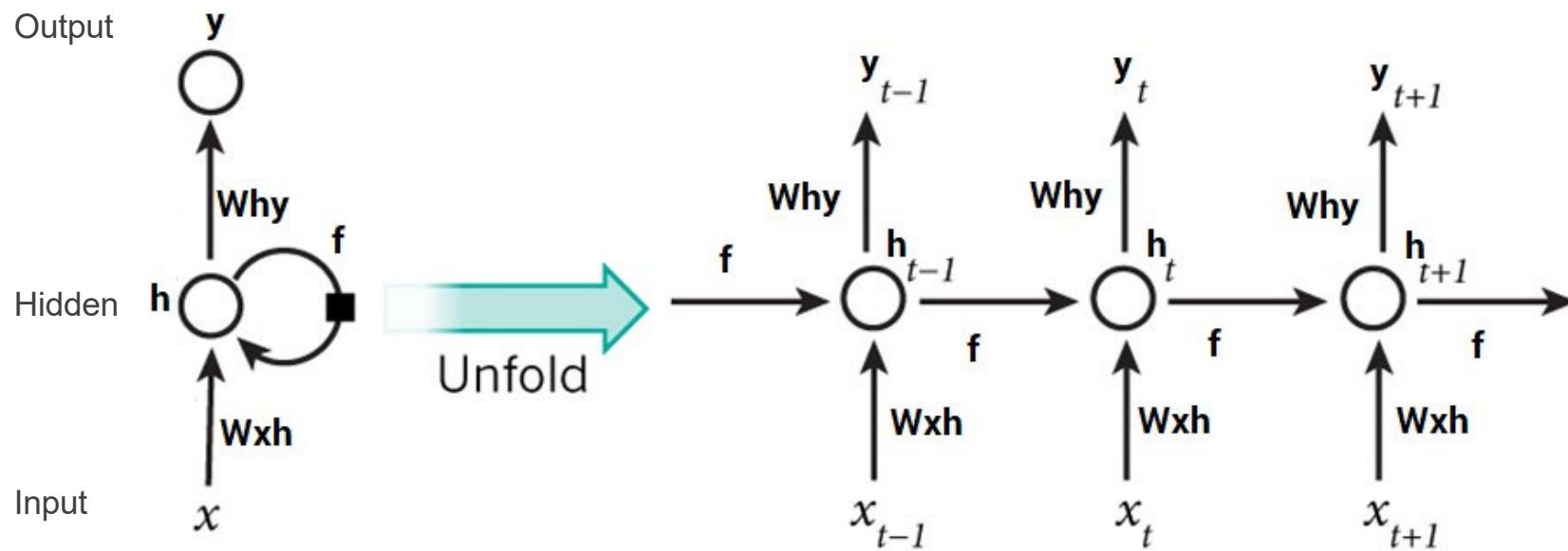
Deep Neural Networks

- Inspired by the structure and functioning of the human brain, DNNs have emerged as powerful tools for sequence modelling in NLP
- Common networks for sequence modelling: Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) Networks, Transformer Networks
- Benefits:
 - Outperforms traditional machine learning models
 - Can learn feature representation and extraction tasks simultaneously (no more feature engineering)
- Issues:
 - Requires a substantial amount of labelled data for training
 - Interpretability issue (black box models)



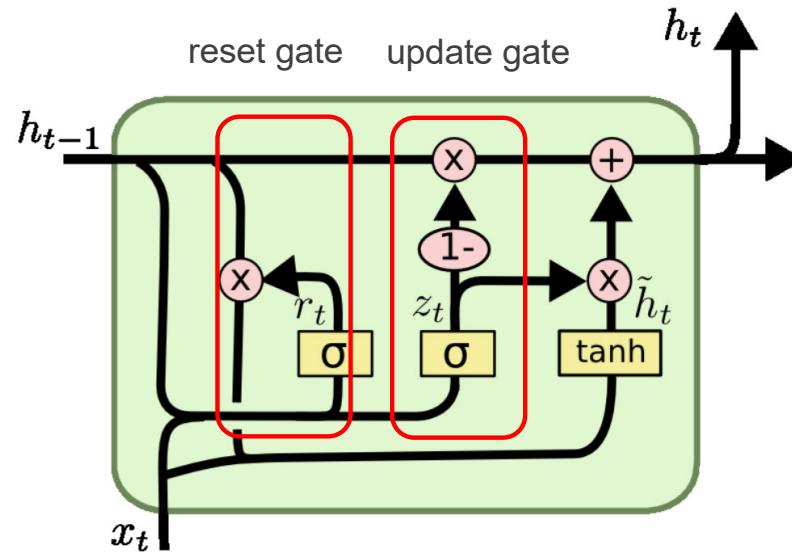
Recurrent Neural Network

- Learn through sequential dependence, getting hidden state from the previous step
- Cons: vanishing gradient problem (or short-term memory), can't handle long-term dependencies often required in NLP



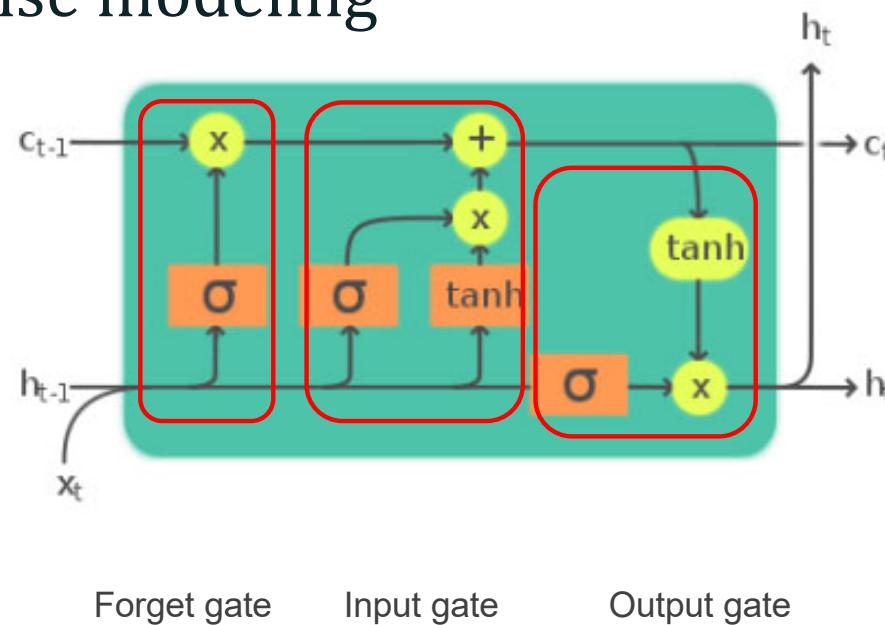
Gated Recurrent Units

- A variant of RNN
- Using gating mechanisms to control the flow of information in the network – update gate and reset gate
- Capturing long-term dependencies by allowing relevant information to pass through hidden states
- Fewer parameters than LSTM, therefore, faster to train



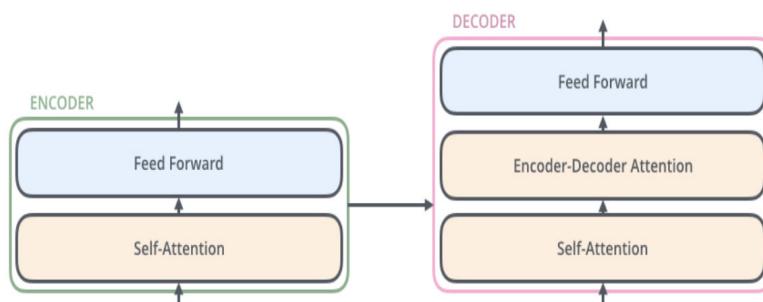
Long-Short Term Memory (LSTM)

- A popular variant of RNN
- Tackle long-term dependency issue by using memory cell and gates (input gate, forget gate, output gate) to selectively update and forget information (cell state and hidden state)
- More precise modeling

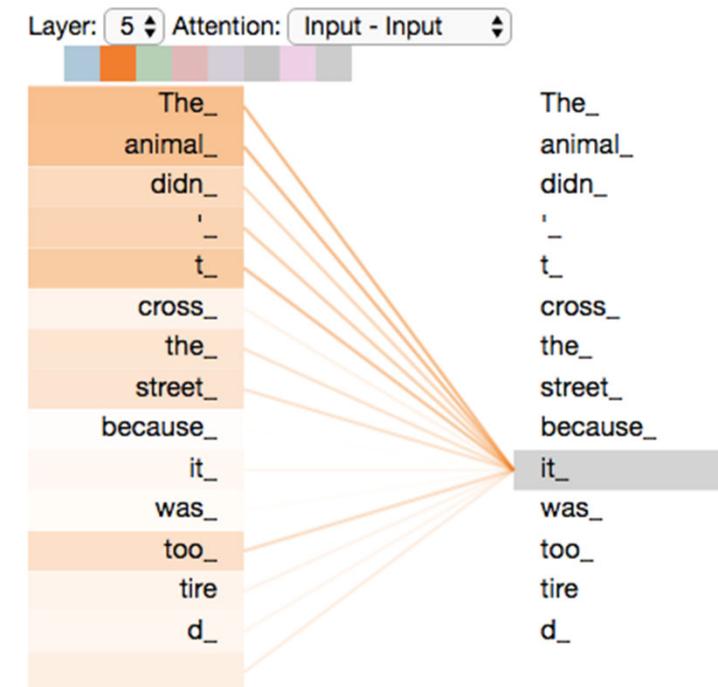


Transformers with Attention

- A novel architecture of DNN with layers of encoders/decoders
- A special **attention layer** allowing each element in the sequence to attend to other elements
- Better handling of long-term dependencies



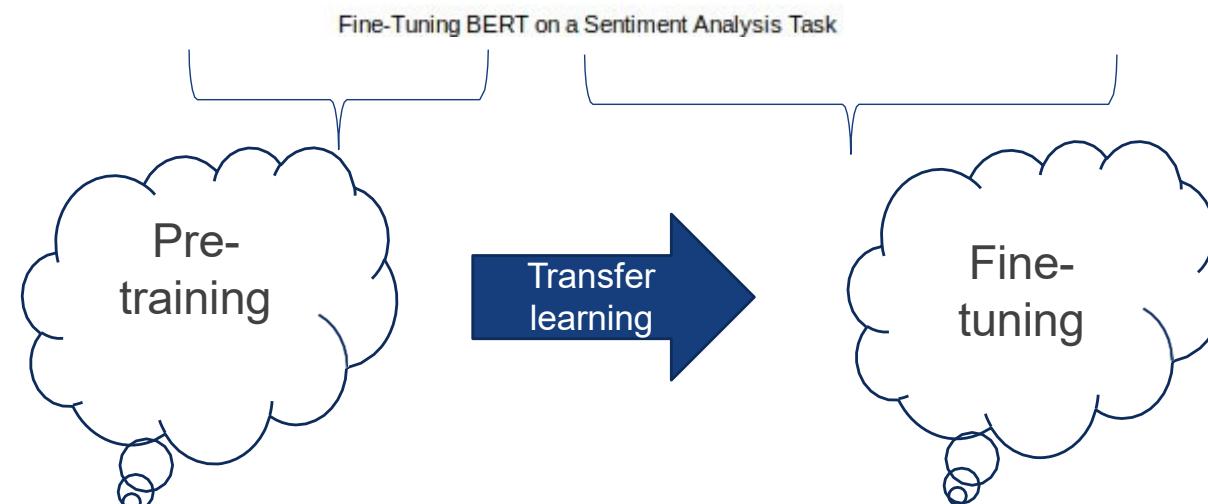
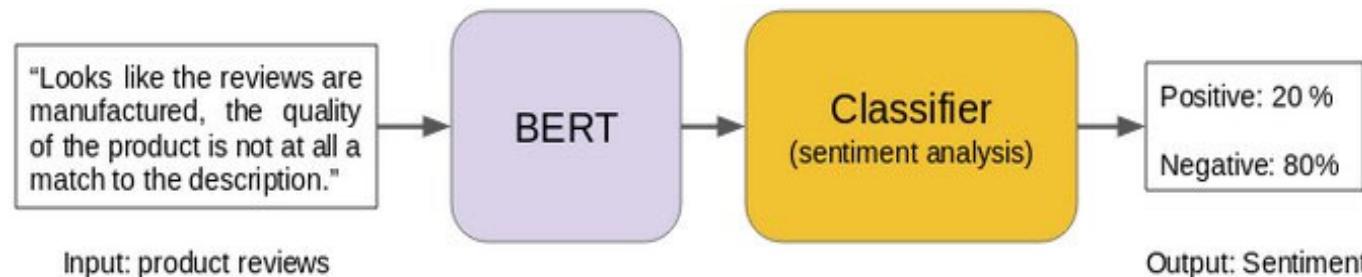
<https://jalammar.github.io/illustrated-transformer/>



Huge advances from 2018

- Elmo (Mar 2018) – contextualized word embeddings
- ***Pre-trained*** models
 - Pre-trained transformers using huge amount of text data
 - The pre-training helps the model to capture context, syntactic and semantic information in the text.
 - The model can be further fine-tuned using small data set to perform a wide variety of NLP tasks, including sentiment mining tasks
 - BERT, GPT/GPT2/GPT3, XLNet, T5, BART, etc.

Fine-tuning LMs for sentiment analysis



By Google –

- BERT base:
No of transformer layers = 12;
total parameters = 110M
- BERT Large:
No. of transformer layers = 24
total parameters = 340M

Sentiment analysis, text classification (eg. spam filters, toxic comments, Q&A)

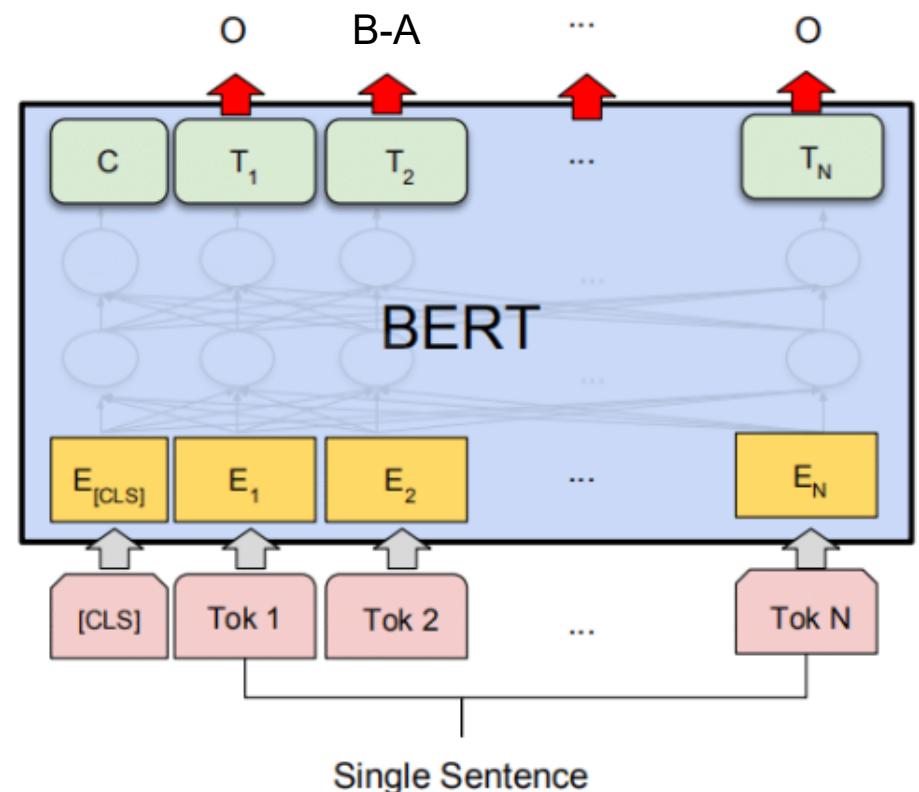
Sentence-level Sentiment Classification

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI
1	AliceMind & DIRL	StructBERT + CLEVER		91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2
2	ERNIE Team - Baidu	ERNIE		90.9	74.4	97.8	93.9/91.8	93.0/92.6	75.2/90.9	91.9	91.4	97.3	92.0	95.9
3	DeBERTa Team - Microsoft	DeBERTa / TuringNLVRv4		90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5
4	HFL iFLYTEK	MacALBERT + DKM		90.7	74.3	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5
+ 5	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5
6	liangzhu ge	Deberta + adv (ensemble)		90.4	72.7	97.3	92.7/90.3	93.2/92.9	75.6/90.8	91.7	91.5	96.4	92.5	95.2
7	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5
8	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5
+ 9	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5
+ 10	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5

<https://gluebenchmark.com/leaderboard>

ABSA with DL

- Most treat extraction as a **sequence tagging** task, using LSTM-based models for **token classification**
- Or by fine-tuning pre-trained transformers
 - Leverage on the knowledge in the pretrained models
 - The model learns to adapt to a specific domain and task
 - Need a relatively small set of labeled data of the relevant domain and task



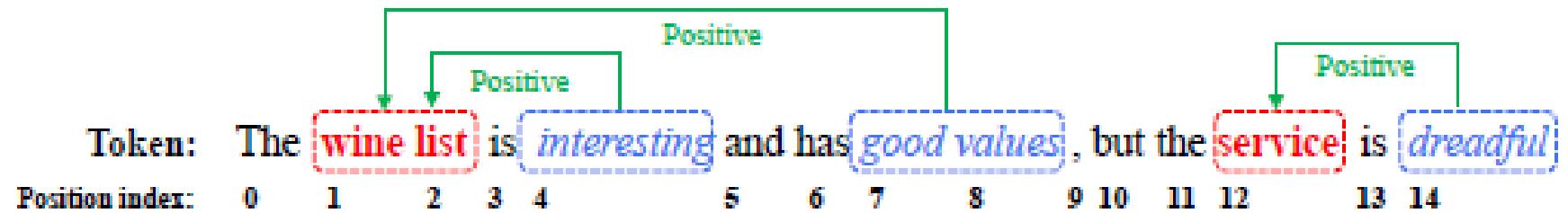
Auto-encoding models like BERT learn to detect boundaries of entities in token classification.

A sequence-to-sequence approach

- As sequence generation, an example solution using BART
 - a strong sequence-to-sequence pre-trained model for natural language generation.
 - With both encoder and decoder layers
 - Pre-training task: take a masked or permuted sentence as input, return the restored sentence.
 - Given input text, generate a sequence of token indexes and class indexes for various ABSA tasks

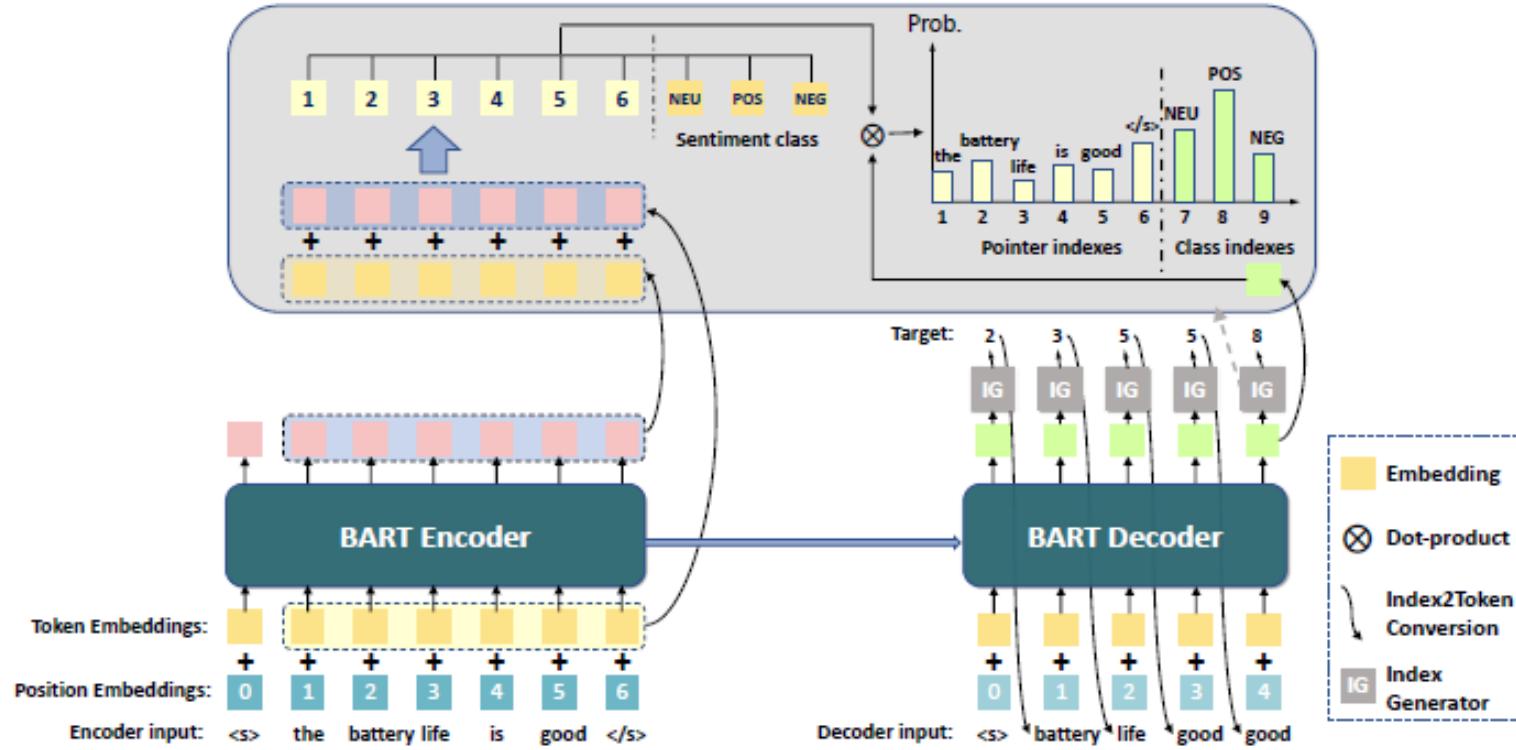
Yan, Hang, et al. "A Unified Generative Framework for Aspect-Based Sentiment Analysis." *arXiv preprint arXiv:2106.04300* (2021).

ABSA subtask formulation



Subtask	Target Sequence
<i>AE</i>	1, 2, 12, 12, </s>
<i>OE</i>	4, 4, 7, 8, 14, 14, </s>
<i>ALSC</i>	<u>1</u> , <u>2</u> , POS, </s> <u>12</u> , <u>12</u> , POS, </s>
<i>AOE</i>	<u>1</u> , <u>2</u> , 4, 4, 7, 8, </s> <u>12</u> , <u>12</u> , 14, 14, </s>
<i>AESC</i>	1, 2, POS, 12, 12, NEG, </s>
<i>Pair</i>	1, 2, 4, 4, 1, 2, 7, 8, 12, 12, 14, 14, </s>
<i>Triplet</i>	1, 2, 4, 4, POS, 1, 2, 7, 8, POS, 12, 12, 14, 14, POS, </s>

Overall architecture



Source sequence: "<s> the battery life is good </s>"

Target sequence: "2 3 5 5 8 6" (<s> battery life good good POS </s>)

A large, abstract graphic on the left side of the slide features several thick, wavy lines in varying shades of blue. These lines form a dynamic, flowing pattern that spans most of the vertical space. The lines are primarily horizontal or slightly curved, creating a sense of motion and depth.

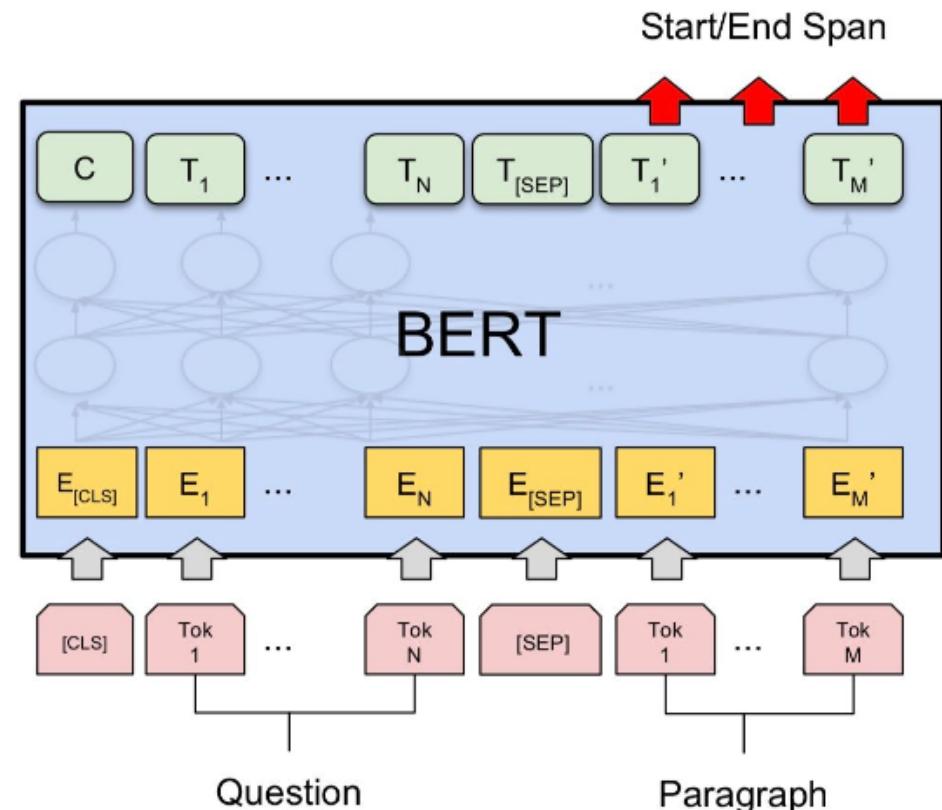
The latest advance: Zero-Shot Approach

Zero-Shot ABSA with LLMs

- Leveraging on powerful NLU capabilities of large pre-trained language models
- Zero-shot: **no training/fine-tuning of the model**, no examples provided to the model
- Various approaches
 - Extraction by question answering (Extractive QA)
 - By natural language inference (NLI)
 - **Prompt-based** methods using instruction-tuned language models like ChatGPT

Extraction as question answering

- **Extractive QA** – Given a question and a context, predict the span of answer in the context paragraph
- Can be used for entity/aspect expression extraction
- Challenge:
 - Only works for explicitly mentioned expressions
 - Need to ask the right question



Extractive QA

- Generic questions do not work well

```
example = "Must try: xiao long bao, chili crab bun, fried rice with pork, braised beef noodles."
result = qa(question="What is this comment about?", context=example)
print(f"Answer: '{result['answer']}', score: {round(result['score'], 4)}")
```

Answer: 'Must try: xiao long bao', score: 0.041

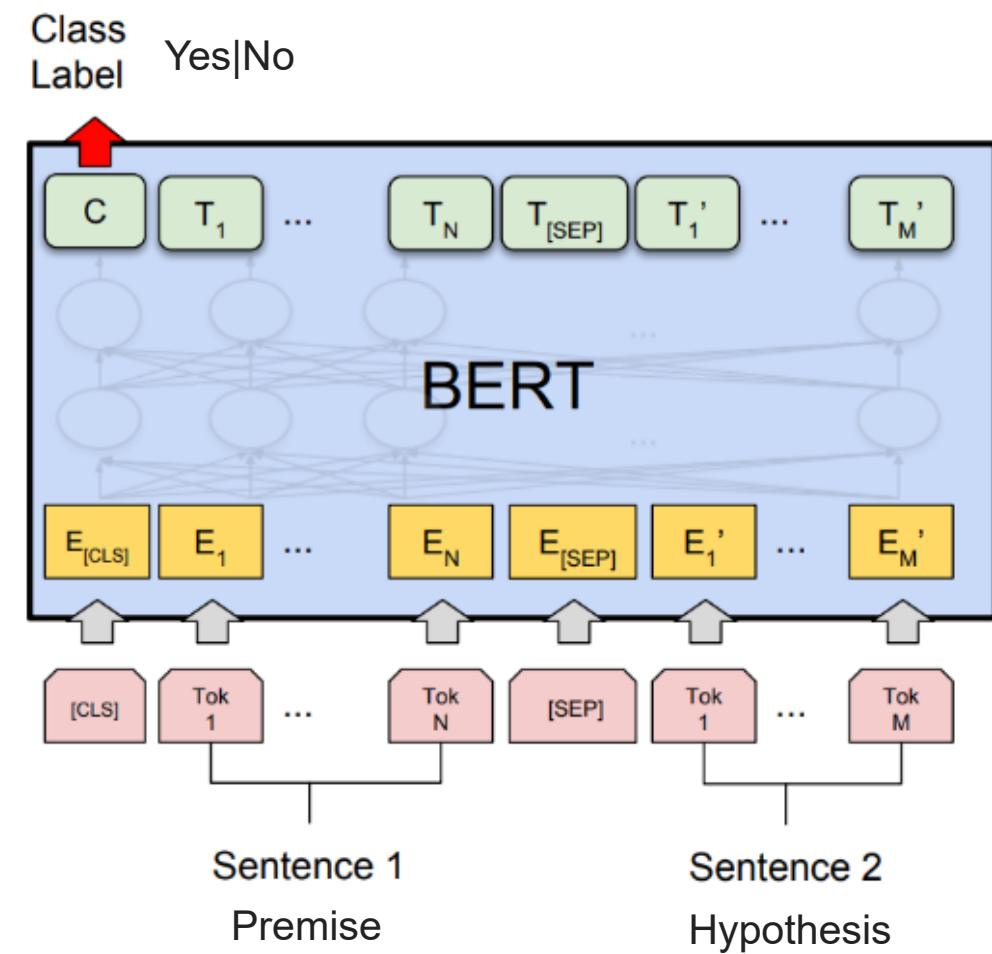
- More specific questions get better results, but do we have to explicitly extract these expressions?

```
result = qa(question="What food items are mentioned?", context=example)
print(f"Answer: '{result['answer']}', score: {round(result['score'], 4)}")
```

Answer: 'braised beef noodles', score: 0.1071

ABSA as natural language inference

- To transform the ABSA into an NLI task
- If the premise can entail the hypothesis, e.g., for product reviews
 - Premise: the review
 - Hypothesis: aspect and polarity
- Models trained to perform NLI can be used to do such ABSA, e.g. BERT+MNLI, BART+MNLI



ABSA as natural language inference



Figure 1: CORN’s prediction on three ABSA (AE, ASC, and E2E) tasks (Best view in color). **Red**, **blue**, and **pink** blocks indicate AE, ASC, and E2E tasks, respectively.

- AE: aspect extraction
- ASC: aspect sentiment classification
- E2E: end-to-end ABSA

Shu, Lei, et al. "Zero-shot aspect-based sentiment analysis." *arXiv preprint arXiv:2202.01924* (2022).

Task	Input	Premise	Hypothesis	NLI labels → ABSA labels
AE	(X, S)	X	$Restaurant \text{ has } S.$	$entail. \rightarrow T, \{neutral, contra.\} \rightarrow Outside$
ASC	(X, A)	X	$A \text{ is great.}$	$entail. \rightarrow POS, neutral \rightarrow NEU, contra. \rightarrow NEG$
E2E(step1)	(X, S)	X	$Restaurant \text{ has } S.$	$entail. \rightarrow <\text{go to step2}>, \{neutral, contra.\} \rightarrow Outside$
E2E(step2)	(X, S)	X	$S \text{ is great.}$	$entail. \rightarrow T-POS, neutral \rightarrow T-NEU, contra. \rightarrow T-NEG$

Prompt-based approaches with LLMs

- Using generative language models (like ChatGPT) further trained to predict responses to the instructions given in the input
- What is prompt?
 - User input/instruction to the model to guide (condition) the model's generation.
 - May include context, desired response, and input
- **The quality and specificity of the prompt can greatly influence the model's response**

What is the sentiment and its target in the quoted sentence? The quoted sentence is: "Although the service is not that great, I still love this reastaurant."

- ✓ Searching for: sentiment analysis "Although the service is not that great, I still love this restaurant."
- ✓ Generating answers for you...

The sentiment in the quoted sentence is positive and the target is the restaurant. I hope this helps!

Prompt engineering

List all sentiments in the quoted text in the form of entity/aspect/sentiment, and return the results in json format. If the aspect of a sentiment is not explicitly mentioned, please put aspect as "overall". Here's the quoted text: "Although the service is not that great, I still love this restaurant."

- ✓ Searching for: sentiment analysis "Although the service is not that great, I still love this restaurant."
- ✓ Generating answers for you...

Here is the sentiment analysis of the text you provided in the form of entity/aspect/sentiment:

```
{  
  "entity": "restaurant",  
  "aspect": "service",  
  "sentiment": "negative"  
},  
{  
  "entity": "restaurant",  
  "aspect": "overall",  
  "sentiment": "positive"  
}
```



The sentiment analysis of the text you provided is negative towards the service aspect of the restaurant entity

Inference for implicit aspects

"The camera is expensive and doesn't fit in the pocket."

Here's the sentiment analysis of the quoted text you provided:

```
{  
  "entity": "camera",  
  "aspect": "price",  
  "sentiment": "negative"  
},  
{  
  "entity": "camera",  
  "aspect": "size",  
  "sentiment": "negative"  
}
```



The sentiment analysis shows that the entity is a camera and the aspects are price and size. Both aspects have negative sentiment. I hope this helps!

Getting aspect categories from LLM

What are the common aspect categories for digital cameras in customer reviews?

Aspect categories are the features or attributes of a product or service that customers evaluate and express their opinions on. For digital cameras, some of the common aspect categories that customers may mention in their reviews are:

- **Image quality:** This refers to how well the digital camera captures the details, colors, and contrast of the scenes and subjects in photos and videos. Customers may comment on the resolution, sharpness, noise, dynamic range, and white balance of the images produced by the camera.
- **Zoom:** This refers to how much the digital camera can magnify the view of distant objects or scenes. Customers may comment on the optical zoom, which uses the lens to change the focal length, and the digital zoom, which uses software to crop and enlarge the image. Customers may also comment on the clarity and stability of the zoomed images.
- **Battery life:** This refers to how long the digital camera can operate on a single charge or a set of batteries. Customers may comment on the number of shots, hours, or days they can use the camera before needing to recharge or replace the batteries. Customers may also comment on the charging speed and convenience of the camera.
- **Ease of use:** This refers to how user-friendly and intuitive the digital camera is to operate and adjust. Customers may comment on the design, layout, and functionality of the buttons, menus, and settings of the camera. Customers may also comment on the responsiveness, accuracy, and feedback of the camera's controls and features.
- **Size and weight:** This refers to the dimensions and mass of the digital camera and its components. Customers may comment on how portable, comfortable, and durable the camera is to carry and hold. Customers may also comment on the balance, grip, and ergonomics of the camera.

These are some of the common aspect categories for digital cameras in customer reviews, but there may be others depending on the specific model, brand, or customer preference. You can learn more about aspect-based sentiment analysis and its applications from this [article](#) or this [glossary](#). I hope this helps you understand the concept better. 😊



Prompt with multi-turn QA

- For tasks that're too complex for one-time prediction
 - Divide the task into several parts (stages), e.g., identify two entities, then determine their relation
 - Instruct the model to predict response stage by stage, i.e., multi-turn QA
- Chained information extraction, depending on each element type from Stage I, e.g., event classification followed by event argument extraction
- Task specific templates designed for prompts and previously extracted information

Wei, Xiang, et al. "Zero-shot information extraction via chatting with chatgpt." *arXiv preprint arXiv:2302.10205* (2023).

Multi-turn QA: entity type -> entity name

- Stage I
- Stage II

Question:

Given sentence: "Japan then laid siege to the Syrian penalty area and had a goal disallowed for offside in the 16th minute." The known entity types are: ['LOC', 'MISC', 'ORG', 'PER']. Please answer: What types of entities are included in this sentence?

Expected Output: LOC, MISC **Output:** LOC, MISC

Question:

According to the sentence above, please output the entities of 'LOC' in the form of list like: ['entity name1', 'entity type1'], ['entity name2', 'entity type2'].....

According to the sentence above, please output the entities of 'MISC' in the form of list like: ['entity name1', 'entity type1'], ['entity name2', 'entity type2'].....

Expected Output: ["Japan", "LOC"], ["Syrian", "MISC"]**Output:** ["Japan", "LOC"], ["Syrian", "LOC"]

Multi-turn QA: relations -> entities

- Stage I

Question:

The given sentence is " Bono said that President Jacques Chirac of France had spoken eloquently of the need to support Africa , though he added that France had not yet come through with the resources ."

List of given relations: ['location-located_in', 'administrative_division-country', 'person-place_lived', 'person-company', 'person-nationality', 'company-founders', 'country-administrative_divisions', 'person-children', 'country-capital', 'deceased_person-place_of_death', 'neighborhood-neighborhood_of', 'person-place_of_birth']

What relations in the given list might be included in this given sentence?

If not present, answer: none.

Respond as a tuple, e.g. (relation 1, relation 2,):

Expected Output: (person-nationality) **Output:** (person-nationality)

- Stage II

Question:

According to the given sentence, the two entities are of type ('person', 'country') and the relation between them is 'person-nationality', find the two entities and list them all by group if there are multiple groups.

If not present, answer: none.

Respond in the form of a table with two columns and a header of ('person', 'country'): -----

Expected Output: (Jacques Chirac, France)

Output: (Jacques Chirac, France)

Multi-turn QA: event type -> arguments

- Stage I

Question:

The list of event types: ['Life:Die', 'Justice:Arrest-Jail', 'Contact:Phone-Write', 'Life:Marry', 'Conflict:Attack', 'Personnel:Nominate', 'Business:Declare-Bankruptcy', 'Justice:Sue']

Give a sentence: "What I do know is Saddam Hussein has butchered over a million of his own citizens."

What types of events are included in this sentence?

Please return the most likely answer according to the list of event types above.

Require the answer in the form: Event type

Expected Output: Life:Die **Output:** Life:Die

- Stage II

Question:

The list of argument roles corresponding to the event type 'Life: Die' is ['Agent', 'Victim', 'Instrument', 'Time', 'Place'].

please extract the event arguments in the given sentence according to the argument roles, and return them in the form of a table. The header of the table is 'event type', 'argument role', 'argument content'.

If no argument role has a corresponding argument content, the argument content returns "None".

Expected Output: "arguments": ["role": "Victim", "argument": "over a million of his own citizens" , { "role": "Agent", "argument": "Saddam Hussein" } **Output:** "arguments": ["role": "Victim", "argument": "over a million of his own citizens" , { "role": "Agent", "argument": "Saddam Hussein" }]

References

- Liu, Bing. "Chapter 6: Aspect and Entity Extraction". *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.
- Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis Lectures on Human Language Technologies* 5.1 (2012): 1-167.
- Pang, Bo, and Lillian Lee. "Opinion Mining and Sentiment Analysis." *Foundations and Trends in Information Retrieval* 2.1-2 (2008): 1-135.
- Stanford Typed Dependencies Manual
(https://nlp.stanford.edu/software/dependencies_manual.pdf)
- Universal Dependencies
(<https://universaldependencies.org/u/dep/all.html/>)

Workshop

- ABSA tutorials
 - The heuristic approach
 - The supervised-learning approach
 - The zero-shot classification approach
- Exercise:
 - Use the restaurant review data set from Day 2 Workshop: train.csv
 - Select two restaurants (each with at least 100 reviews)
 - Data cleaning – filter off reviews with too many sentences
 - Apply ABSA on the two restaurants
 - Aspect categories – food quality, service, ambiance, price/value, cleanliness, location, general
 - You may use any model or technique learned in class.
 - Submit a pdf file (NMSM-D3-yourname.pdf) converted from your ipynb file including the results and your findings.