Tamkang University 淡江大學

Tamkang University

Social Computing and Big Data Analytics

社群運算與大數據分析

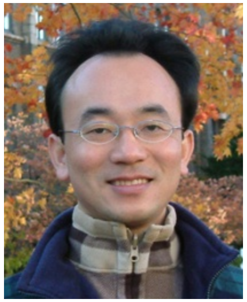
Sentiment Analysis on Social Media with Deep Learning

(深度學習社群媒體情感分析)

1052SCBDA11

MIS MBA (M2226) (8606)

Wed, 8,9, (15:10-17:00) (L206)



Min-Yuh Day

戴敏育

Assistant Professor

專任助理教授

Dept. of Information Management, Tamkang University

淡江大學 資訊管理學系

http://mail. tku.edu.tw/myday/

2017-05-10

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2017/02/15	Course Orientation for Social Computing and Big Data Analytics (社群運算與大數據分析課程介紹)
2	2017/02/22	Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data (資料科學與大數據分析：探索、分析、視覺化與呈現資料)
3	2017/03/01	Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem (大數據基礎：MapReduce典範、Hadoop與Spark生態系統)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
4	2017/03/08	Big Data Processing Platforms with SMACK: Spark, Mesos, Akka, Cassandra and Kafka (大數據處理平台SMACK： Spark, Mesos, Akka, Cassandra, Kafka)
5	2017/03/15	Big Data Analytics with Numpy in Python (Python Numpy 大數據分析)
6	2017/03/22	Finance Big Data Analytics with Pandas in Python (Python Pandas 財務大數據分析)
7	2017/03/29	Text Mining Techniques and Natural Language Processing (文字探勘分析技術與自然語言處理)
8	2017/04/05	Off-campus study (教學行政觀摩日)

課程大綱 (Syllabus)

週次 (Week)　日期 (Date)　內容 (Subject/Topics)

9　2017/04/12　Social Media Marketing Analytics
(社群媒體行銷分析)

10　2017/04/19　期中報告 (Midterm Project Report)

11　2017/04/26　Deep Learning with Theano and Keras in Python
(Python Theano 和 Keras 深度學習)

12　2017/05/03　Deep Learning with Google TensorFlow
(Google TensorFlow 深度學習)

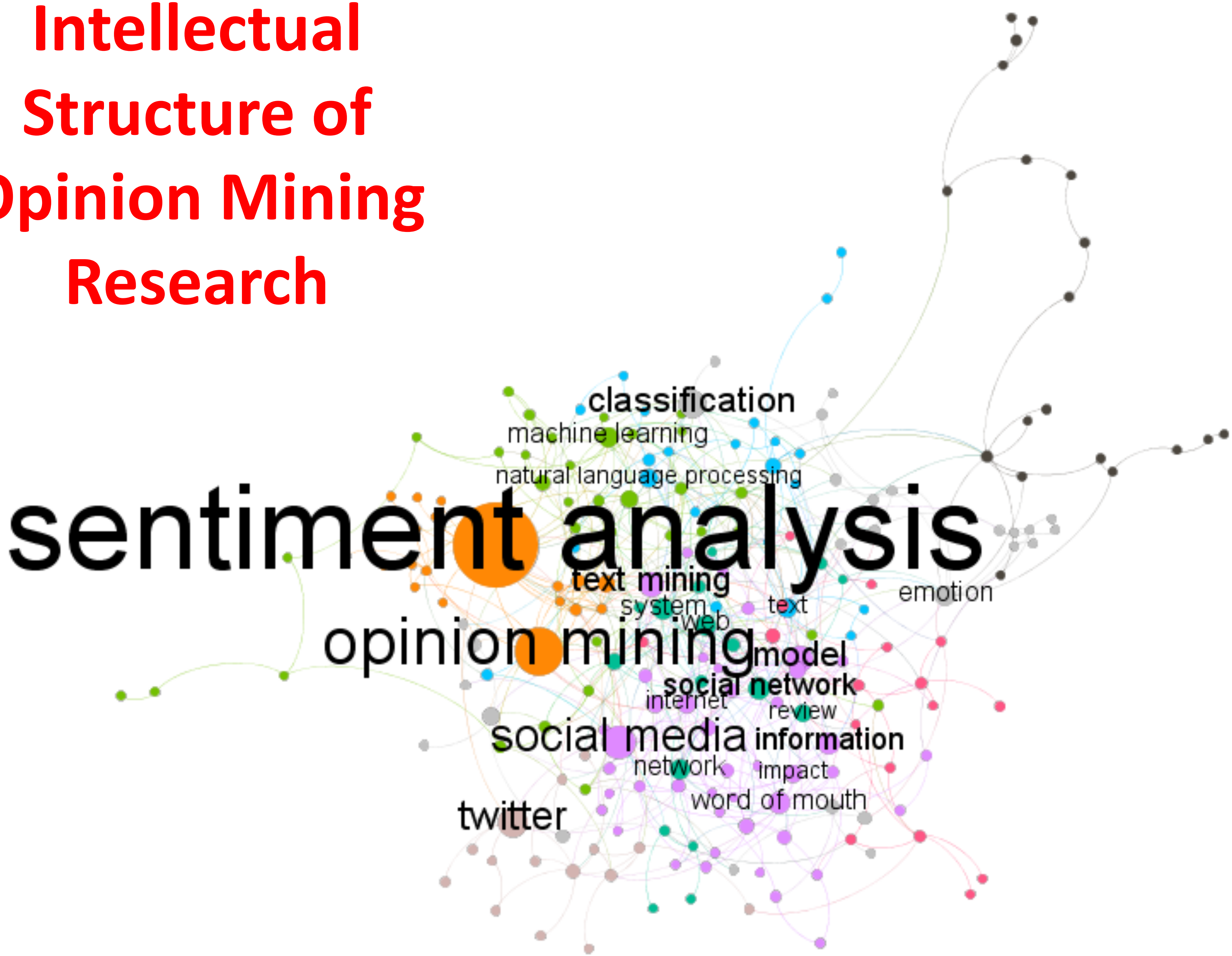
13　2017/05/10　Sentiment Analysis on Social Media with Deep Learning
(深度學習社群媒體情感分析)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
14	2017/05/17	Social Network Analysis (社會網絡分析)
15	2017/05/24	Measurements and Tools of Social Network Analysis (社會網絡分析量測與工具)
16	2017/05/31	Invited Talk: From Blog to Job Bank (社群平台分析) [Invited Speaker: Dr. Rick Cheng-Yu Lu, CDO, 104]
17	2017/06/07	Final Project Presentation I (期末報告 I)
18	2017/06/14	Final Project Presentation II (期末報告 II)

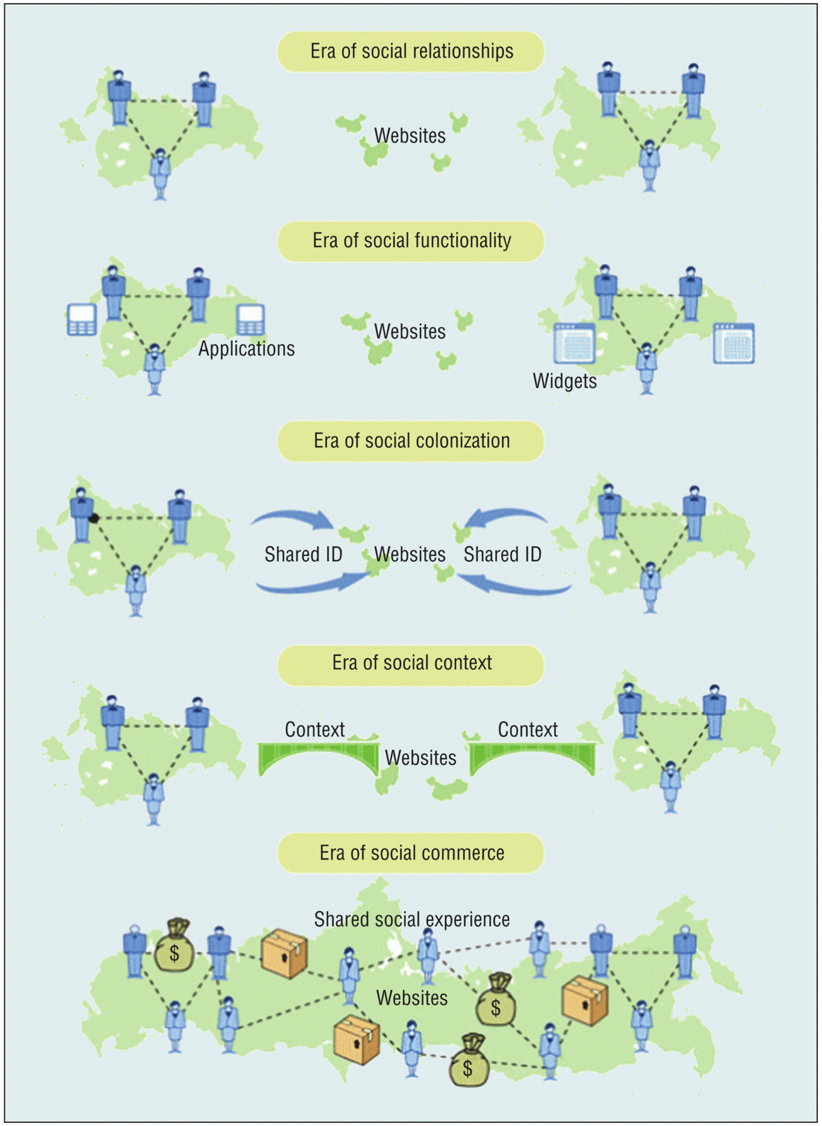
Sentiment Analysis
on
Social Media
with
Deep Learning

Intellectual Structure of Opinion Mining Research



Source: Zhu, Yongjun, Meen Chul Kim, and Chaomei Chen.
"An investigation of the intellectual structure of opinion mining research." *Information Research* 22, no. 1 (2017).

The Five-eras Vision of Affective Computing and Sentiment Analysis



1. Era of Social Relationships
2. Era of Social Functionality
3. Era of Social Colonization
4. Era of Social Context
5. Era of Social Commerce

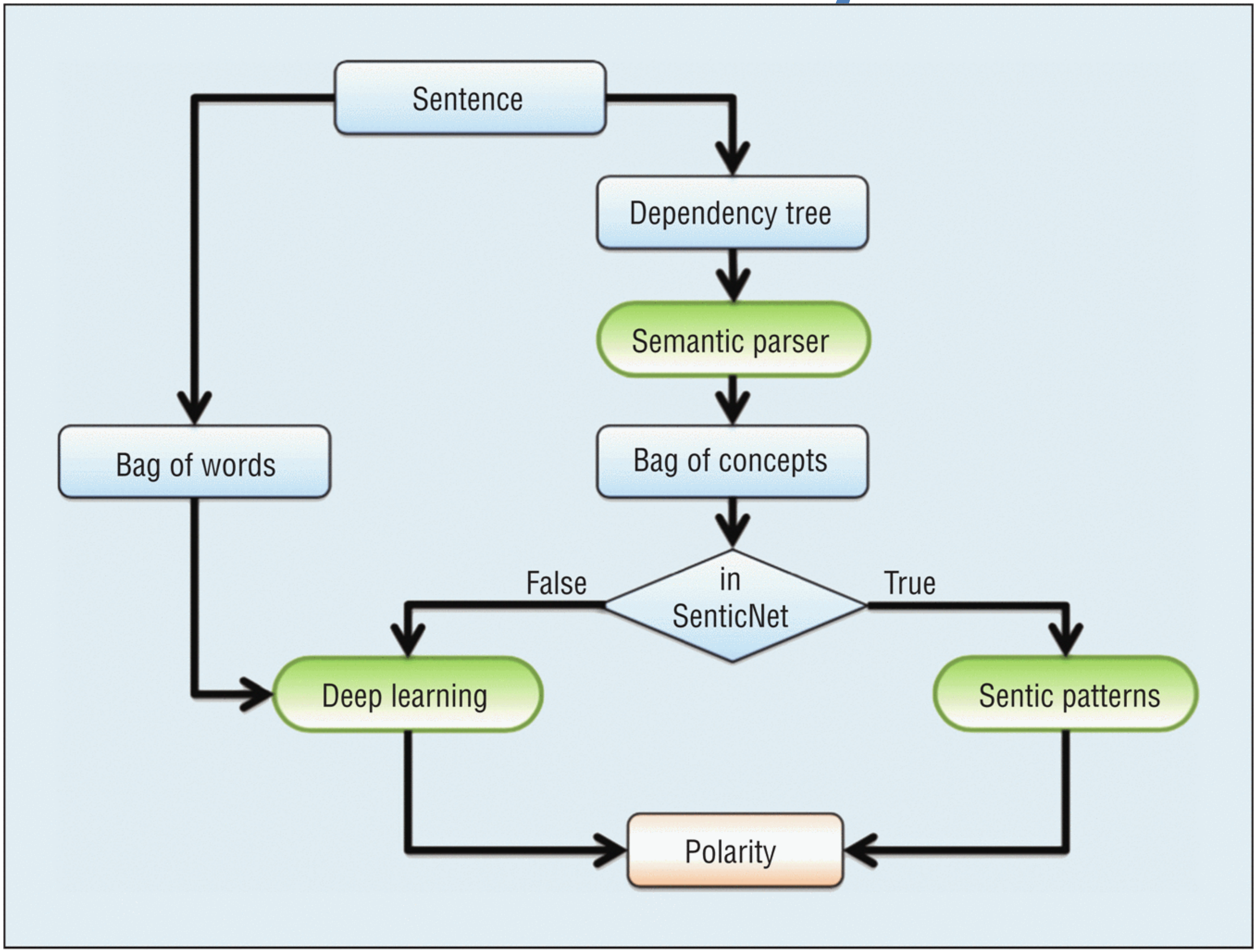
Source: Cambria, Erik. "Affective computing and sentiment analysis." *IEEE Intelligent Systems* 31, no. 2 (2016): 102-107.

The Five-eras Vision of Affective Computing and Sentiment Analysis

5. Era of Social Commerce

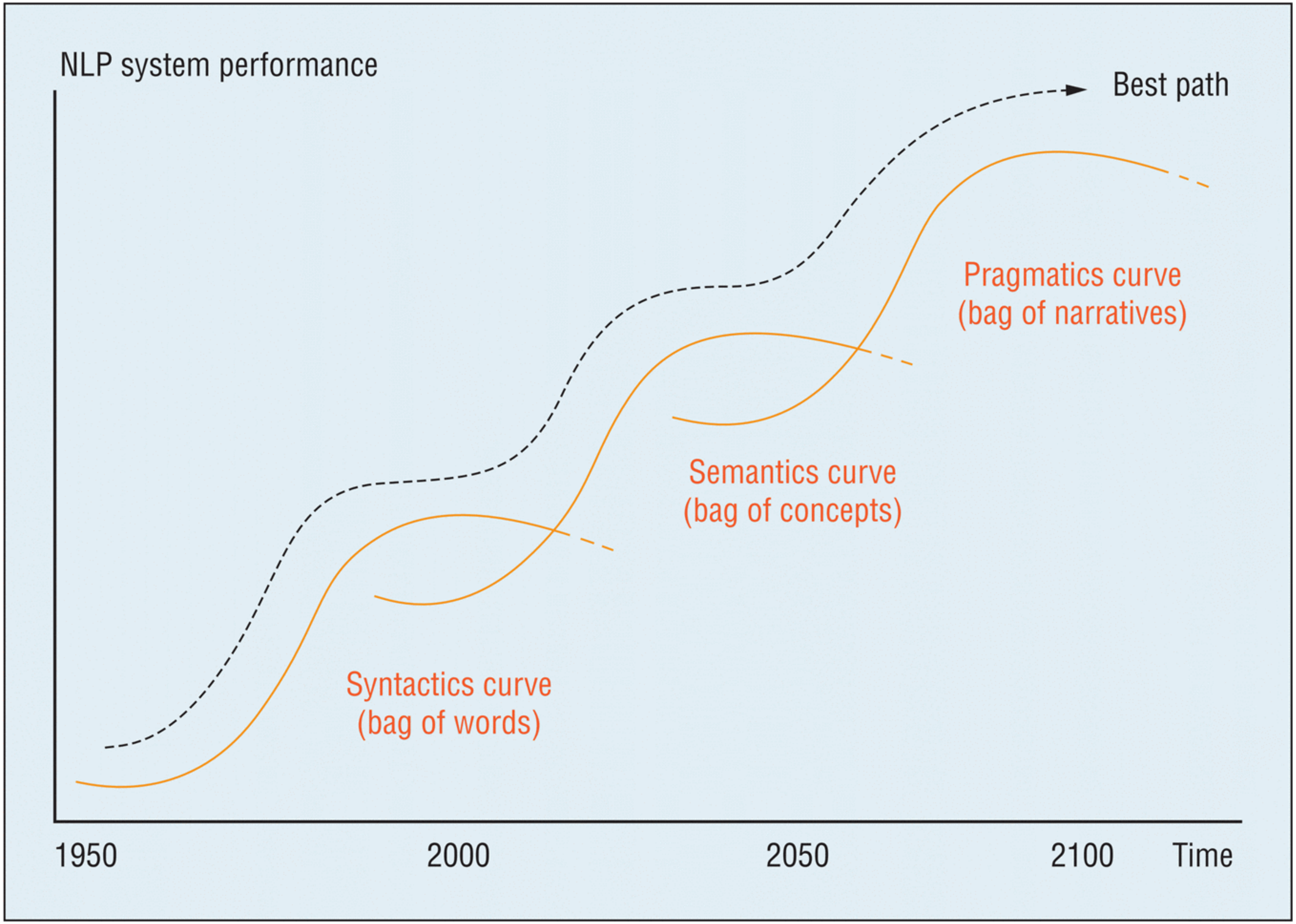
Source: Cambria, Erik. "Affective computing and sentiment analysis." *IEEE Intelligent Systems* 31, no. 2 (2016): 102-107.

Sentic Computing's Hybrid Framework for Polarity Detection



Source: Cambria, Erik. "Affective computing and sentiment analysis." *IEEE Intelligent Systems* 31, no. 2 (2016): 102-107.

Evolution of Natural Language Processing (NLP) Research



Source: Cambria, Erik. "Affective computing and sentiment analysis." *IEEE Intelligent Systems* 31, no. 2 (2016): 102-107.

Sentiment Analysis

Example of Opinion: review segment on iPhone

"I bought an iPhone a few days ago.

It was such a nice phone.

The touch screen was really cool.

The voice quality was clear too.

However, my mother was mad with me as I did not tell her before I bought it.

She also thought the phone was too expensive, and wanted me to return it to the shop. … "

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Example of Opinion: review segment on iPhone

"(1) I bought an iPhone a few days ago.

(2) It was such a **nice** phone.

(3) The touch screen was really **cool**.

(4) The voice quality was **clear** too.

+Positive Opinion

(5) However, my mother was mad with me as I did not tell her before I bought it.

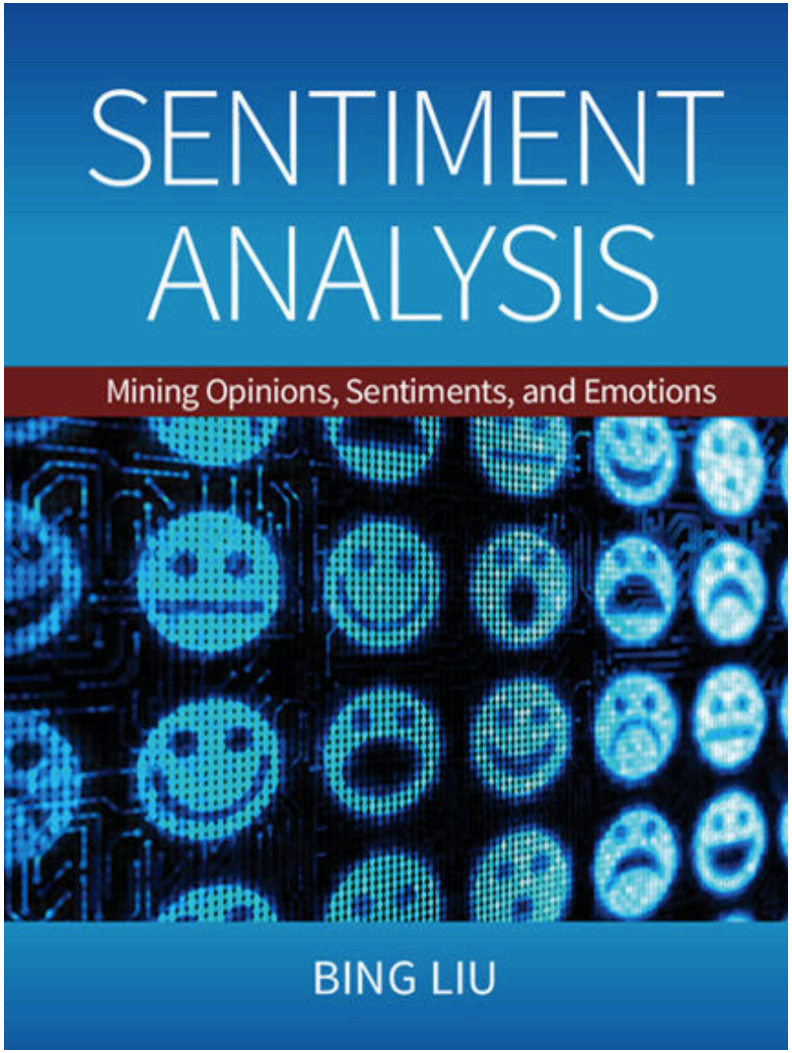
(6) She also thought the phone was too **expensive**, and wanted me to return it to the shop. … "

-Negative Opinion

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Architectures of Sentiment Analytics

Bing Liu (2015),
Sentiment Analysis:
Mining Opinions, Sentiments, and Emotions,
Cambridge University Press



http://www.amazon.com/Sentiment-Analysis-Opinions-Sentiments-Emotions/dp/1107017890

Sentiment Analysis and Opinion Mining

- Computational study of
  opinions,
  sentiments,
  subjectivity,
  evaluations,
  attitudes,
  appraisal,
  affects,
  views,
  emotions,
  ets., expressed in text.
  - Reviews, blogs, discussions, news, comments, feedback, or any other documents

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Research Area of Opinion Mining

- Many names and tasks with difference objective and models
  - Sentiment analysis
  - Opinion mining
  - Sentiment mining
  - Subjectivity analysis
  - Affect analysis
  - Emotion detection
  - Opinion spam detection

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Sentiment Analysis

- Sentiment
  - A thought, view, or attitude, especially one based mainly on emotion instead of reason
- Sentiment Analysis
  - opinion mining
  - use of natural language processing (NLP) and computational techniques to automate the extraction or classification of sentiment from typically unstructured text

Applications of Sentiment Analysis

- Consumer information
  - Product reviews
- Marketing
  - Consumer attitudes
  - Trends
- Politics
  - Politicians want to know voters’ views
  - Voters want to know policitians’ stances and who else supports them
- Social
  - Find like-minded individuals or communities

Sentiment detection

- How to interpret features for sentiment detection?
  - Bag of words (IR)
  - Annotated lexicons (WordNet, SentiWordNet)
  - Syntactic patterns
- Which features to use?
  - Words (unigrams)
  - Phrases/n-grams
  - Sentences

Problem statement of Opinion Mining

- Two aspects of abstraction
  - Opinion definition
    - What is an opinion?
    - What is the structured definition of opinion?
  - Opinion summarization
    - Opinion are subjective
      - An opinion from a single person (unless a VIP) is often not sufficient for action
    - We need opinions from many people, and thus opinion summarization.

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

What is an opinion?

- Id: **Abc123** on **5-1-2008** "*I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …*"
- One can look at this review/blog at the
  - Document level
    - Is this review + or -?
  - Sentence level
    - Is each sentence + or -?
  - Entity and feature/aspect level

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Entity and aspect/feature level

- Id: **Abc123** on **5-1-2008** *"I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …"*
- What do we see?
  - Opinion targets: entities and their features/aspects
  - Sentiments: positive and negative
  - Opinion holders: persons who hold the opinions
  - Time: when opinion are expressed

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Two main types of opinions

- Regular opinions: Sentiment/Opinion expressions on some target entities
  - Direct opinions: sentiment expressions on one object:
    - "The touch screen is really cool."
    - "The picture quality of this camera is great"
  - Indirect opinions: comparisons, relations expressing similarities or differences (objective or subjective) of more than one object
    - "phone X is cheaper than phone Y." (objective)
    - "phone X is better than phone Y." (subjective)
- Comparative opinions: comparisons of more than one entity.
  - "iPhone is better than Blackberry."

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Subjective and Objective

- Objective
  - An objective sentence expresses some factual information about the world.
  - "I returned the phone yesterday."
  - Objective sentences can implicitly indicate opinions
    - "The earphone broke in two days."
- Subjective
  - A subjective sentence expresses some personal feelings or beliefs.
  - "The voice on my phone was not so clear"
  - Not every subjective sentence contains an opinion
    - "I wanted a phone with good voice quality"
- ➔ Subjective analysis

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Sentiment Analysis vs. Subjectivity Analysis

Sentiment Analysis	Subjectivity Analysis
Positive	Subjective
Negative	
Neutral	Objective

A (regular) opinion

- Opinion (a restricted definition)
  - An opinion (regular opinion) is simply a positive or negative sentiment, view, attitude, emotion, or appraisal about an entity or an aspect of the entity from an opinion holder.
- Sentiment orientation of an opinion
  - Positive, negative, or neutral (no opinion)
  - Also called:
    - Opinion orientation
    - Semantic orientation
    - Sentiment polarity

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Entity and aspect

- Definition of Entity:
  - An *entity e* is a product, person, event, organization, or topic.
  - e is represented as
    - A hierarchy of components, sub-components.
    - Each node represents a components and is associated with a set of attributes of the components
- An opinion can be expressed on any node or attribute of the node
- Aspects(features)
  - represent both components and attribute

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Opinion Definition

- An opinion is a quintuple
  (e_j, a_{jk}, so_{ijkl}, h_i, t_l)
  where
  - e_j is a target entity.
  - a_{jk} is an aspect/feature of the entity e_j .
  - so_{ijkl} is the sentiment value of the opinion from the opinion holder on feature of entity at time.
    so_{ijkl} is +ve, -ve, or neu, or more granular ratings
  - h_i is an opinion holder.
  - t_l is the time when the opinion is expressed.
- (e_j, a_{jk}) is also called opinion target

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Terminologies

- Entity: object
- Aspect: feature, attribute, facet
- Opinion holder: opinion source

- Topic: entity, aspect

- Product features, political issues

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Subjectivity and Emotion

- Sentence subjectivity
  - An objective sentence presents some factual information, while a subjective sentence expresses some personal feelings, views, emotions, or beliefs.
- Emotion
  - Emotions are people's subjective feelings and thoughts.

Classification Based on Supervised Learning

- Sentiment classification
  - Supervised learning Problem
  - Three classes
    - *Positive*
    - *Negative*
    - *Neutral*

Opinion words in Sentiment classification

- topic-based classification
  - topic-related words are important
    - e.g., *politics, sciences, sports*
- Sentiment classification
  - topic-related words are unimportant
  - **opinion words** (also called **sentiment words)**
    - **that indicate positive or negative opinions** are important,
      e.g., *great, excellent, amazing, horrible, bad, worst*

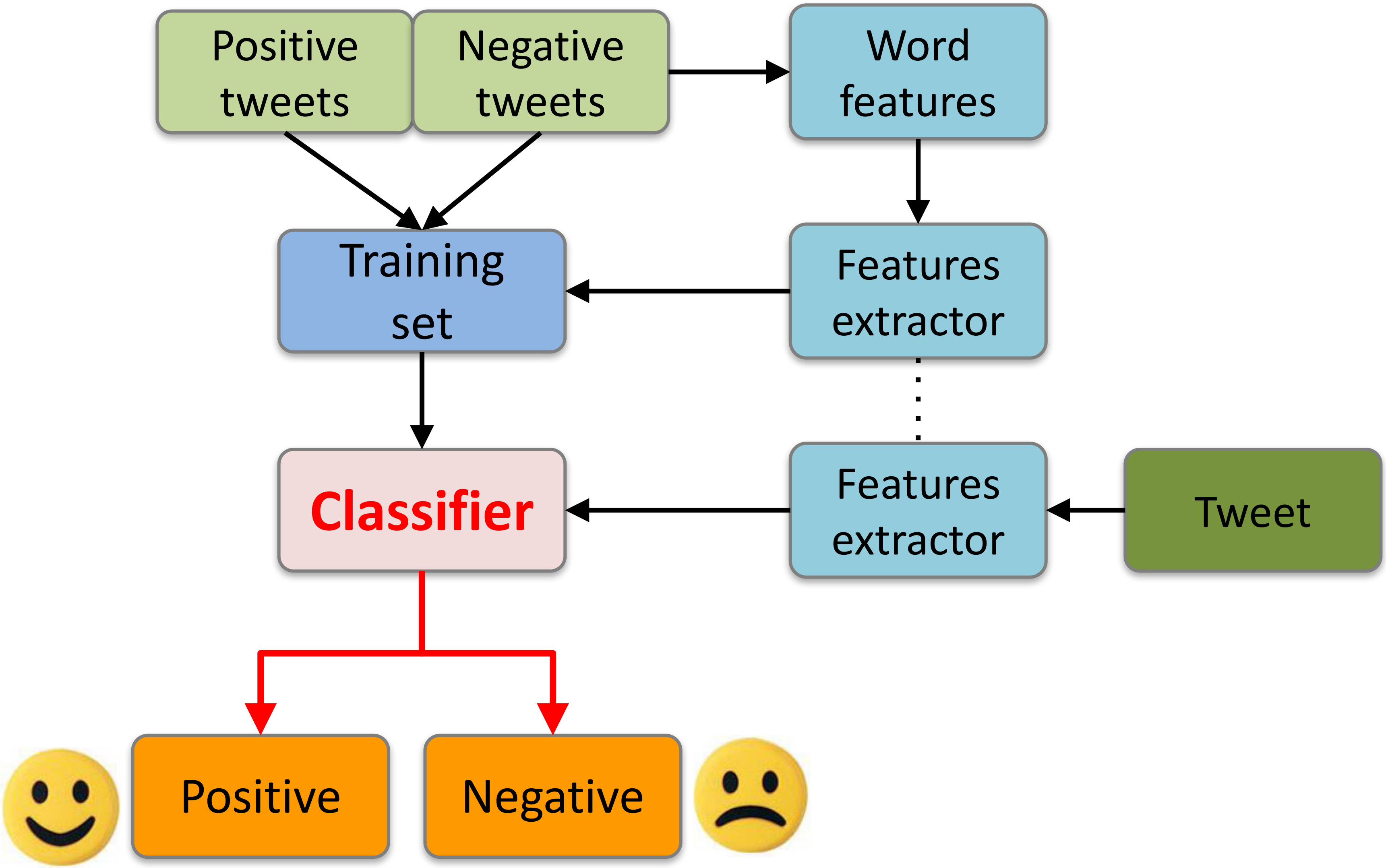
Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

Features in Opinion Mining

- *Terms and their frequency*
  - *TF-IDF*
- *Part of speech (POS)*
  - Adjectives
- *Opinion words and phrases*
  - *beautiful, wonderful, good, and amazing are positive opinion words*
  - *bad, poor, and terrible are negative opinion words.*
  - opinion phrases and idioms,
    e.g., *cost someone an arm and a leg*
- *Rules of opinions*
- *Negations*
- *Syntactic dependency*

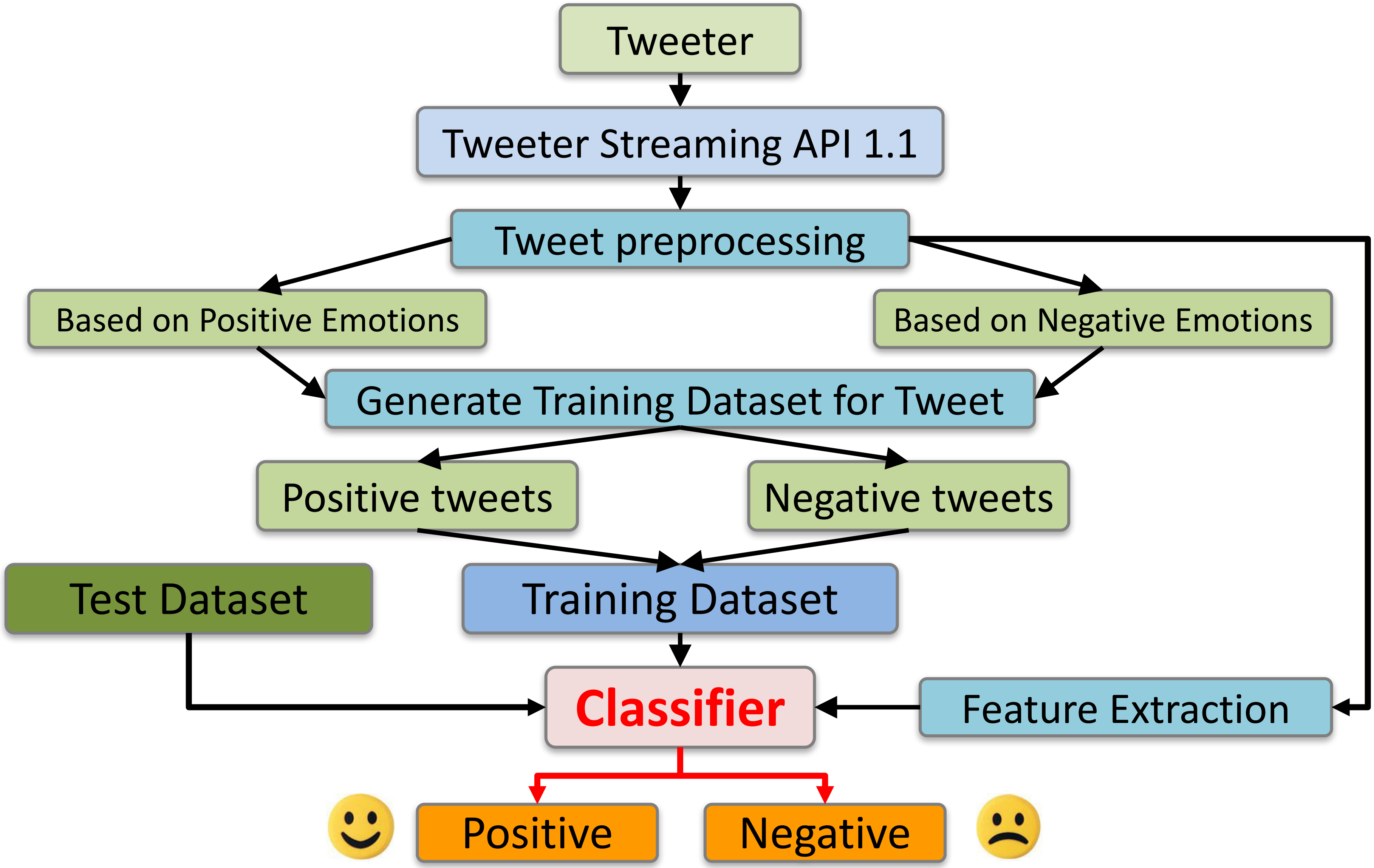
Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

Sentiment Analysis Architecture



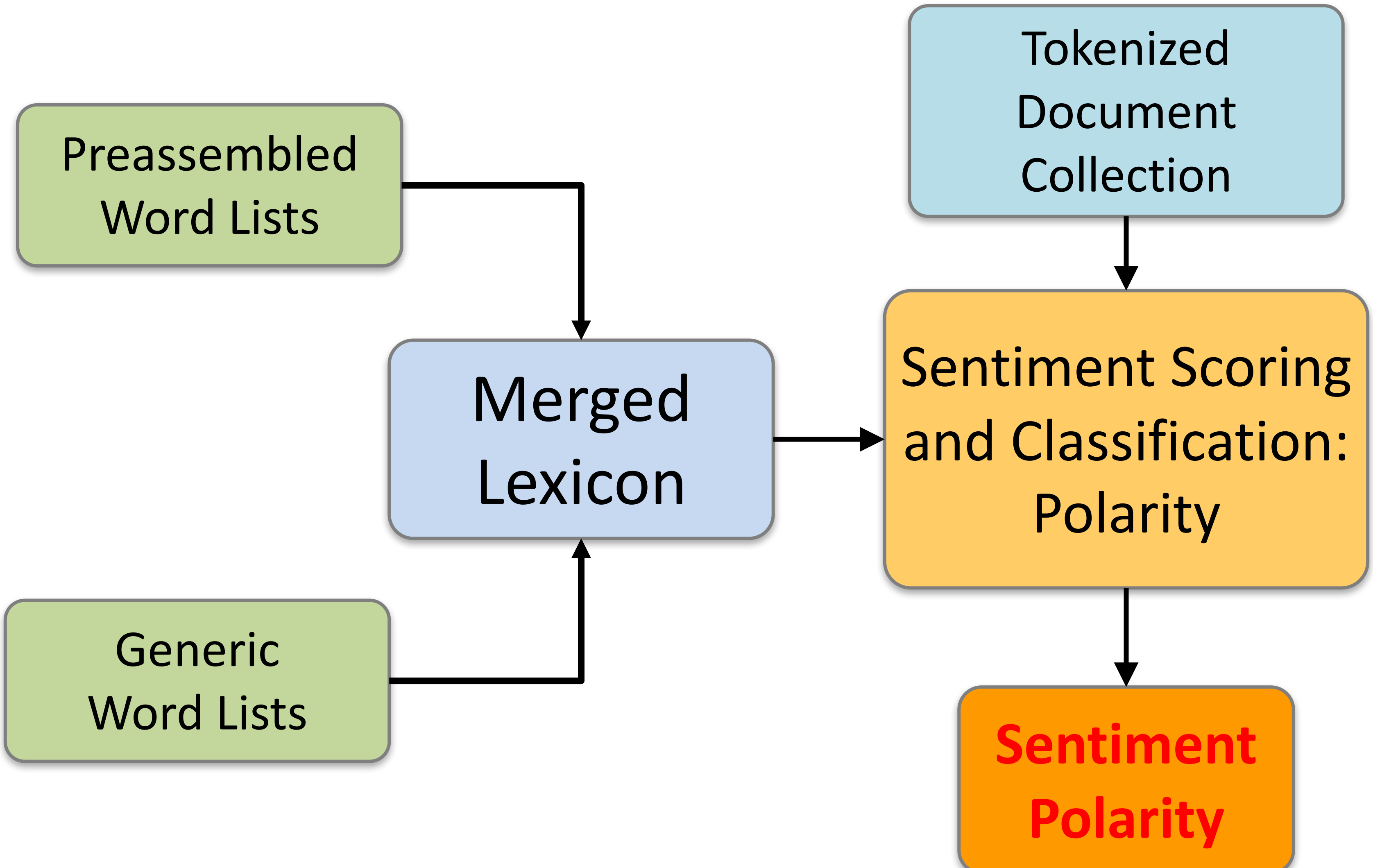
Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Sentiment Classification Based on Emoticons



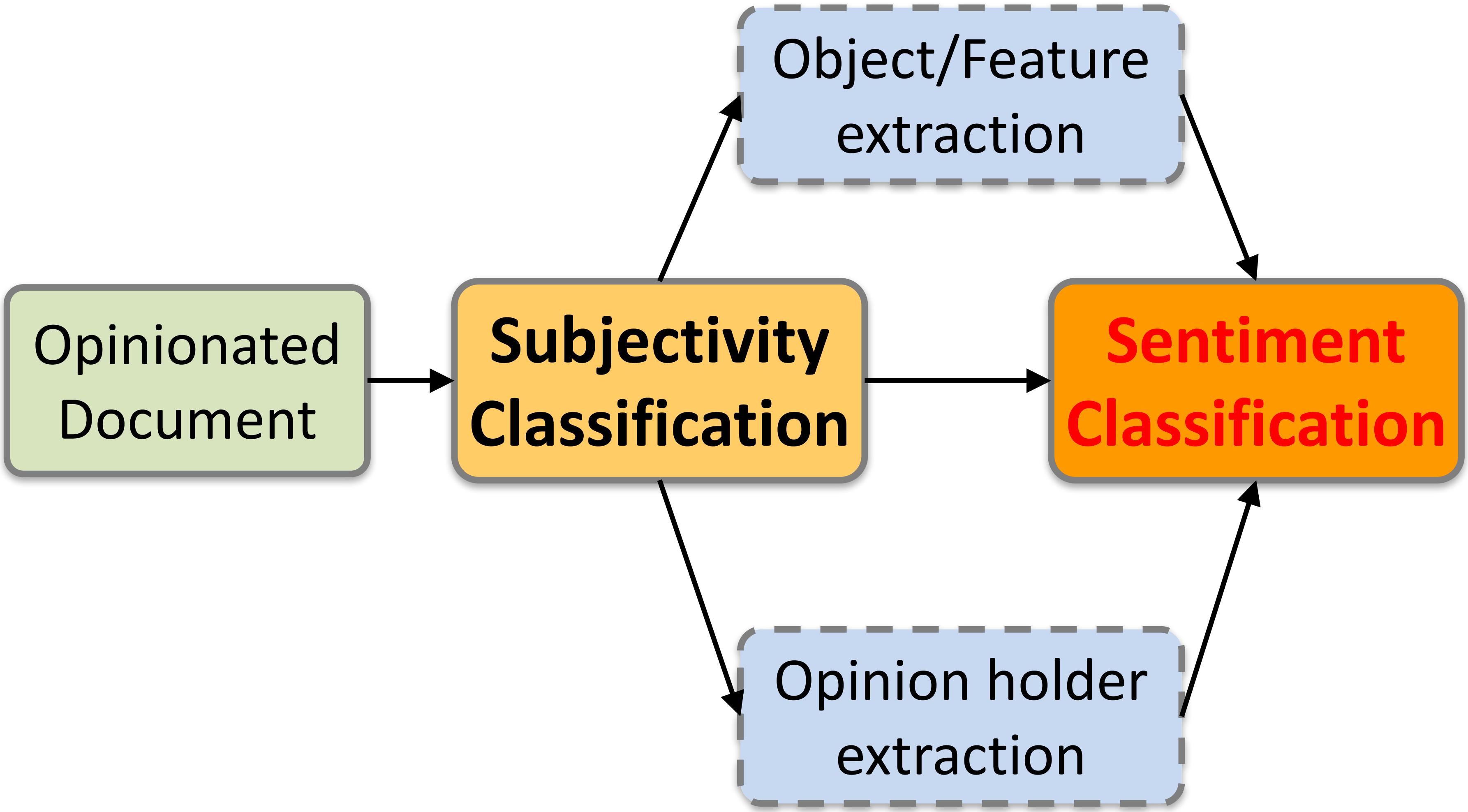
Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Lexicon-Based Model



Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Sentiment Analysis Tasks

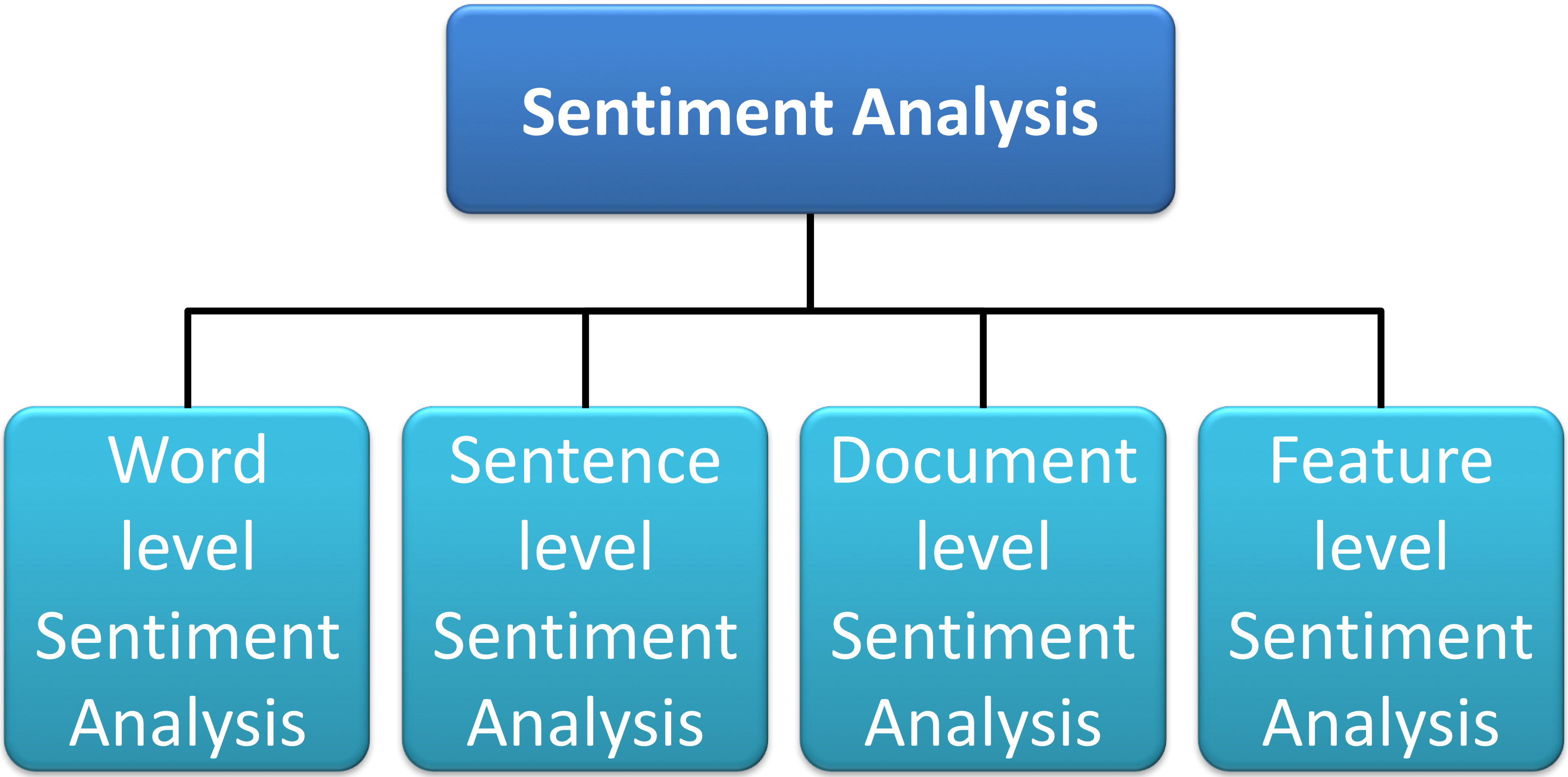


Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Sentiment Analysis
vs.
Subjectivity Analysis

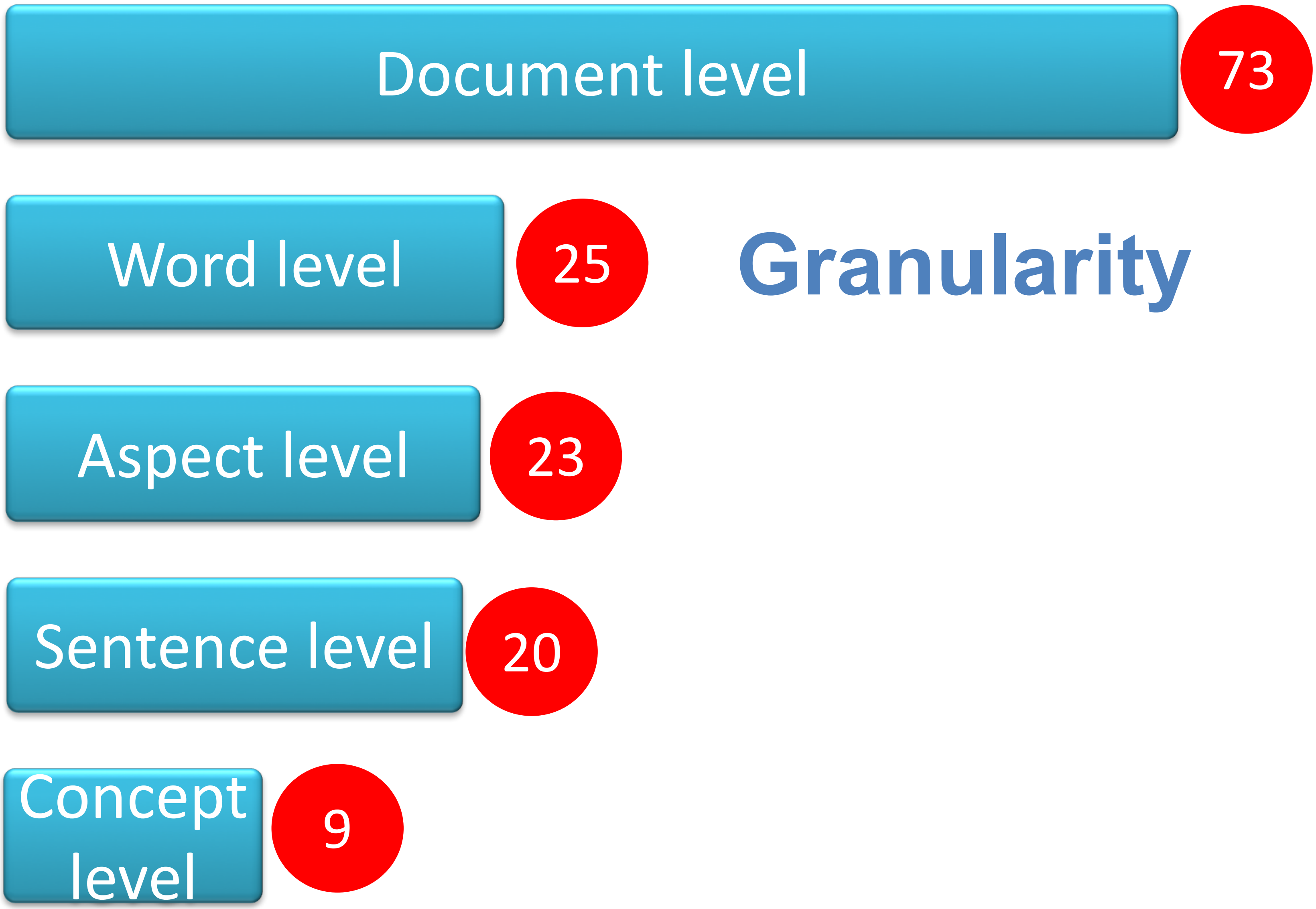
Sentiment Analysis	Subjectivity Analysis
Positive Negative	Subjective
Neutral	Objective

Levels of Sentiment Analysis



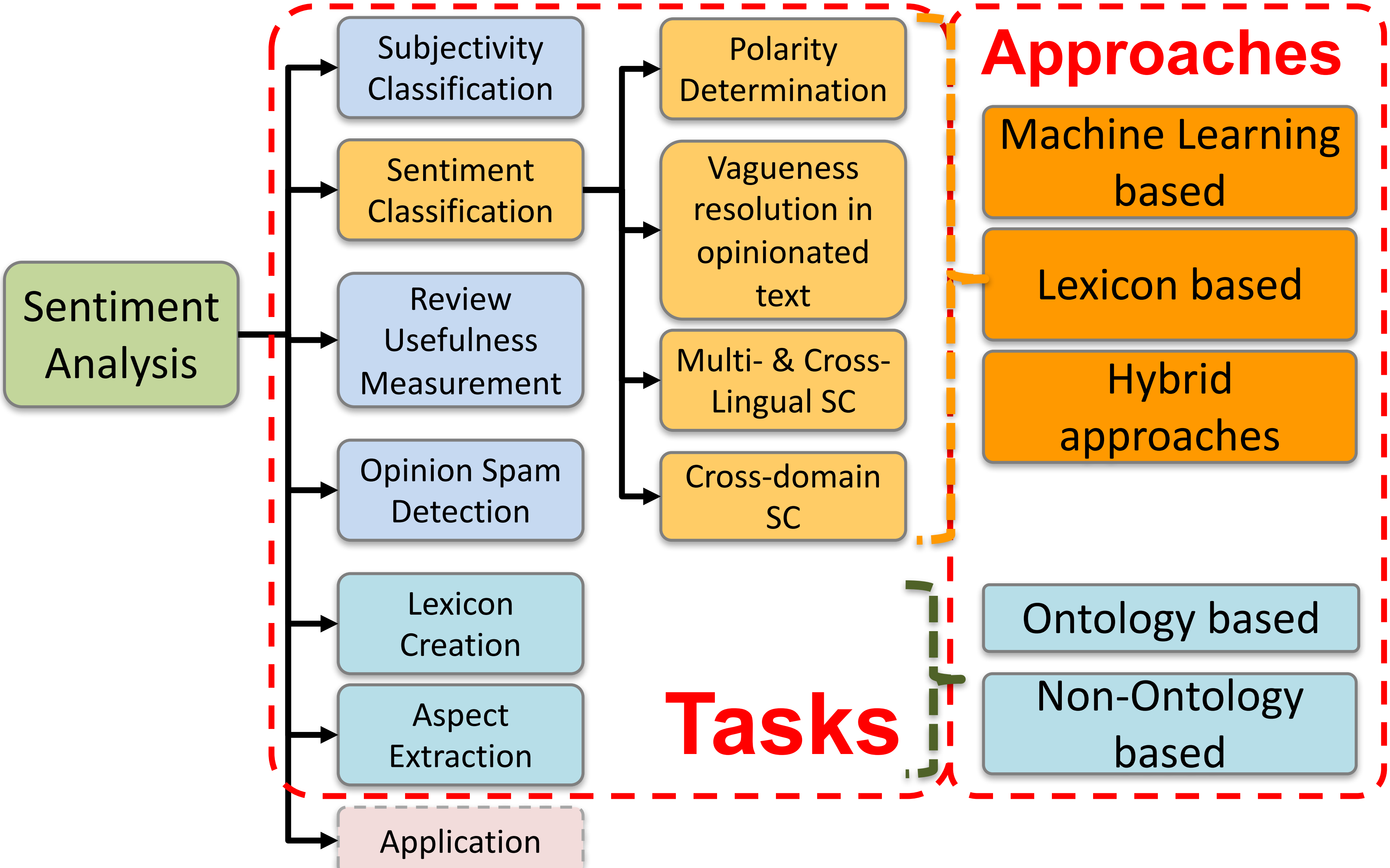
Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Levels of Sentiment Analysis



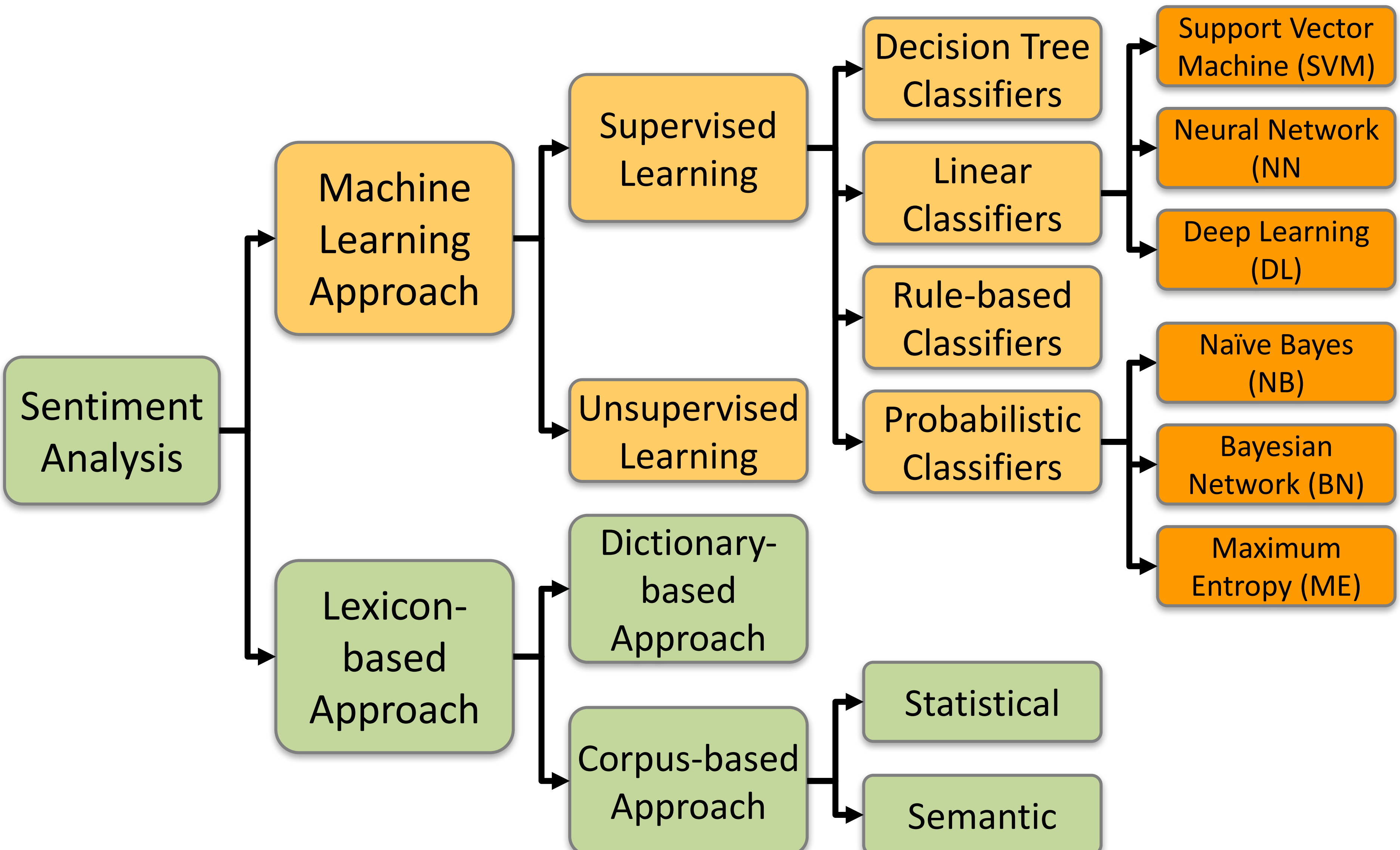
Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Analysis



Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Classification Techniques



Source: Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.

Machine Learning Models

Deep Learning	Kernel
Association rules	Ensemble
Decision tree	Dimensionality reduction
Clustering	Regression Analysis
Bayesian	Instance based

Source: Sunila Gollapudi (2016), Practical Machine Learning, Packt Publishing

A Brief Summary of Sentiment Analysis Methods

Study	Analysis Task	Sentiment Identification		Sentiment Aggregation		Nature of Measure
		Method	Level	Method	Level	
Hu and Li, 2011	Polarity	ML (Probabilistic model)	Snippet			Valence
Li and Wu, 2010	Polarity	Lexicon/Rule	Phrase	Sum	Snippet	Valence
Thelwall et al., 2010	Polarity	Lexicon/Rule	Sentence	Max & Min	Snippet	Range
Boiy and Moens, 2009	Both	ML (Cascade ensemble)	Sentence			Valence
Chung 2009	Polarity	Lexicon	Phrase	Average	Sentence	Valence
Wilson, Wiebe, and Hoffmann, 2009	Both	ML (SVM, AdaBoost, Rule, etc.)	Phrase			Valence
Zhang et al., 2009	Polarity	Lexicon/Rule	Sentence	Weighted average	Snippet	Valence
Abbasi, Chen, and Salem, 2008	Polarity	ML (GA + feature selection)	Snippet			Valence
Subrahmanian and Reforgiato, 2008	Polarity	Lexicon/Rule	Phrase	Rule	Snippet	Valence
Tan and Zhang 2008	Polarity	ML (SVM, Winnow, NB, etc.)	Snippet			Valence
Airoldi, Bai, and Padman, 2007	Polarity	ML (Markov Blanket)	Snippet			Valence
Das and Chen, 2007	Polarity	ML (Bayesian, Discriminate, etc.)	Snippet	Average	Daily	Valence
Liu et al., 2007	Polarity	ML (PLSA)	Snippet			Valence
Kennedy and Inkpen, 2006	Polarity	Lexicon/Rule, ML (SVM)	Phrase	Count	Snippet	Valence
Mishne 2006	Polarity	Lexicon	Phrase	Average	Snippet	Valence
Liu et al., 2005	Polarity	Lexicon/Rule	Phrase	Distribution	Object	Range
Mishne 2005	Polarity	ML (SVM)	Snippet			Valence
Popescu and Etzioni 2005	Polarity	Lexicon/Rule	Phrase			Valence
Efron 2004	Polarity	ML (SVN, NB)	Snippet			Valence
Wilson, Wiebe, and Hwa, 2004	Both	ML (SVM, AdaBoost, Rule, etc.)	Sentence			Valence
Nigam and Hurst 2004	Polarity	Lexicon/Rule	Chunk	Rule	Sentence	Valence
Dave, Lawrence, and Pennock, 2003	Polarity	ML (SVM, Rainbow, etc.)	Snippet			Valence
Nasukawa and Yi 2003	Polarity	Lexicon/Rule	Phrase	Rule	Sentence	Valence
Yi et al., 2003	Polarity	Lexicon/Rule	Phrase	Rule	Sentence	Valence
Yu and Hatzivassiloglou 2003	Both	ML (NB) + Lexicon/Rule	Phrase	Average	Sentence	Valence
Pang, Lee, and Vaithyanathan 2002	Polarity	ML (SVM, MaxEnt, NB)	Snippet			Valence
Subasic and Huettner 2001	Polarity	Lexicon/Fuzzy logic	Phrase	Average	Snippet	Valence
Turney 2001	Polarity	Lexicon/Rule	Phrase	Average	Snippet	Valence

(Both = Subjectivity and Polarity; ML= Machine Learning; Lexicon/Rule= Lexicon enhanced by linguistic rules)

Source: Zhang, Z., Li, X., and Chen, Y. (2012), "Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews," ACM Trans. Manage. Inf. Syst. (3:1) 2012, pp 1-23.,

Word-of-Mouth (WOM)

- “This book is the best written documentary thus far, yet sadly, there is no soft cover edition.”

- “This book is the best written documentary thus far, yet sadly, there is no soft cover edition.”

Source: Zhang, Z., Li, X., and Chen, Y. (2012), "Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews," ACM Trans. Manage. Inf. Syst. (3:1) 2012, pp 1-23.,

This
book
is
the
best
written
documentary
thus
far
,
yet
sadly
,
there
is
no
soft
cover
edition
.

Word	POS
This	DT
book	NN
is	VBZ
the	DT
best	JJS
written	VBN
documentary	NN
thus	RB
far	RB
,	,
yet	RB
sadly	RB
,	,
there	EX
is	VBZ
no	DT
soft	JJ
cover	NN
edition	NN
.	.

Source: Zhang, Z., Li, X., and Chen, Y. (2012), "Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews," ACM Trans. Manage. Inf. Syst. (3:1) 2012, pp 1-23.,

Conversion of text representation

Word Vector (WV)	Polarity Score Vector (PSV) *pscore*	*nscore*		Microstate Sequence (MS)
This	0	0	Neutral (0)	0
book	0	0	Neutral (0)	0
is	0	0	Neutral (0)	0
the	0	0	Neutral (0)	0
best	0.75	0	Positive (0.75)	1
written	0	0	Neutral (0)	0
documentary	0	0	Neutral (0)	0
thus	0.375	0	Positive (0.375)	1
far	0.375	0	Positive (0.375)	1
,				
yet	0	0.125	Negative (0.125)	-1
sadly	0.25	0.5	Negative (0.25)	-1
,				
there	0	0	Neutral (0)	0
is	0	0	Neutral (0)	0
no	0	0.75	Negative (0.75)	-1
soft	0	0	Neutral (0)	0
cover	0	0	Neutral (0)	0
edition	0	0	Neutral (0)	0
.				

SentiWordNet Lookup → Microstate Mapping → Probability Mapping →

Probability Distribution (P)

P("1")=3/17

P("-1")=3/17

P("0")=11/17

Source: Zhang, Z., Li, X., and Chen, Y. (2012), "Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews," ACM Trans. Manage. Inf. Syst. (3:1) 2012, pp 1-23.,

Example of SentiWordNet

POS ID PosScore NegScore SynsetTerms Gloss

a 00217728 0.75 0 beautiful#1 delighting the senses or exciting intellectual or emotional admiration; "a beautiful child"; "beautiful country"; "a beautiful painting"; "a beautiful theory"; "a beautiful party“

a 00227507 0.75 0 best#1 (superlative of `good') having the most positive qualities; "the best film of the year"; "the best solution"; "the best time for planting"; "wore his best suit“

r 00042614 0 0.625 unhappily#2 sadly#1 in an unfortunate way; "sadly he died before he could see his grandchild“

r 00093270 0 0.875 woefully#1 sadly#3 lamentably#1 deplorably#1 in an unfortunate or deplorable manner; "he was sadly neglected"; "it was woefully inadequate“

r 00404501 0 0.25 sadly#2 with sadness; in a sad manner; "`She died last night,' he said sadly"

SenticNet

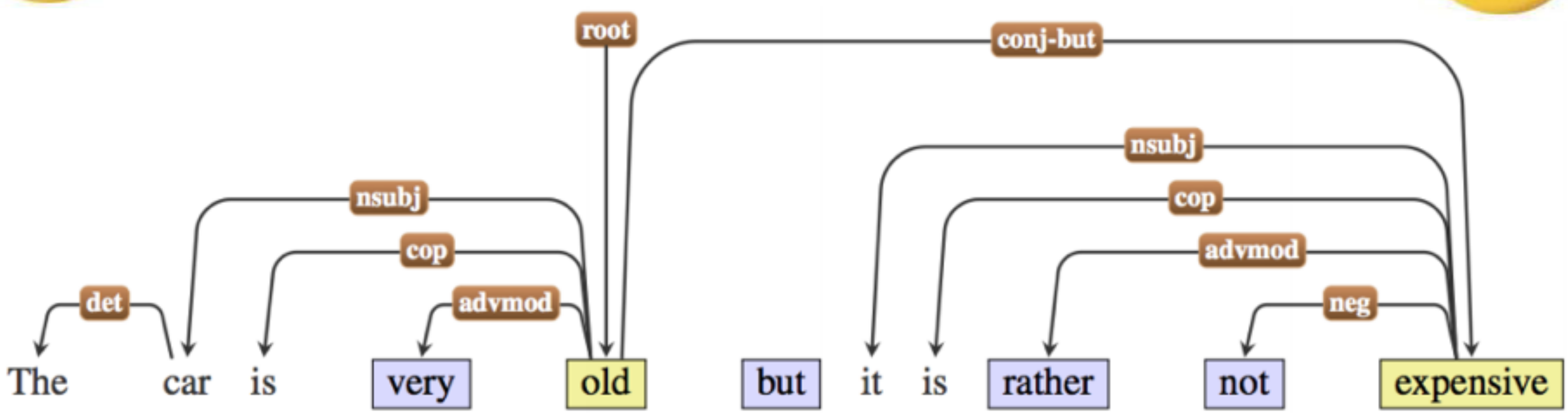
The car is very old but it is rather not expensive.

The car is very old but it is rather not expensive.

The car is very old but it is rather not expensive.

Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Polarity Detection with SenticNet

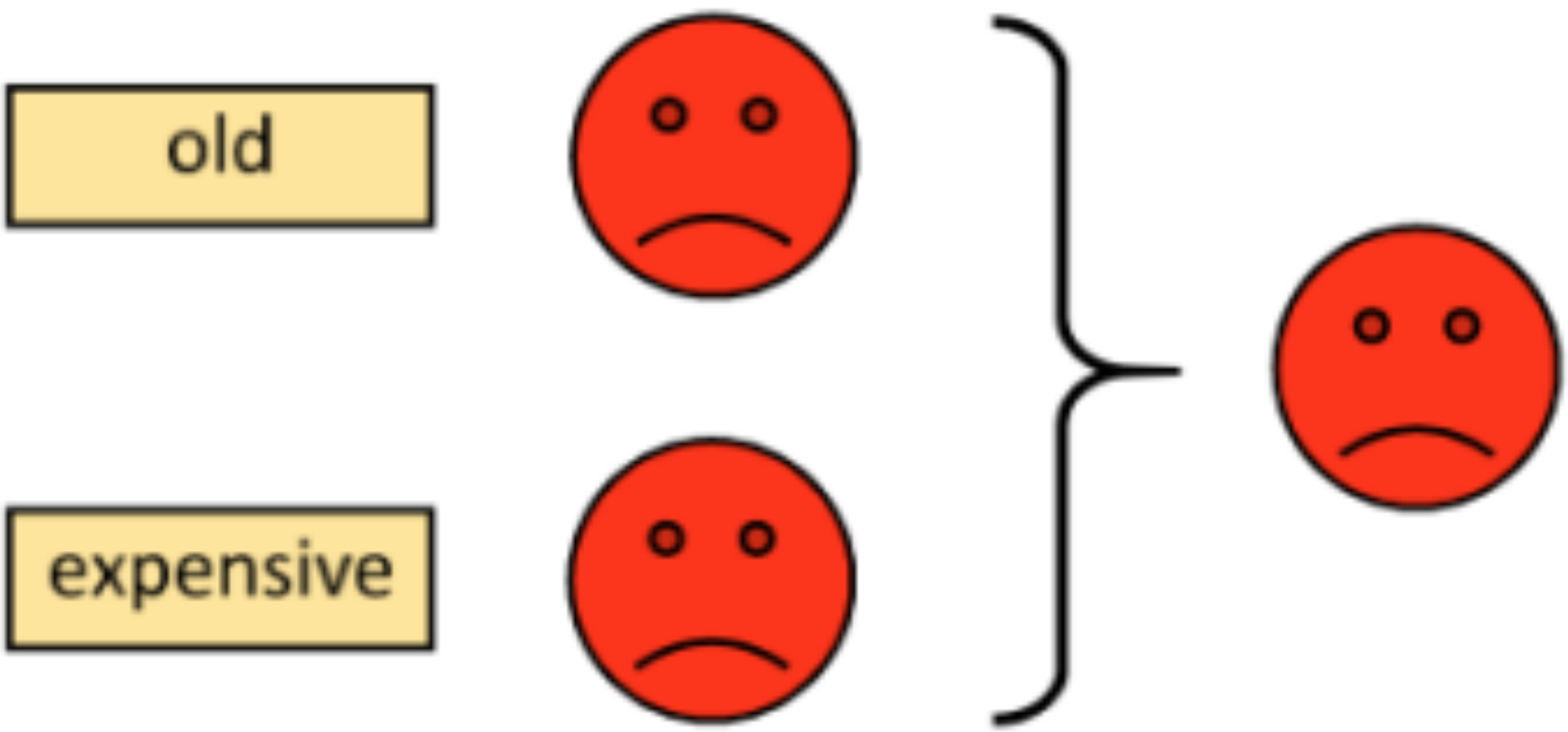


The car is very old but it is rather not expensive.

The car is very old but it is rather not expensive.

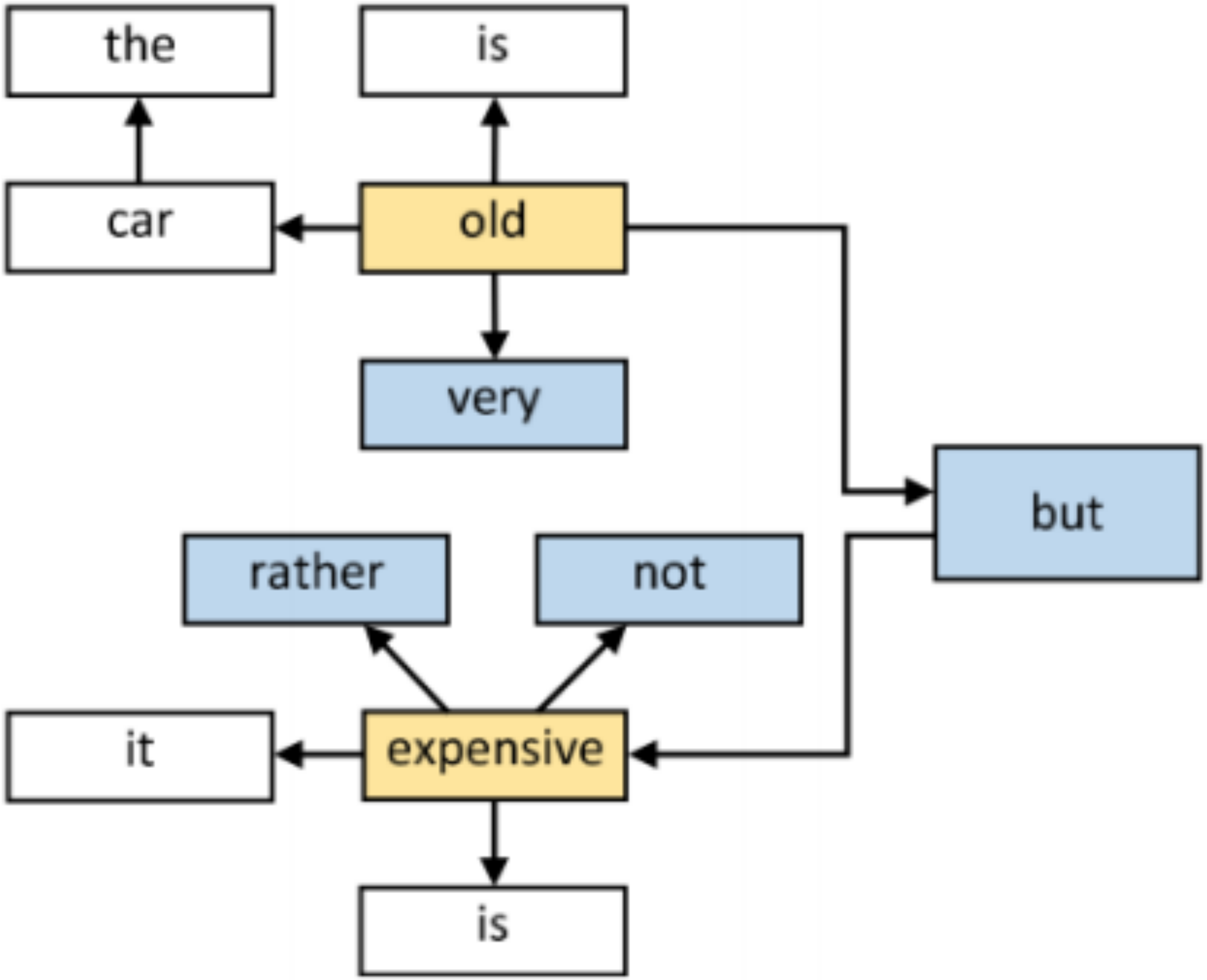
Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Polarity Detection with SenticNet



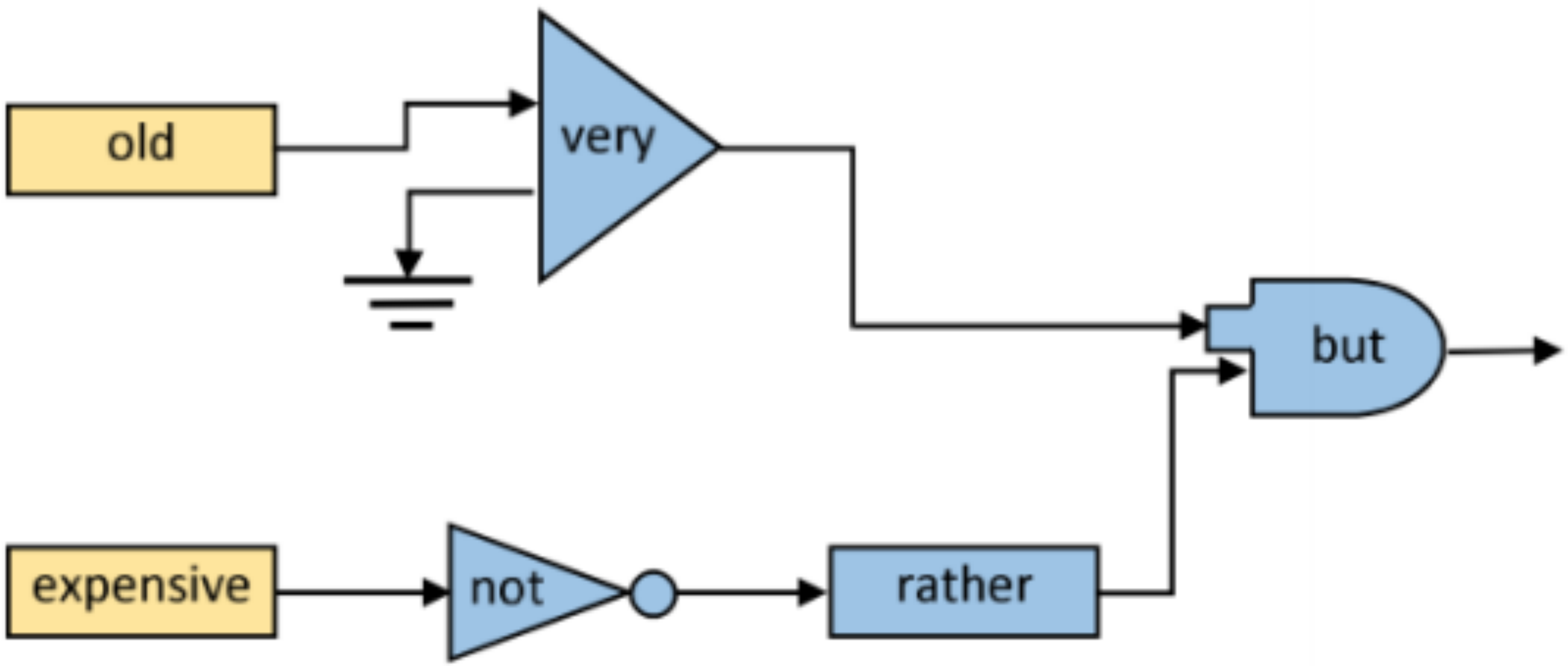
Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Polarity Detection with SenticNet



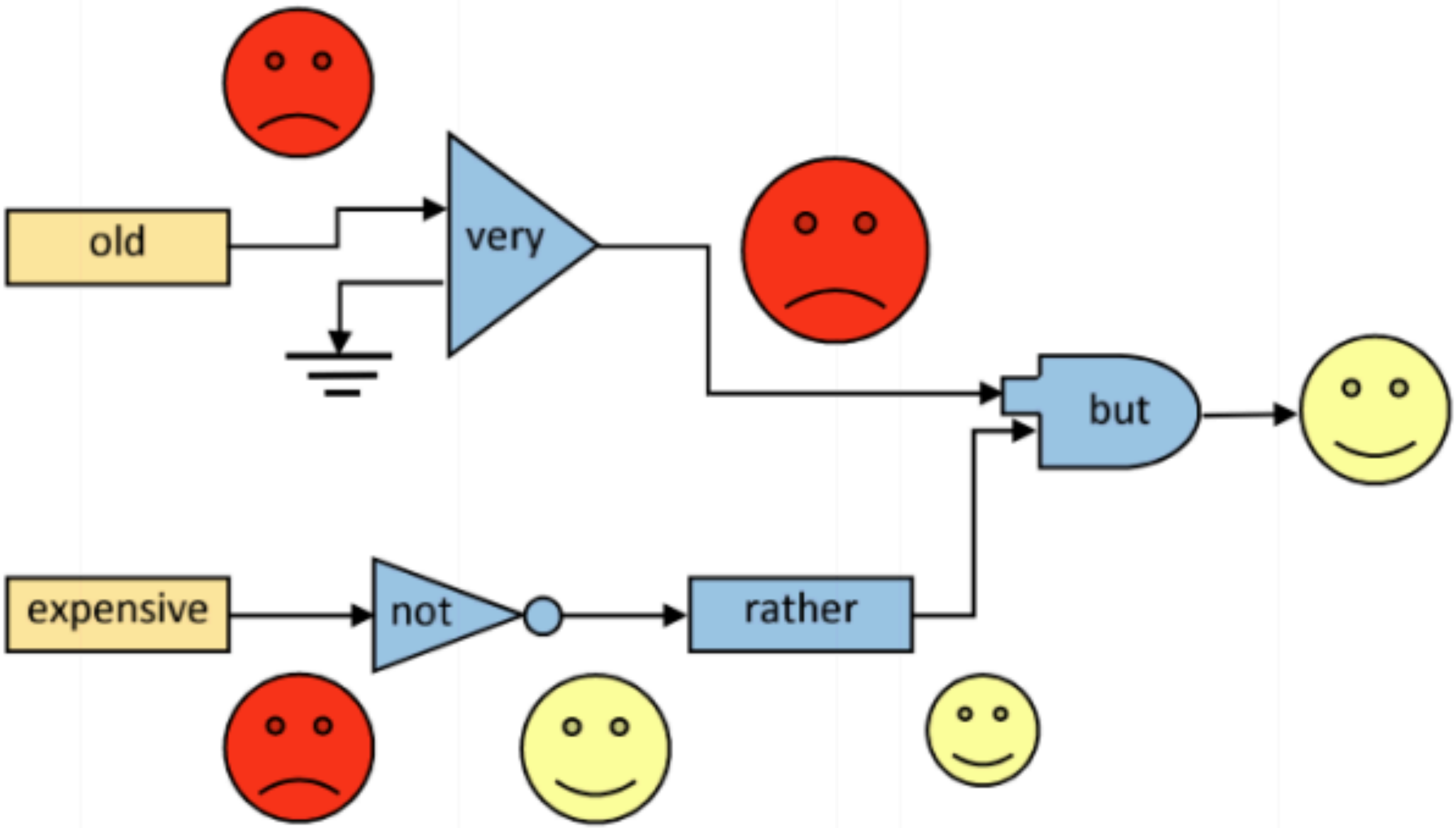
Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Polarity Detection with SenticNet



Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Polarity Detection with SenticNet



Source: Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives." In *the 26th International Conference on Computational Linguistics (COLING), Osaka*. 2016.

Evaluation

(Accuracy of Classification Model)

Evaluation of Text Mining and Sentiment Analysis

- Evaluation of Information Retrieval
- Evaluation of Classification Model (Prediction)
  - Accuracy
  - Precision
  - Recall
  - F-score

Assessment Methods for Classification

- Predictive accuracy
  - Hit rate
- Speed
  - Model building; predicting
- Robustness
- Scalability
- Interpretability
  - Transparency, explainability

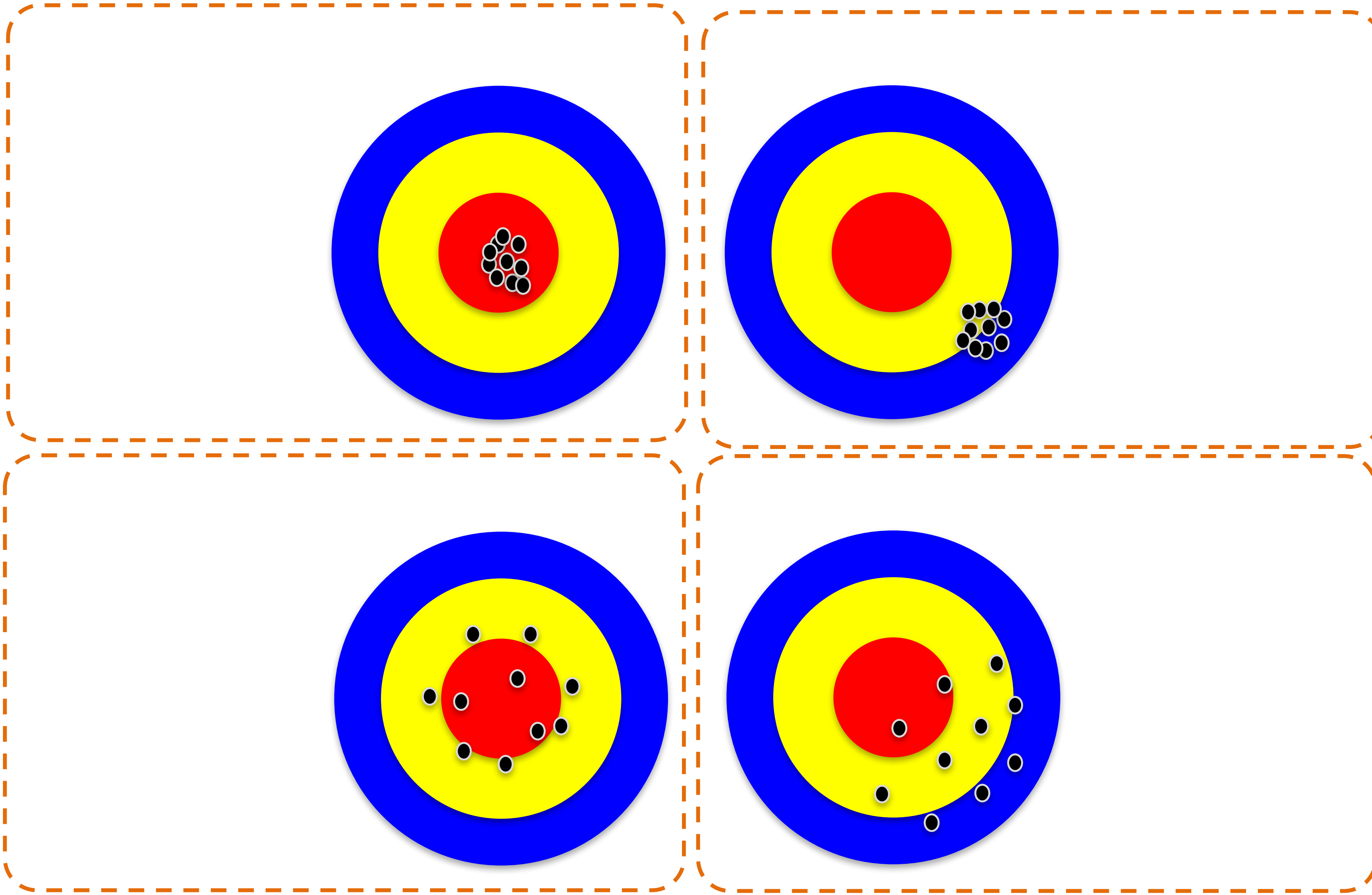
Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Accuracy

Validity

Precision

Reliability



Accuracy vs. Precision

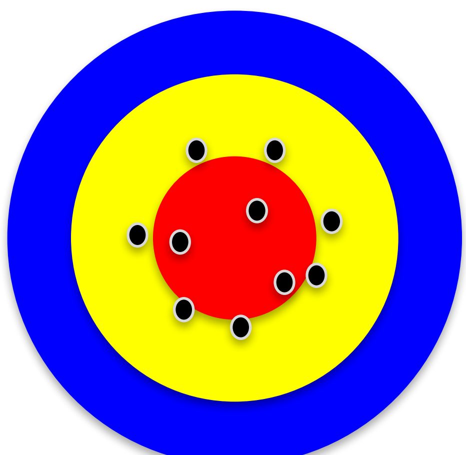
A

High Accuracy
High Precision

B

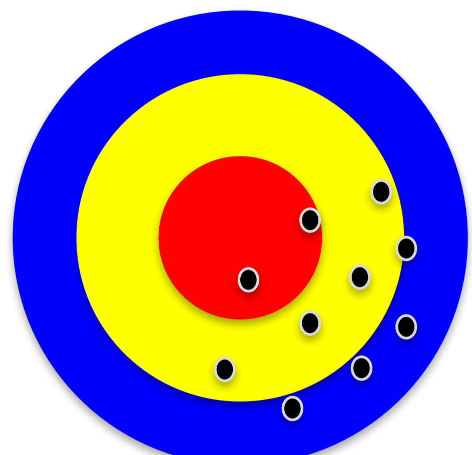
Low Accuracy
High Precision

C



High Accuracy
Low Precision

D



Low Accuracy
Low Precision

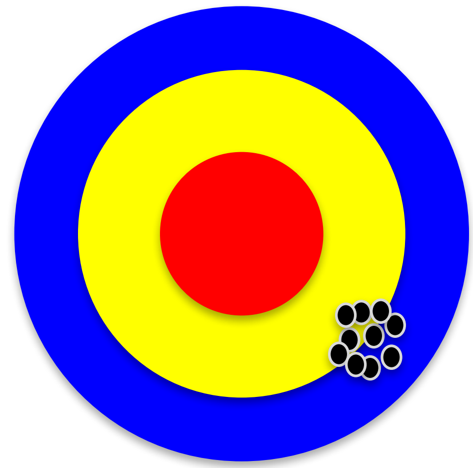
Accuracy vs. Precision

A

High Accuracy
High Precision

High Validity
High Reliability

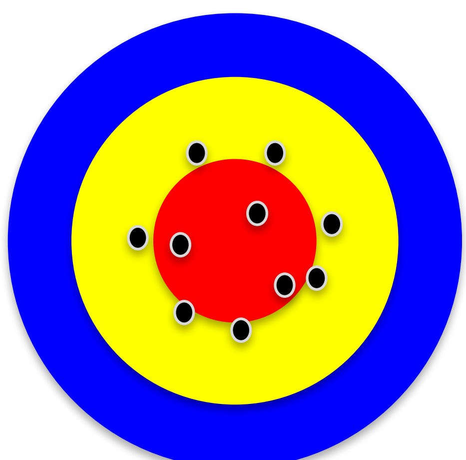
B



Low Accuracy
High Precision

Low Validity
High Reliability

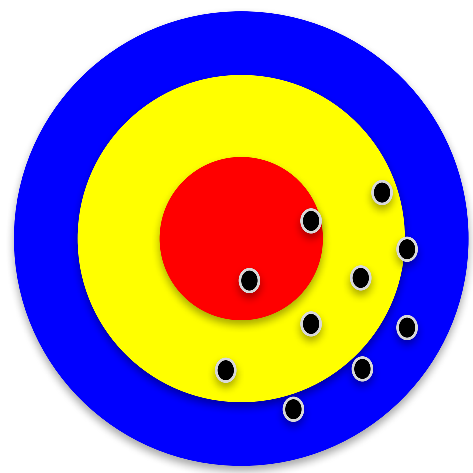
C



High Accuracy
Low Precision

High Validity
Low Reliability

D



Low Accuracy
Low Precision

Low Validity
Low Reliability

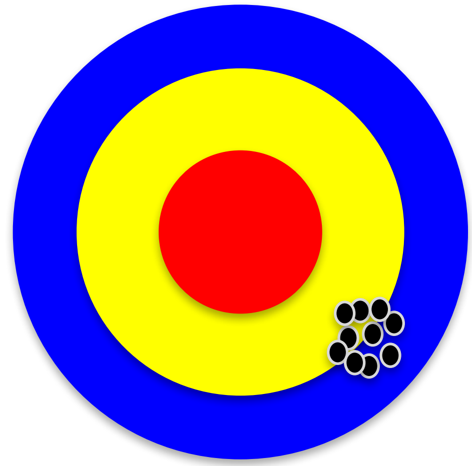
Accuracy vs. Precision

A

High Accuracy
High Precision

High Validity
High Reliability

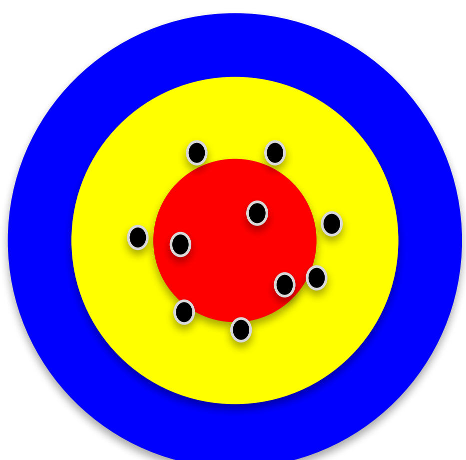
B



Low Accuracy
High Precision

Low Validity
High Reliability

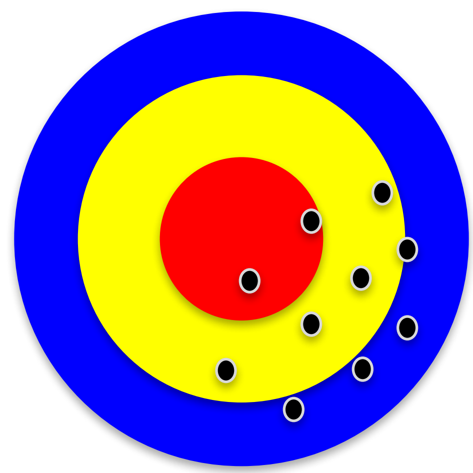
C



High Accuracy
Low Precision

High Validity
Low Reliability

D



Low Accuracy
Low Precision

Low Validity
Low Reliability

Accuracy of Classification Models

- In classification problems, the primary source for accuracy estimation is the confusion matrix

		True Class: Positive	True Class: Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
Predicted Class	Negative	False Negative Count (FN)	True Negative Count (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

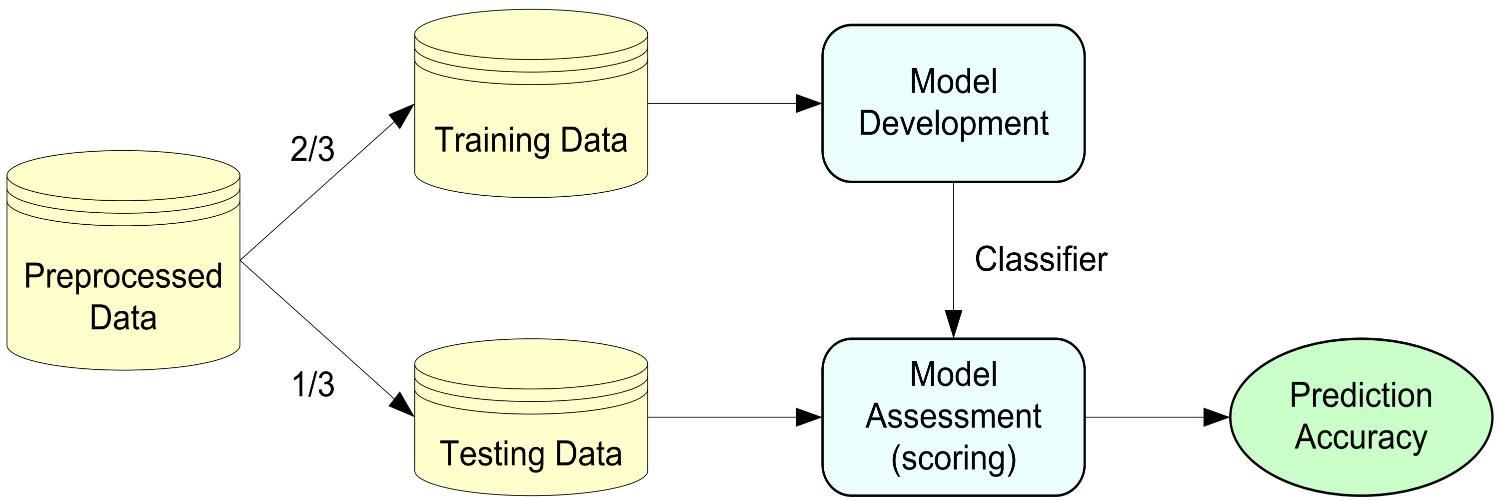
$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Estimation Methodologies for Classification

- Simple split (or holdout or test sample estimation)
  - Split the data into 2 mutually exclusive sets training (~70%) and testing (30%)



  - For ANN, the data is split into three sub-sets (training [~60%], validation [~20%], testing [~20%])

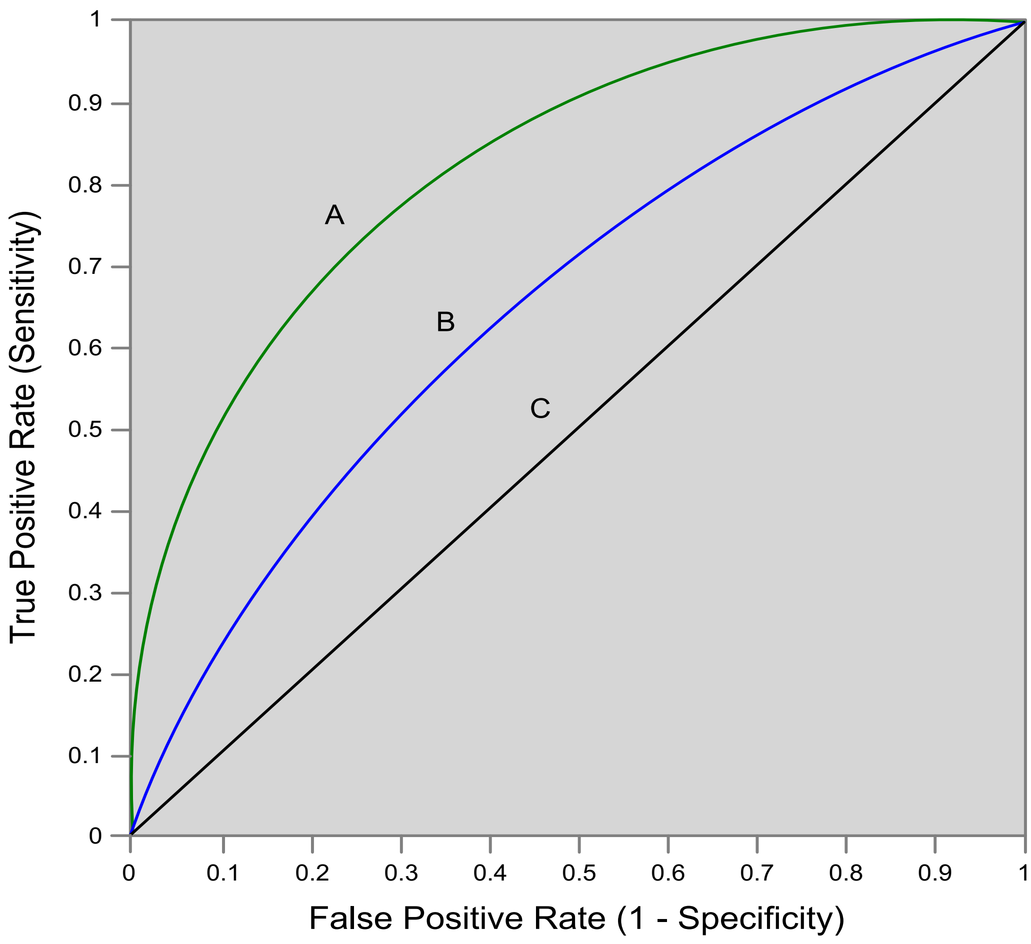
Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Estimation Methodologies for Classification

- *k*-Fold Cross Validation (rotation estimation)
  - Split the data into *k* mutually exclusive subsets
  - Use each subset as testing while using the rest of the subsets as training
  - Repeat the experimentation for *k* times
  - Aggregate the test results for true estimation of prediction accuracy training
- Other estimation methodologies
  - Leave-one-out, bootstrapping, jackknifing
  - Area under the ROC curve

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Estimation Methodologies for Classification – ROC Curve



Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Sensitivity =True Positive Rate

Specificity =True Negative Rate

		True Class (actual value)		
		Positive	Negative	total
Predictive Class (prediction outcome)	Positive	True Positive (TP)	False Positive (FP)	P'
	Negative	False Negative (FN)	True Negative (TN)	N'
total		P	N	

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

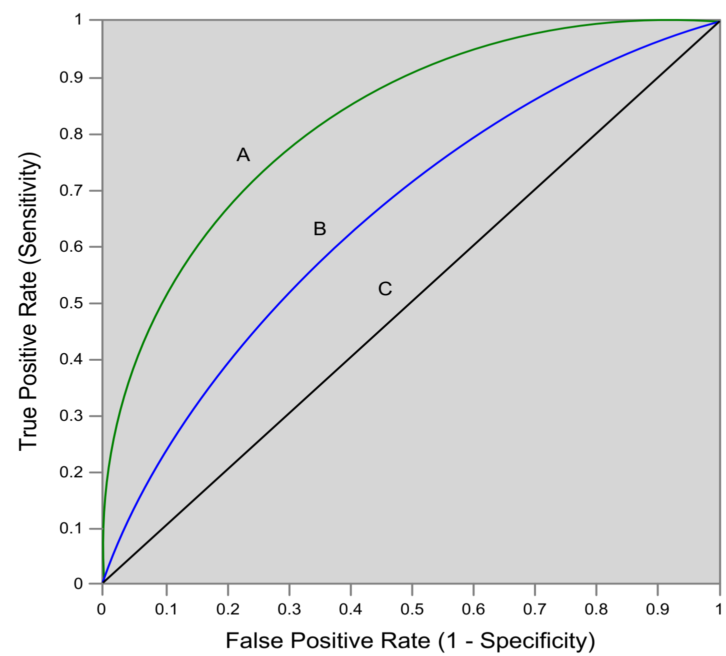
$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$True\ Positive\ Rate\ (\text{Sensitivity}) = \frac{TP}{TP + FN}$$

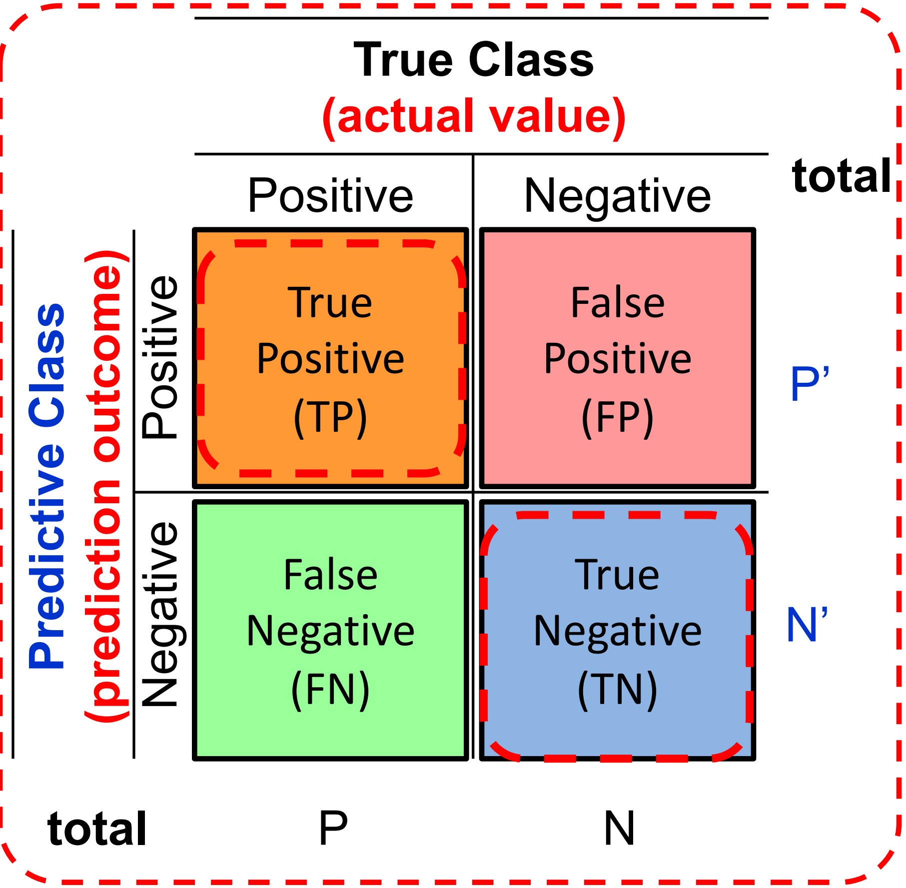
$$True\ Negative\ Rate\ (\text{Specificity}) = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN}$$

$$False\ Positive\ Rate\ (1\text{-Specificity}) = \frac{FP}{FP + TN}$$

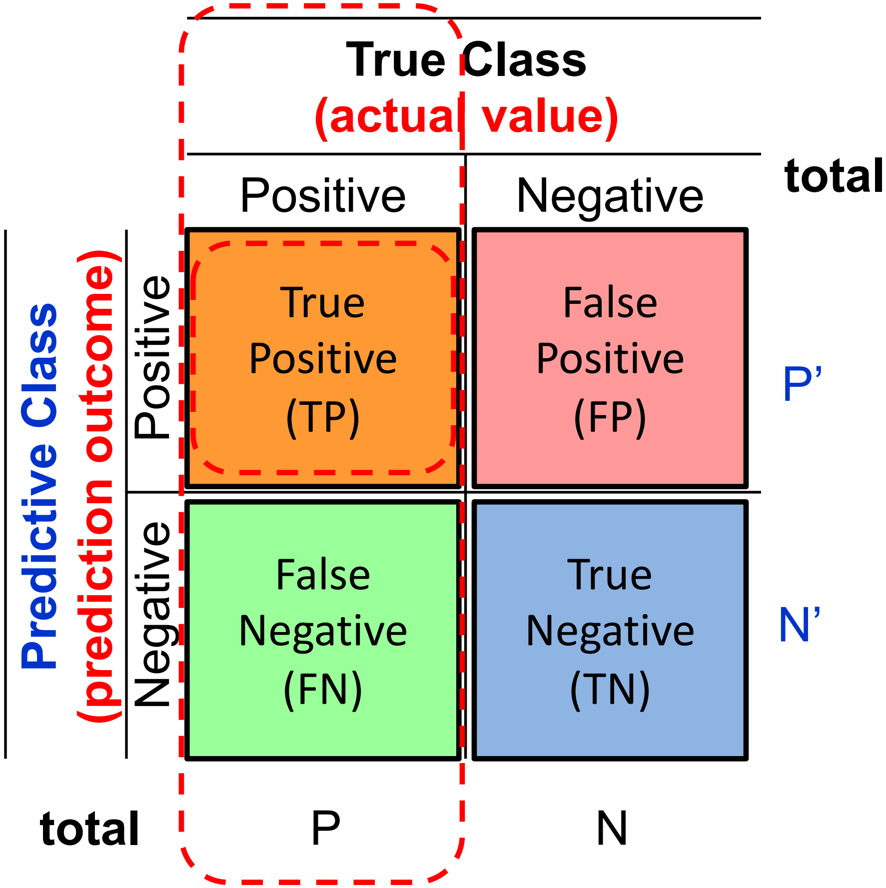


Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy (ACC)
= TP + TN / (TP + TN + FP + FN)

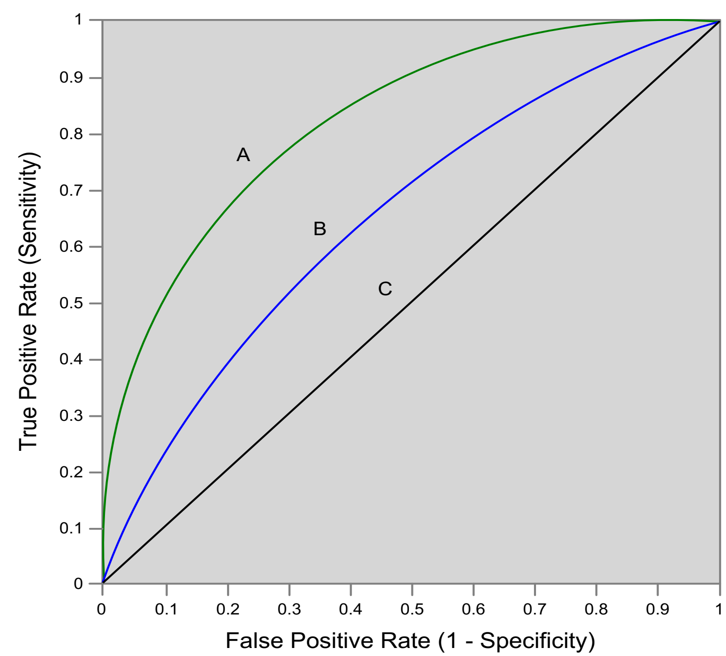


$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

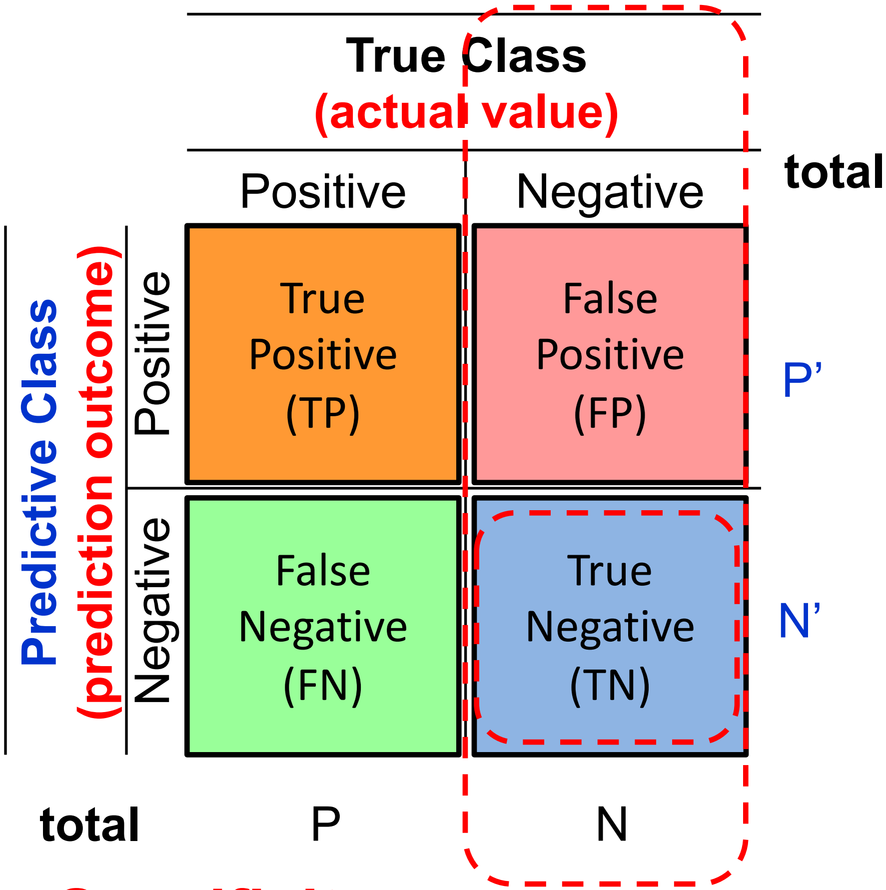
$$Recall = \frac{TP}{TP + FN}$$

$$True\ Positive\ Rate\ (\text{Sensitivity}) = \frac{TP}{TP + FN}$$

Sensitivity
= True Positive Rate
= Recall
= Hit rate
= TP / (TP + FN)



Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic



Specificity

= True Negative Rate

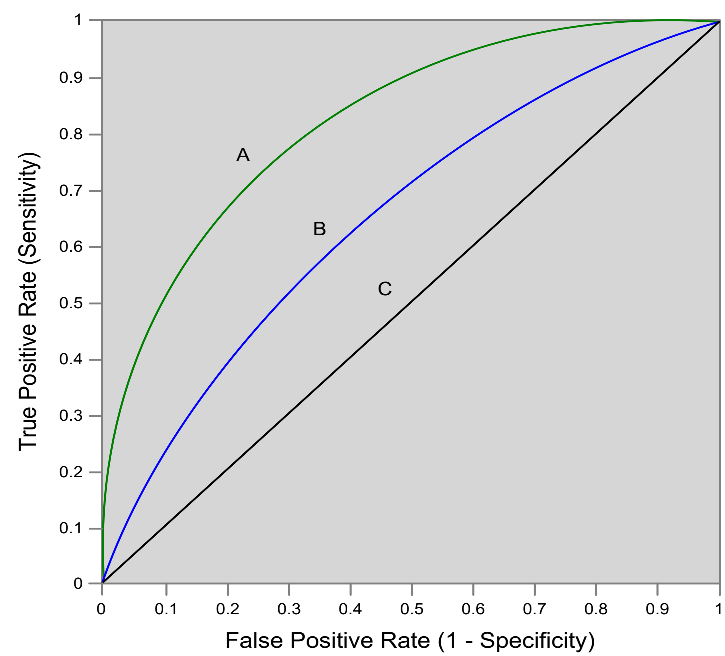
= TN / N

= TN / (TN+ FP)

$$True\ Negative\ Rate\ (\text{Specificity}) = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate\ (1\text{-Specificity}) = \frac{FP}{FP + TN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$



Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

		True Class (actual value)		
		Positive	Negative	total
Predictive Class (prediction outcome)	Positive	True Positive (TP)	False Positive (FP)	P’
	Negative	False Negative (FN)	True Negative (TN)	N’
total		P	N	

Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

$$Recall = \frac{TP}{TP + FN}$$

F1 score (F-score)(F-measure)
is the harmonic mean of precision and recall
= 2TP / (P + P’)
= 2TP / (2TP + FP + FN)

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

A

63 (TP)	28 (FP)	91
37 (FN)	72 (TN)	109
100	100	200

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate
= TP / (TP + FN)

Specificity
= True Negative Rate
= TN / N
= TN / (TN + FP)

TPR = 0.63

$$Recall = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate\ (\text{Specificity}) = \frac{TN}{TN + FP}$$

FPR = 0.28

$$\text{False}\ Positive\ Rate\ (1\text{-Specificity}) = \frac{FP}{FP + TN}$$

PPV = 0.69
=63/(63+28)
=63/91

$$Precision = \frac{TP}{TP + FP}$$

Precision
= Positive Predictive Value (PPV)

F1 = 0.66
= 2*(0.63*0.69)/(0.63+0.69)
= (2 * 63) /(100 + 91)
= (0.63 + 0.69) / 2 =1.32 / 2 =0.66

$$F = 2 * \frac{precision * recall}{precision + recall}$$

F1 score (F-score)
(F-measure)
is the harmonic mean of precision and recall
= 2TP / (P + P')
= 2TP / (2TP + FP + FN)

ACC = 0.68
= (63 + 72) / 200
= 135/200 = 67.5

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

A

63 (TP)	28 (FP)	91
37 (FN)	72 (TN)	109
100	100	200

TPR = 0.63

FPR = 0.28

PPV = 0.69
=63/(63+28)
=63/91

F1 = 0.66
= 2*(0.63*0.69)/(0.63+0.69)
= (2 * 63) /(100 + 91)
= (0.63 + 0.69) / 2 =1.32 / 2 =0.66

ACC = 0.68
= (63 + 72) / 200
= 135/200 = 67.5

B

77 (TP)	77 (FP)	154
23 (FN)	23 (TN)	46
100	100	200

TPR = 0.77

FPR = 0.77

PPV = 0.50

F1 = 0.61

ACC = 0.50

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

$$Recall = \frac{TP}{TP + FN}$$

Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

C

24 (TP)	88 (FP)	112
76 (FN)	12 (TN)	88
100	100	200

TPR = 0.24
FPR = 0.88
PPV = 0.21
F1 = 0.22
ACC = 0.18

C'

76 (TP)	12 (FP)	88
24 (FN)	88 (TN)	112
100	100	200

TPR = 0.76
FPR = 0.12
PPV = 0.86
F1 = 0.81
ACC = 0.82

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

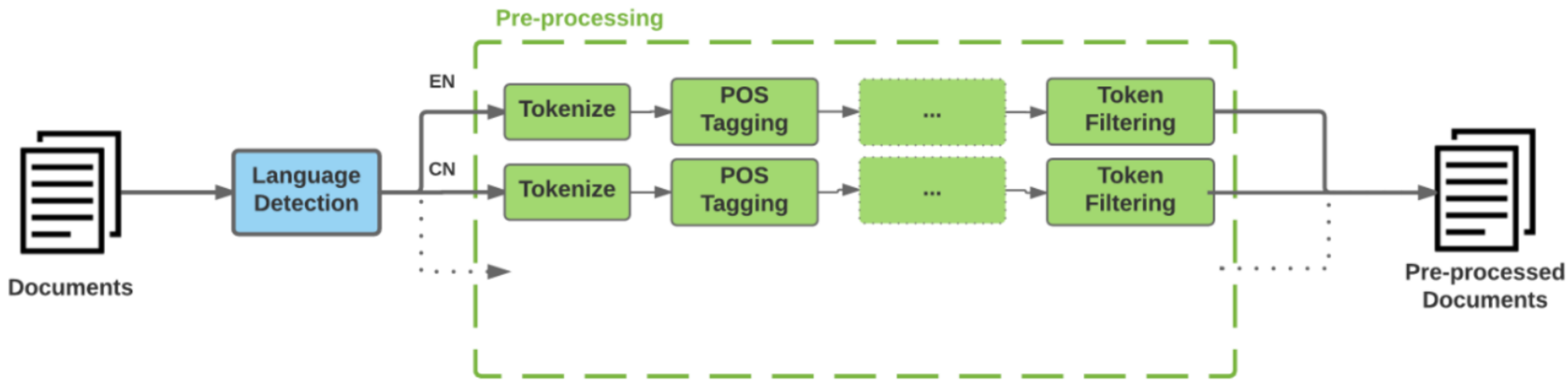
$$Recall = \frac{TP}{TP + FN}$$

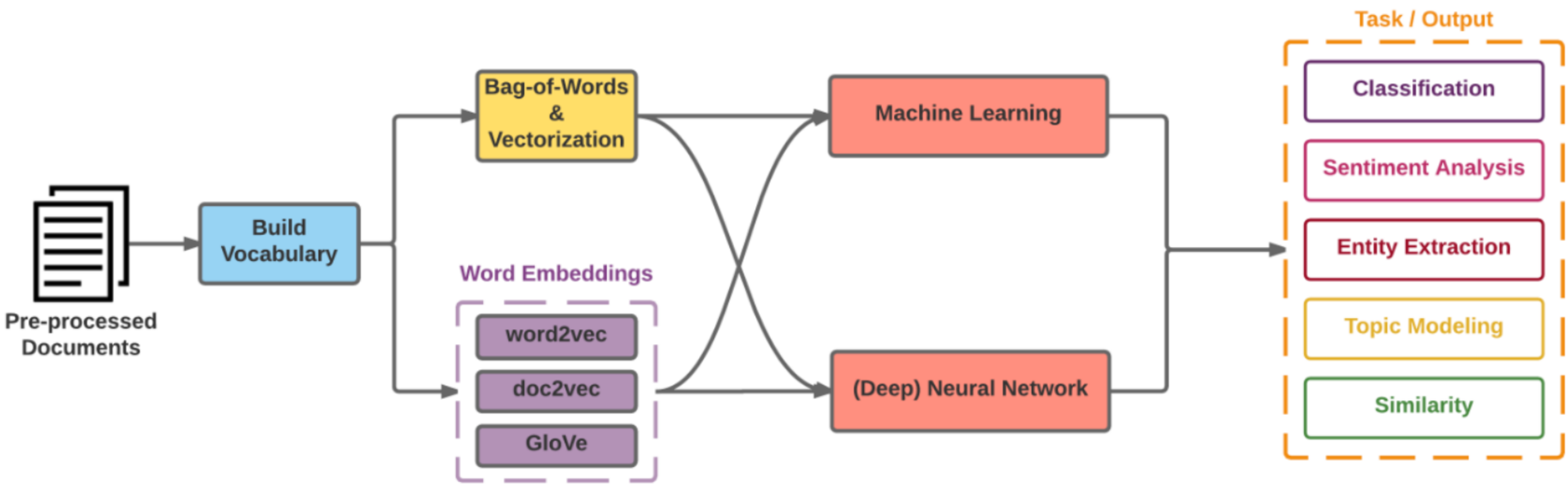
Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

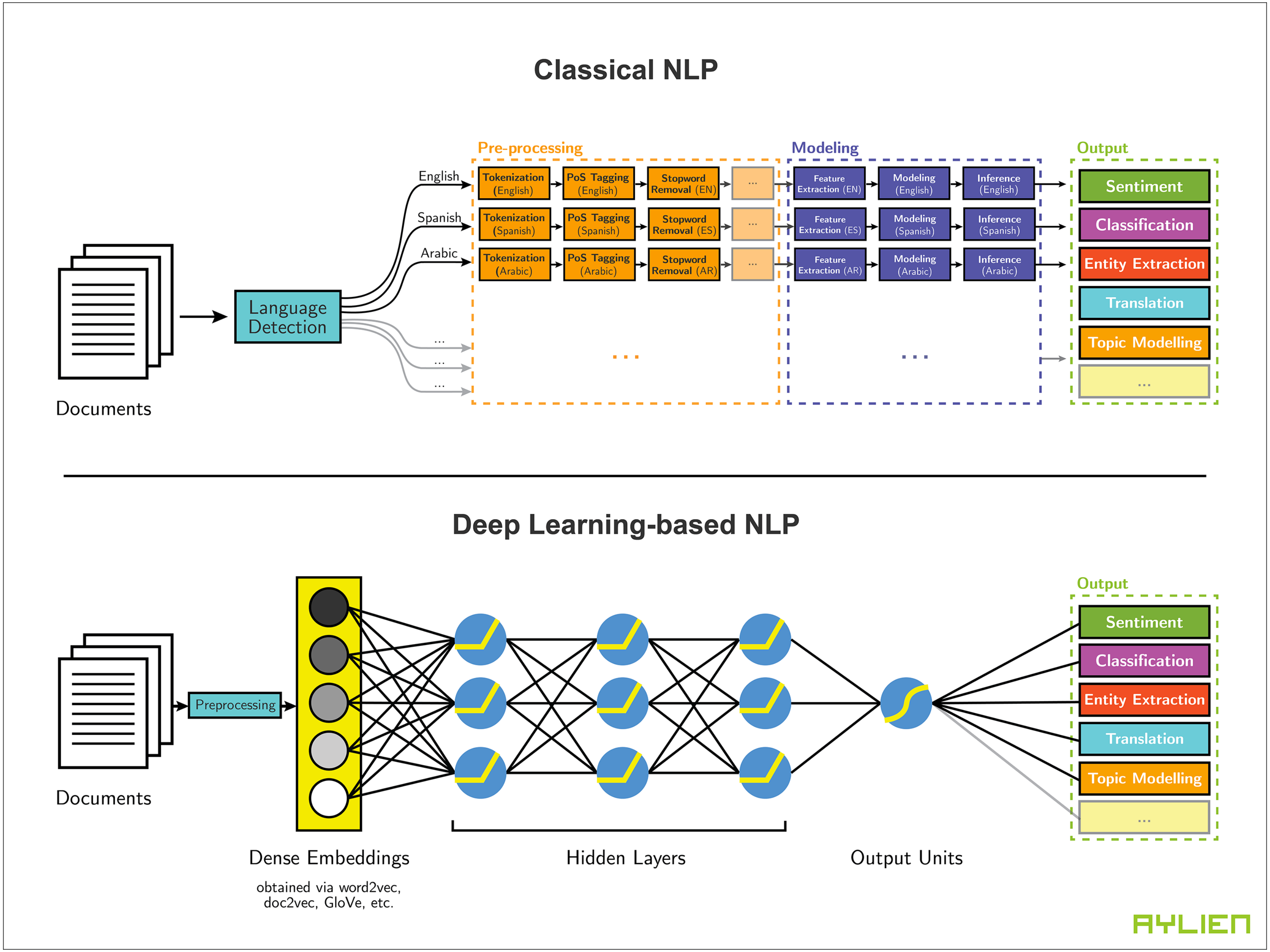
Modern NLP Pipeline





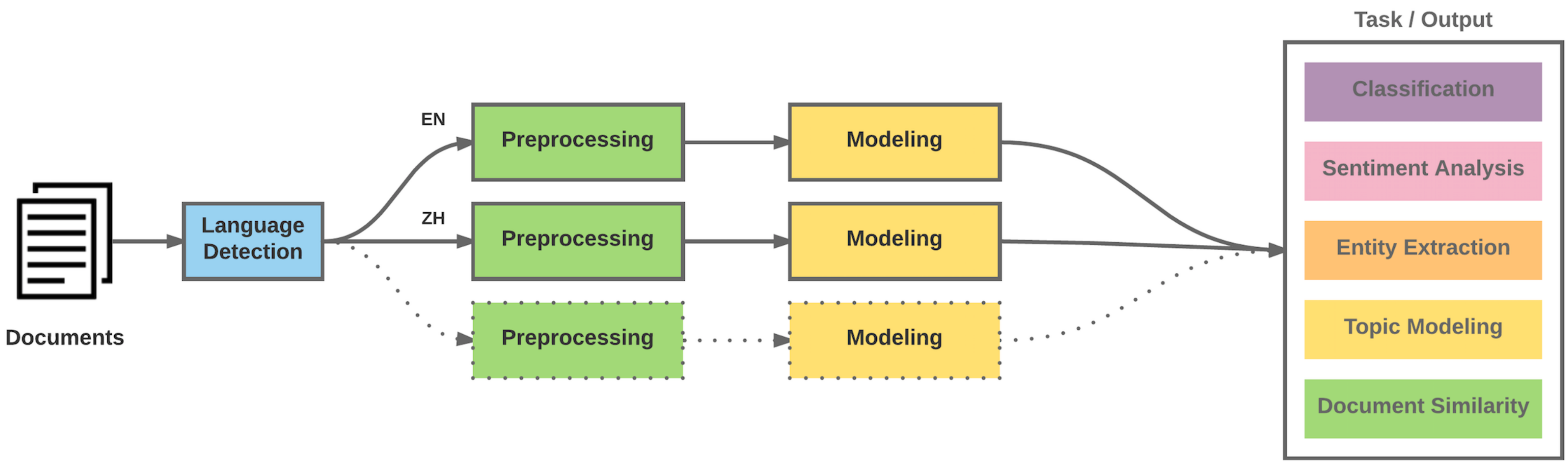
Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

NLP



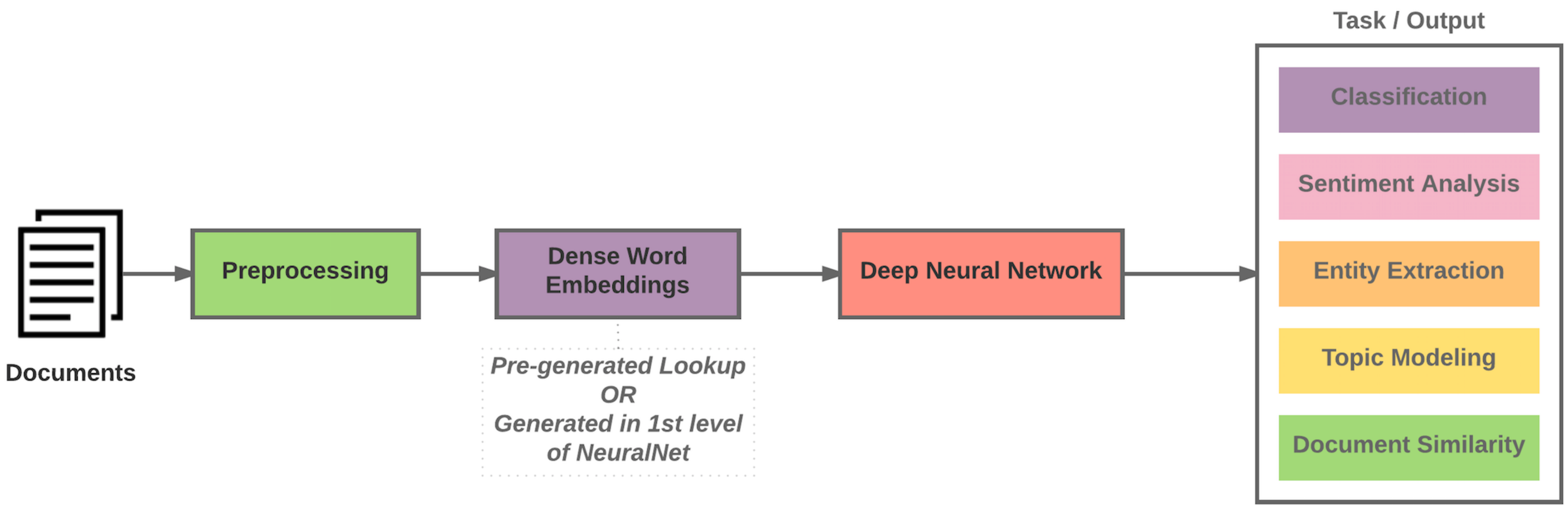
Source: http://blog.aylien.com/leveraging-deep-learning-for-multilingual/

Modern NLP Pipeline



Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

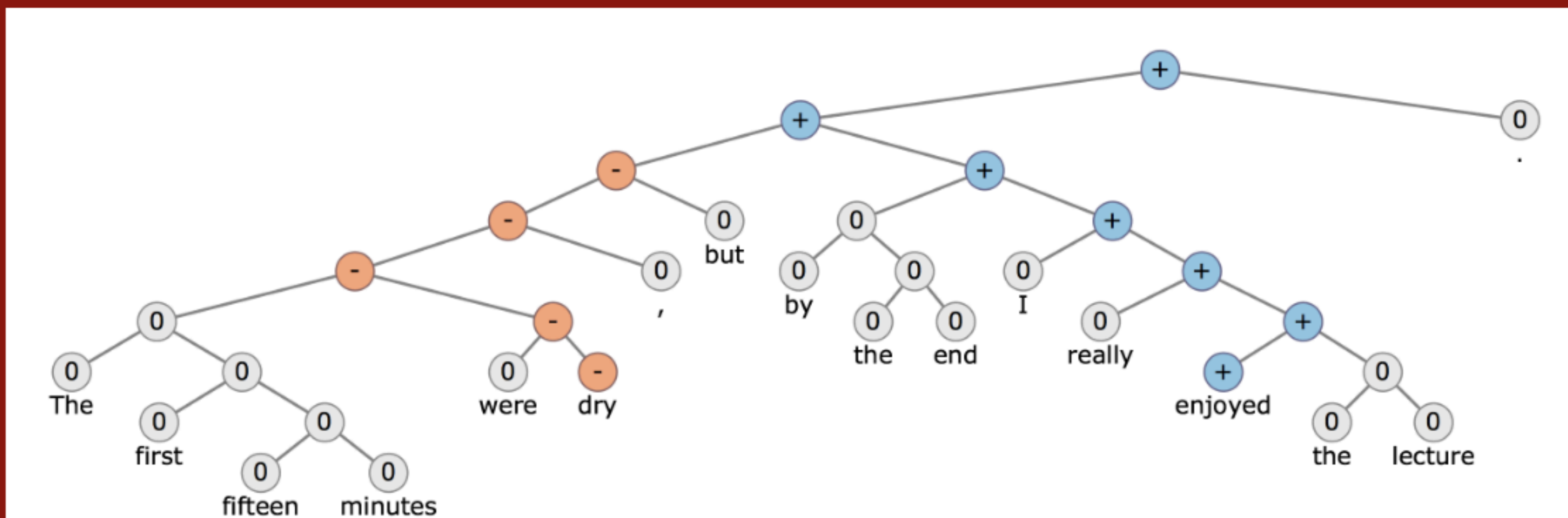
Deep Learning NLP



Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

CS224d: Deep Learning for Natural Language Processing

CS224d: Deep Learning for Natural Language Processing



Course Description

Natural language processing (NLP) is one of the most important technologies of the information age. Understanding complex language utterances is also a crucial part of artificial intelligence. Applications of NLP are everywhere because people communicate most everything in language: web search, advertisement, emails, customer service, language translation, radiology reports, etc. There are a large variety of underlying tasks and machine learning models powering NLP applications. Recently, deep learning approaches

http://cs224d.stanford.edu/

Deeply Moving: Deep Learning for Sentiment Analysis

Sentiment Analysis | **Information** | Live Demo | Sentiment Treebank | Help the Model | Source Code

Deeply Moving: Deep Learning for Sentiment Analysis

This website provides a live demo for predicting the sentiment of movie reviews. Most sentiment prediction systems work just by looking at words in isolation, giving positive points for positive words and negative points for negative words and then summing up these points. That way, the order of words is ignored and important information is lost. In constrast, our new deep learning model actually builds up a representation of whole sentences based on the sentence structure. It computes the sentiment based on how words compose the meaning of longer phrases. This way, the model is not as easily fooled as previous models. For example, our model learned that *funny* and *witty* are positive but the following sentence is still negative overall:

This movie was actually neither that funny, nor super witty.

The underlying technology of this demo is based on a new type of *Recursive Neural Network* that builds on top of grammatical structures. You can also browse the Stanford Sentiment Treebank, the dataset on which this model was trained. The model and dataset are described in an upcoming EMNLP paper. Of course, no model is perfect. You can help the model learn even more by labeling sentences we think would help the model or those you try in the live demo.

Paper Title and Abstract

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Paper: Download pdf

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng and Christopher Potts

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)

Dataset Downloads:

Main zip file with readme (6mb)
Dataset raw counts (5mb)
Train,Dev,Test Splits in PTB Tree Format

Code: Download Page

Press: Stanford Press Release

Dataset visualization and web design by Jason Chuang. Live demo by Jean Wu, Richard Socher, Rukmani Ravisundaram and Tayyab Tariq.

http://nlp.stanford.edu/sentiment/

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang,
Christopher D. Manning, Andrew Y. Ng and Christopher Potts
Stanford University, Stanford, CA 94305, USA
richard@socher.org,{aperelyg,jcchuang,ang}@cs.stanford.edu
{jeaneis,manning,cgpotts}@stanford.edu

Abstract

Semantic word spaces have been very useful but cannot express the meaning of longer phrases in a principled way. Further progress towards understanding compositionality in tasks such as sentiment detection requires richer supervised training and evaluation resources and more powerful models of composition. To remedy this, we introduce a Sentiment Treebank. It includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences and presents new challenges for sentiment compositionality. To address them, we introduce the Recursive Neural Tensor Network. When trained on the new treebank, this model out-

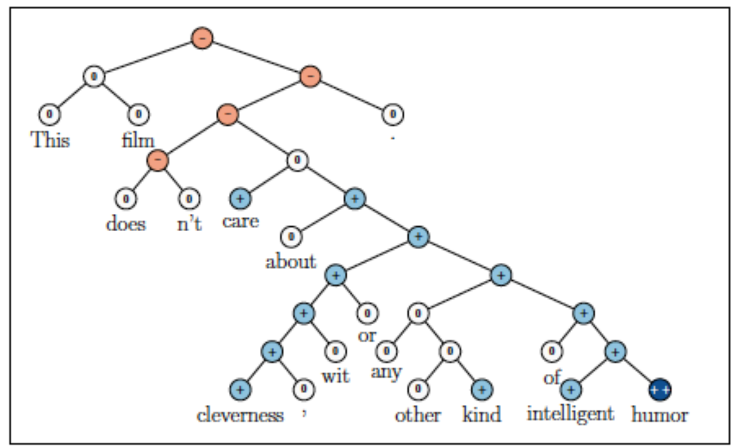
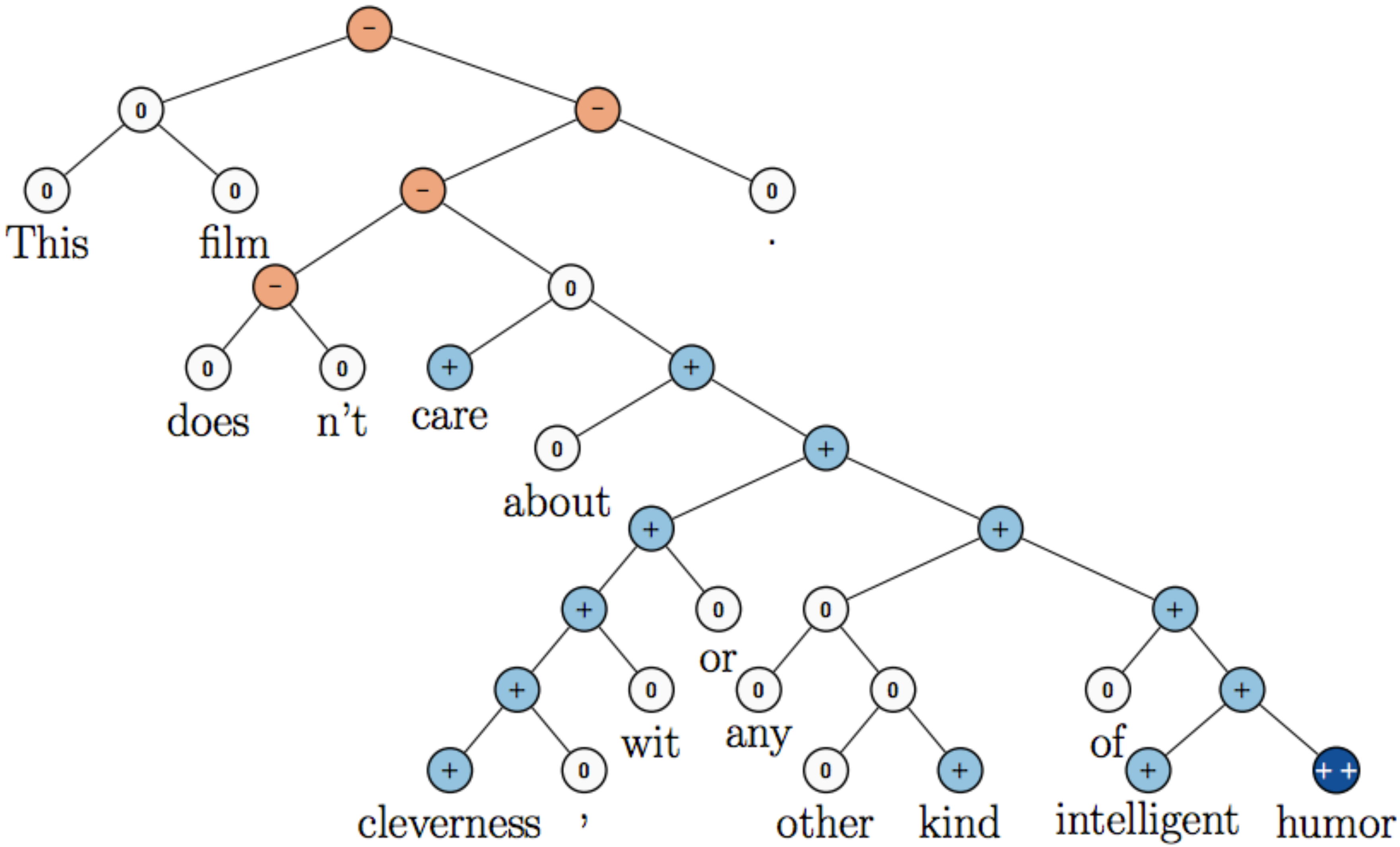


Figure 1: Example of the Recursive Neural Tensor Network accurately predicting 5 sentiment classes, very negative to very positive (– –, –, 0, +, + +), at every node of a parse tree and capturing the negation and its scope in this sentence.

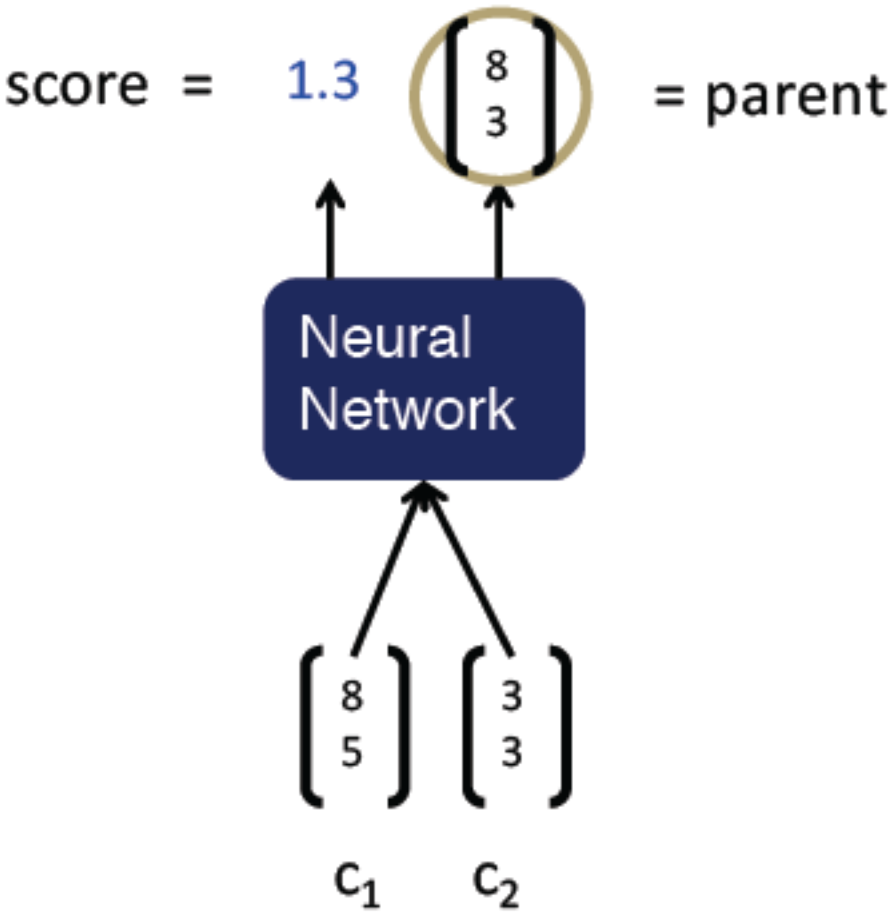
Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

Recursive Neural Tensor Network (RNTN)



Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

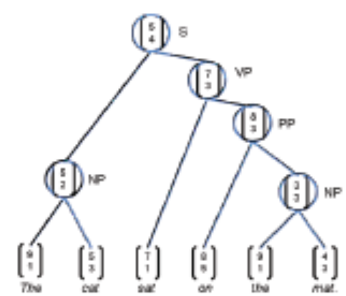
Recursive Neural Network Definition



score $= U^{\mathsf{T}} p$

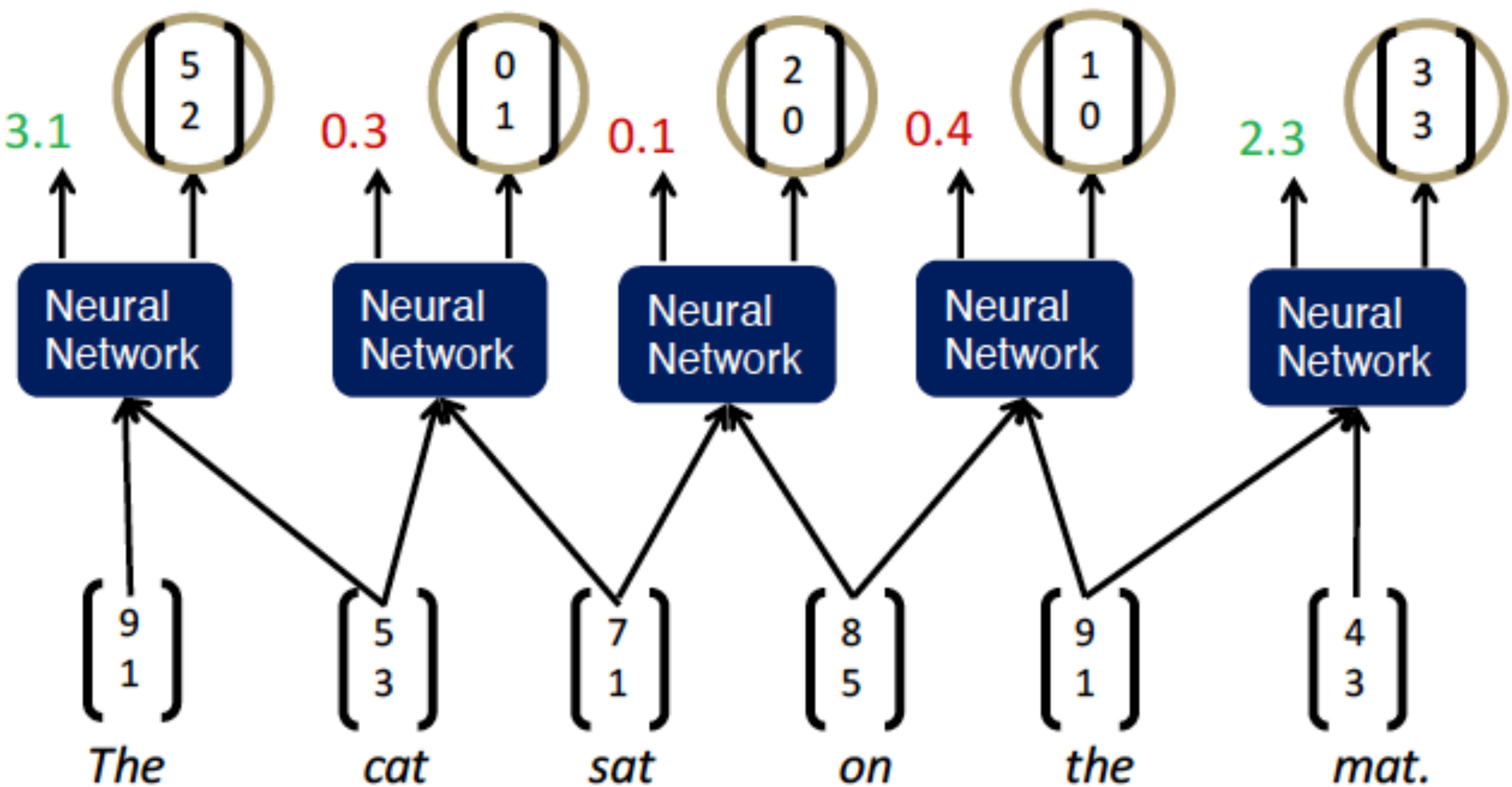
$$\boxed{p = \tanh\left(W\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),}$$

Same W parameters at all nodes of the tree



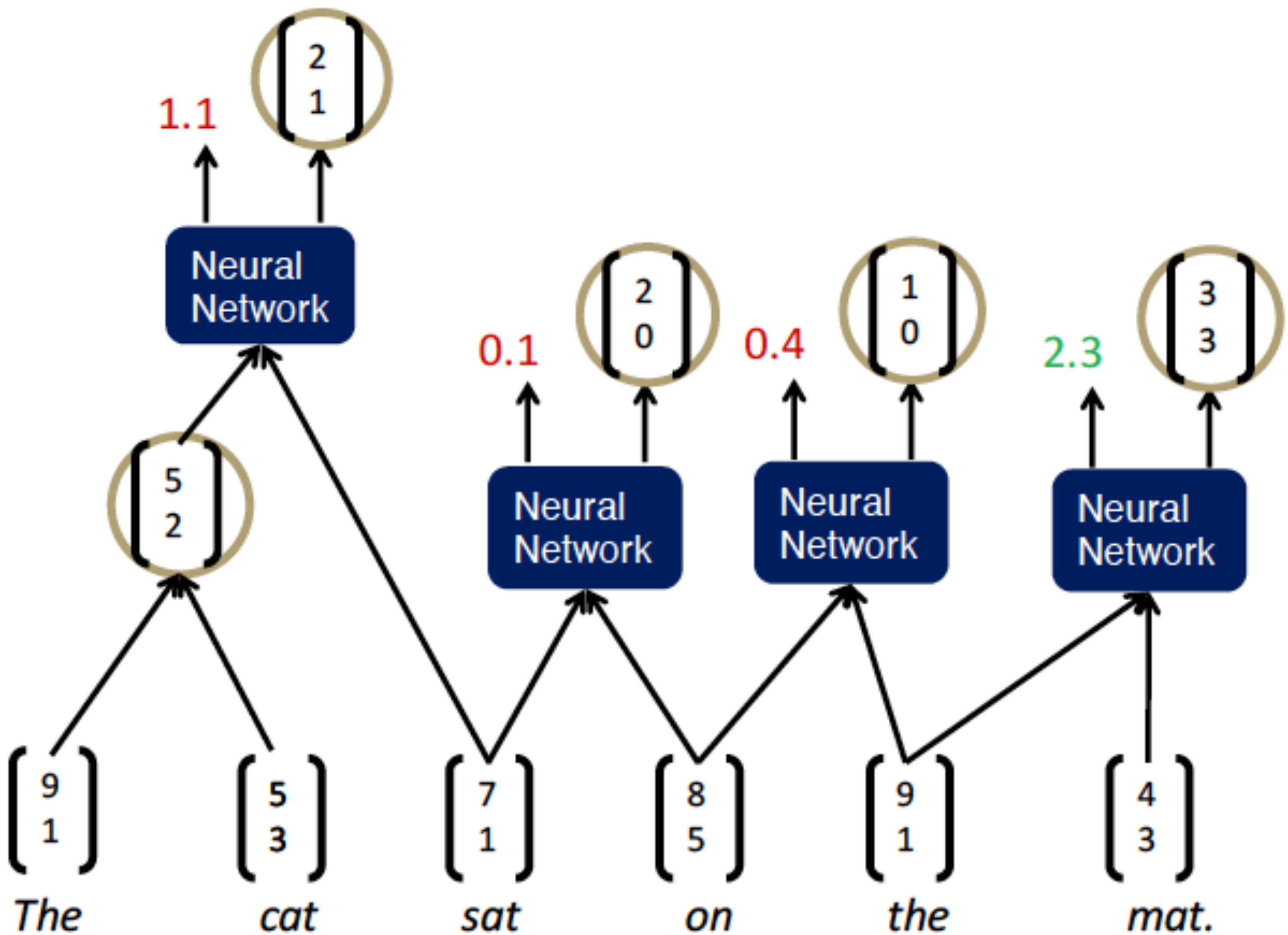
Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

Parsing a sentence with an RNN



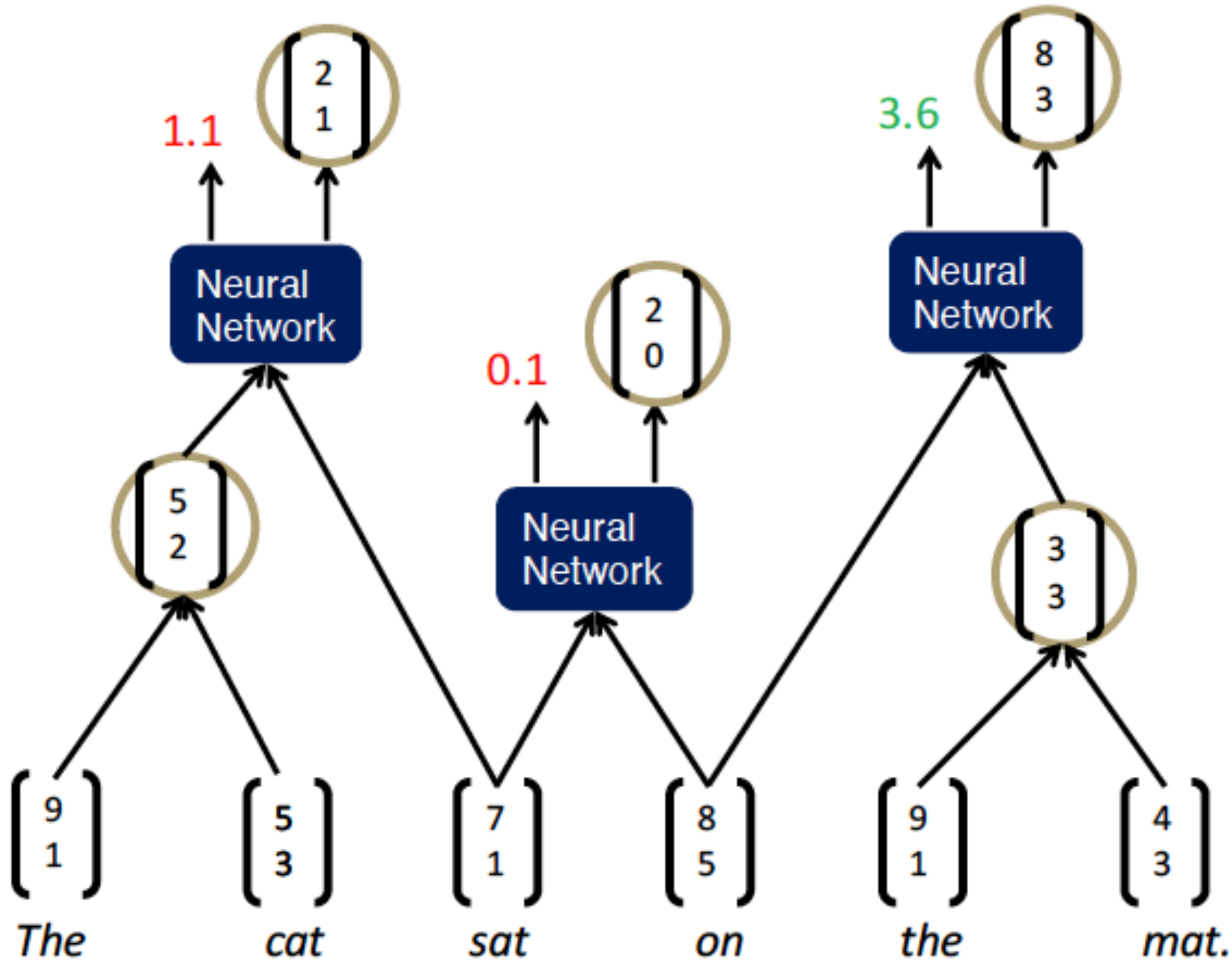
Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

Parsing a sentence with an RNN



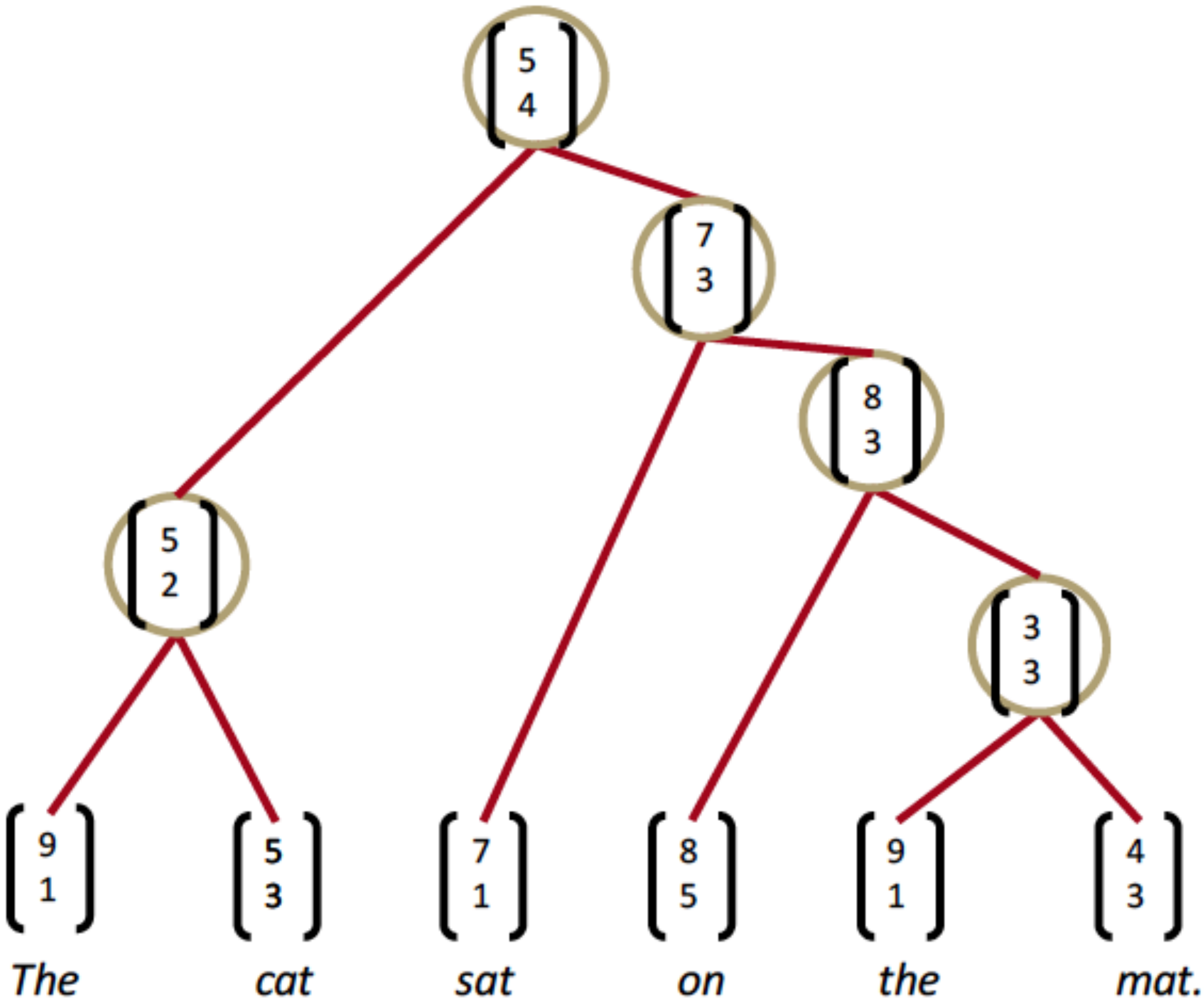
Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

Parsing a sentence with an RNN



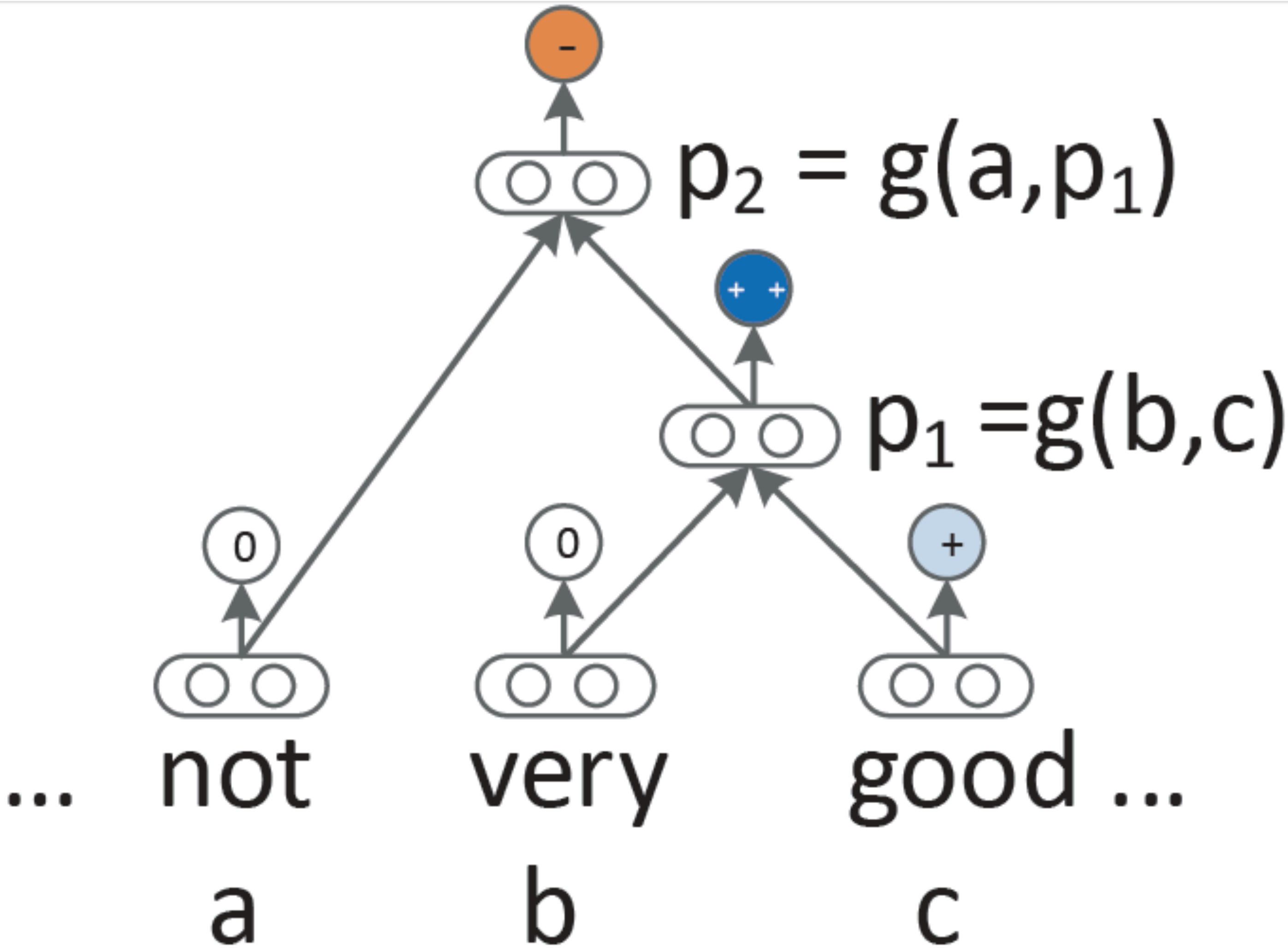
Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

Parsing a sentence with an RNN



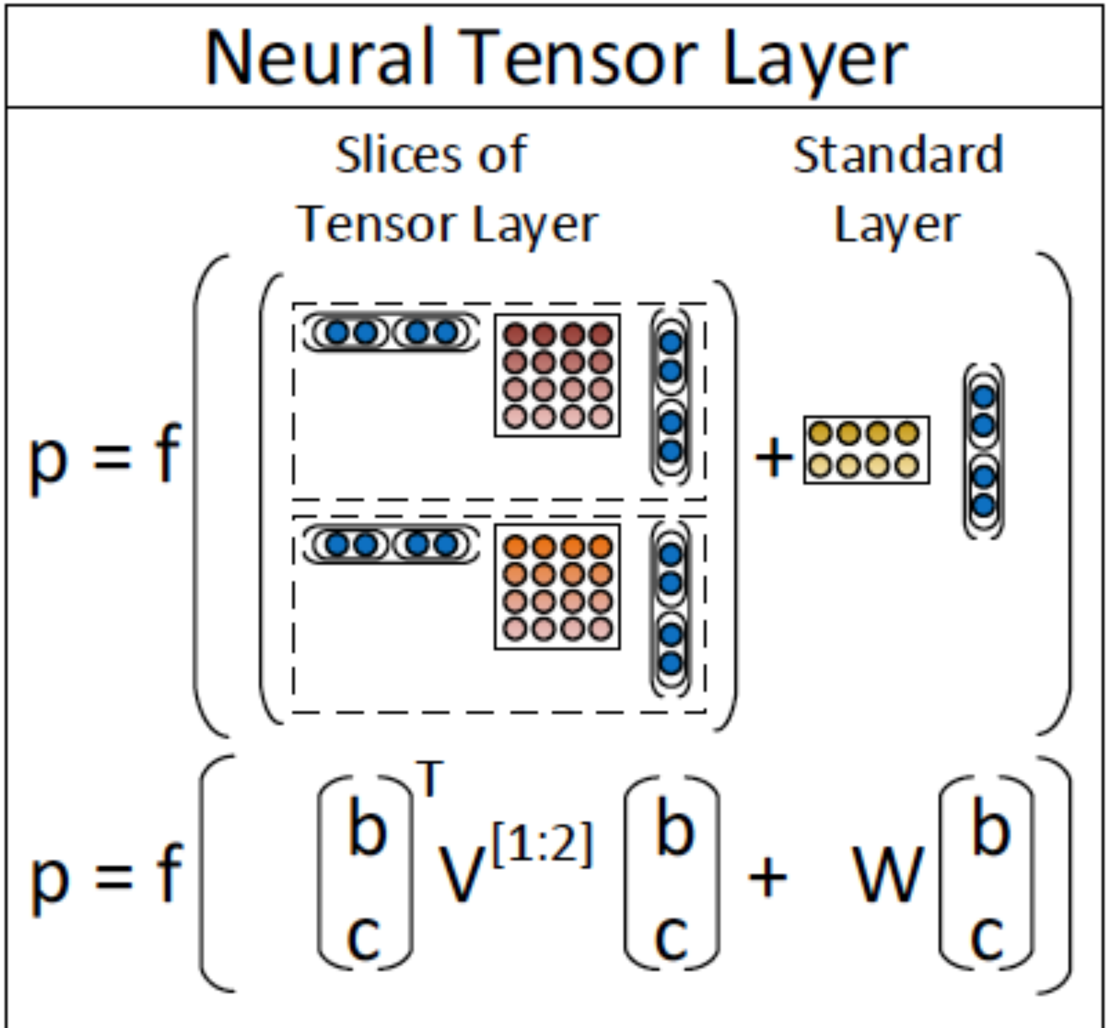
Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

Recursive Neural Network (RNN) models for sentiment



Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

Recursive Neural Tensor Network (RNTN)



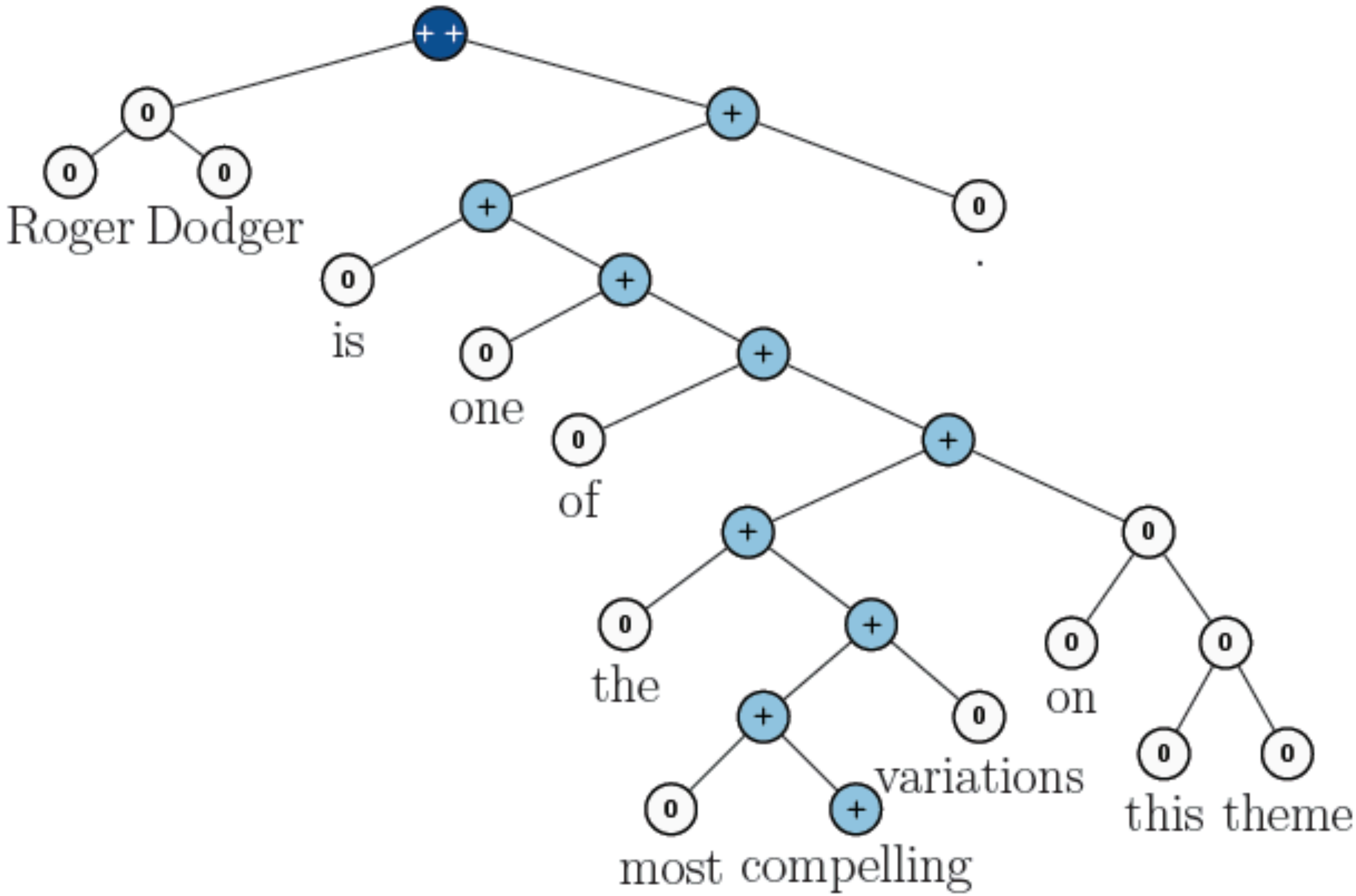
Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

Roger Dodger is one of the most compelling variations on this theme.

Roger Dodger is one of the least compelling variations on this theme.

Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

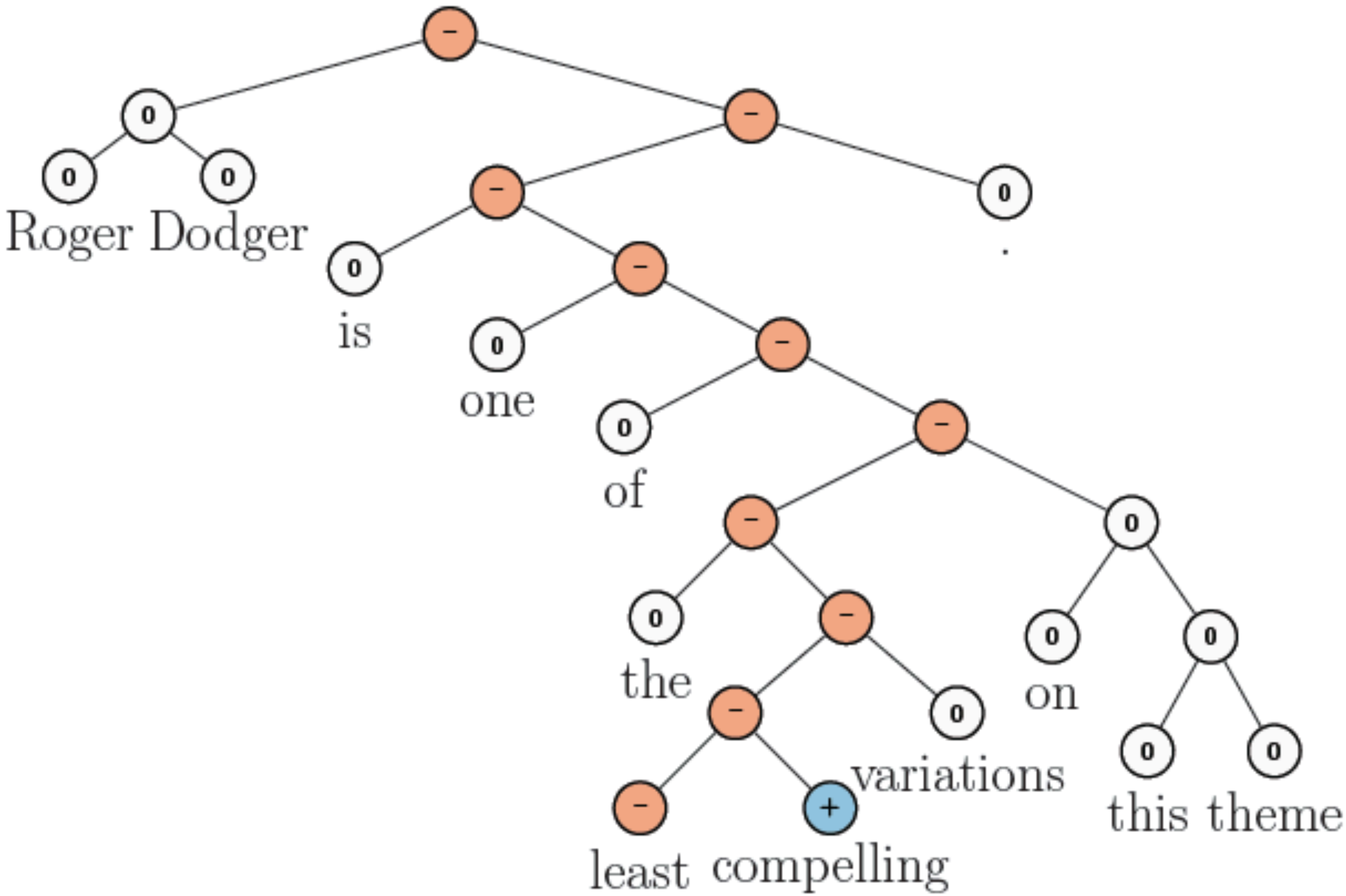
RNTN for Sentiment Analysis



Roger Dodger is one of the most compelling variations on this theme.

Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

RNTN for Sentiment Analysis



Roger Dodger is one of the least compelling variations on this theme.

Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

Accuracy for fine grained (5-class) and binary predictions at the sentence level (root) and for all nodes

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

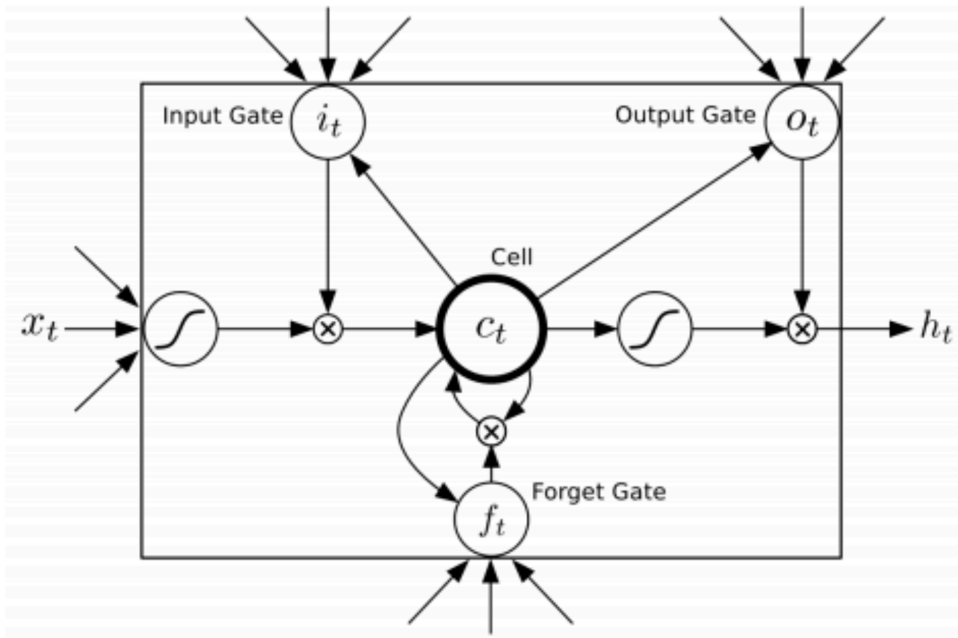
Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

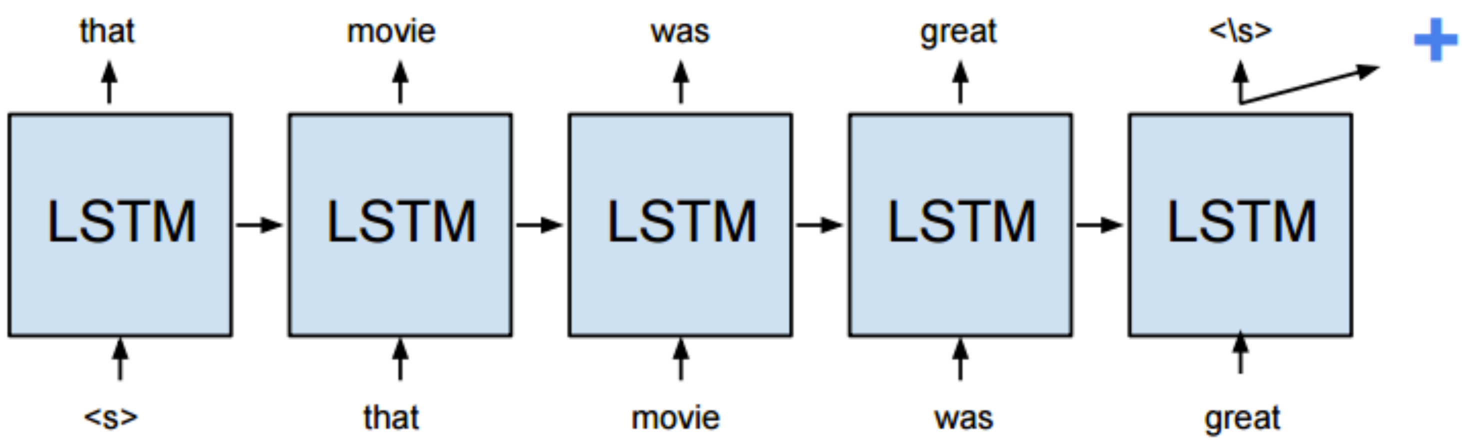
Accuracy of negation detection

Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

Source: Richard Socher et al. (2013) "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", EMNLP 2013

Long Short-Term Memory (LSTM)





Source: https://cs224d.stanford.edu/reports/HongJames.pdf

Deep Learning for Sentiment Analysis CNN RNTN LSTM

Model	Fine (5-class)	Binary
DCNN (Blunsom, et al. 2014)	0.485	0.868
RNTN (Socher, et al. 2013)	0.457	0.854
CNN-non-static (Kim, 2014)	0.480	0.872
CNN-multi-channel (Kim, 2014)	0.474	0.881
DRNN w. pretrained word-embeddings (Irsoy and Cardie, 2014)	0.498	0.866
Paragraph Vector (Le and Mikolov. 2014)	0.487	0.878
Dependency Tree-LSTM (Tai, et al, 2015)	0.484	0.857
Constituency Tree-LSTM (Tai, et al, 2015)	0.439	0.820
Constituency Tree-LSTM (Glove vectors) (Tai, et al, 2015)	0.510	0.880
Paragraph Vector	0.391	0.798
LSTM	0.456	0.843
Deep Recursive-NN	0.469	0.847

Source: https://cs224d.stanford.edu/reports/HongJames.pdf

Performance Comparison of Sentiment Analysis Methods

	Method	Data Set	Acc.	Author
Machine Learning	SVM	Movie reviews	86.40%	Pang, Lee[23]
	CoTraining SVM	Twitter	82.52%	Liu[14]
	Deep learning	Stanford Sentiment Treebank	80.70%	Richard[18]
Lexical based	Corpus	Product reviews	74.00%	Turkey
	Dictionary	Amazon's Mechanical Turk	---	Taboada[20]
Cross-lingual	Ensemble	Amazon	81.00%	Wan,X[16]
	Co-Train	Amazon, ITI68	81.30%	Wan,X.[16]
	EWGA	IMDb movie review	>90%	Abbasi,A.
	CLMM	MPQA,NTCIR,ISI	83.02%	Mengi
Cross-domain	Active Learning	Book, DVD, Electronics, Kitchen	80% (avg)	Li, S
	Thesaurus			Bollegala[22]
	SFA			Pan S J[15]

Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, Vol 139, No. 11, 2016. pp.5-15

Kumar Ravi and Vadlamani Ravi (2015),

"A survey on opinion mining and sentiment analysis: tasks, approaches and applications."

Knowledge-Based Systems,

89, pp.14-46

Knowledge-Based Systems 89 (2015) 14–46

ELSEVIER

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Knowledge-Based SYSTEMS

A survey on opinion mining and sentiment analysis: Tasks, approaches and applications

CrossMark

Kumar Ravi [a,b], Vadlamani Ravi [a,*]

[a] Center of Excellence in CRM and Analytics, Institute for Development and Research in Banking Technology, Castle Hills Road No. 1, Masab Tank, Hyderabad 500057, AP, India
[b] School of Computer & Information Sciences, University of Hyderabad, Hyderabad 500046, AP, India

Table 5
Sentiment classification accuracy reported on common datasets.

S#	Dataset	Articles	Obtained result
1	Pang and Lee [167]	[156]	92.70% accuracy
2		[112]	90.45% F_1
3		[169]	90.2% accuracy
4		[35]	89.6% accuracy
5		[54]	87.70% accuracy
6		[46]	87.4% accuracy
7		[50]	86.5% accuracy
8		[26]	85.35% accuracy
9		[162]	81% F_1
10		[124]	79% accuracy & 86% F_1
11		[61]	76.6% accuracy
12		[69]	76.37% accuracy
13		[48]	75% precision
14		[98]	79% precision
15	Pang et al. [33]	[109]	Approx. 90% accuracy
16		[165]	88.5% accuracy
17		[172]	87% accuracy
18		[33]	82.9% accuracy
19		[156]	78.08% accuracy
20		[180]	75% accuracy
21		[48]	60% precision
22		[195]	86.04%
23	Blitzer et al. [149]	[45]	84.15% accuracy
24		[99]	80.9% (Avg.) accuracy
25		[54]	85.15% (Avg.) Max. 88.65% accuracy on Kitchen reviews
28		[165]	88.7% accuracy
29		[61]	71.92% accuracy

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Classification Accuracy

S#	Dataset	Articles	Obtained result
1	Pang and Lee [167]	[156]	92.70% accuracy
2		[112]	90.45% F_1
3		[169]	90.2% accuracy
4		[35]	89.6% accuracy
5		[54]	87.70% accuracy
6		[46]	87.4% accuracy
7		[50]	86.5% accuracy
8		[26]	85.35% accuracy
9		[162]	81% F_1
10		[124]	79% accuracy & 86% F_1
11		[61]	76.6% accuracy
12		[69]	76.37% accuracy
13		[48]	75% precision
14		[98]	79% precision

B. Pang, L. Lee, A sentiment education: sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, July 2004, p. 271

Sentiment Classification Accuracy

S#	Dataset	Articles	Obtained result
15	Pang et al. [33]	[109]	Approx. 90% accuracy
16		[165]	88.5% accuracy
17		[172]	87% accuracy
18		[33]	82.9% accuracy
19		[156]	78.08% accuracy
20		[180]	75% accuracy
21		[48]	60% precision
22		[195]	86.04%

B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10, Association for Computational Linguistics, 2002, pp. 79–86.

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Classification Accuracy

S#	Dataset	Articles	Obtained result
23	Blitzer et al. [149]	[45]	84.15% accuracy
24		[99]	80.9% (Avg.) accuracy
25		[54]	85.15% (Avg.) Max. 88.65% accuracy on Kitchen reviews
28		[165]	88.7% accuracy
29		[61]	71.92% accuracy

J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL'07, vol. 7, 2007, pp. 187–205 (13, 29).

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Techniques for Sentiment Analysis

Applied techniques	#Articles
SVM	55
Dictionary based approaches (DBA)	41
NB	28
NN	11
DT	9
Maximum entropy	8
Logistic regression	9
Linear regression	8
Ontology	8
LDA	8
Random forest	4
SVR	5
CRF and rCRP	5
Boosting	4
SVM-SMO	4
Fuzzy logic	3
Rule miner	4
EM	3
K-medoids	1
RBF NN	1

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Analysis Articles in Journals (2002-2014)

S#	Name of journals	#Articles
1	Expert Systems with Applications	33
2	Decision Support Systems	28
3	Knowledge-based Systems	17
4	IEEE Intelligent Systems	12
5	IEEE Transactions on Knowledge and Data Engineering	6
6	IEEE Transactions on Affective Computing	3
7	Information Sciences	3
8	Information Processing and Management	3
9	Computer Speech and Language	2
10	Communications of the ACM	2
11	Journal of Computer Science and Technology	2
12	Journal of Informetrics	2
13	Information Retrieval	2
14	Computer Speech and Language	2
15	Inf. Retrieval	1

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Publicly Available Datasets for Sentiment Analysis

S#	Data set	Type	Lang.	Web resource	Details
1	Stanford large movie data set	Movie Reviews	English	http://ai.stanford.edu/~amaas/data/sentiment/	Movie Reviews
2	COAE2008	Product Reviews	Chinese	http://ir-china.org.cn/coae2008.html	2739 documents for movie, education, finance, economics, house, computer, mobile phones, etc. 1525 +ve, 1214 –ve
3	Boacar	Car Reviews	Chinese	http://www.riche.com.cn/boacar/	11 type of car TradeMarks and total review 1000 words, having 578 POS, 428 –ve reviews
4	[187]	Reviews, forums	English	http://sifaka.cs.uiuc.edu/~wang296/Data/	Accessed: 27 August, 2014
5	[188]	Reviews	English	http://uilab.kaist.ac.kr/research/WSDM11	Aspect oriented dataset. Accessed: 18 December, 2014
6	Movie-v2.0	Movie Reviews	English	http://www.cs.cornell.edu/people/pabo/movie-review-data/	Data size: 2000 Positive: 1000 Negative: 1000
7	Multi-domain	Multi-domain	English	http://www.cs.jhu.edu/~mdreze/datasets/sentiment	
8	SkyDrive de Hermit Dave	Spanish Word Lists	Spanish	https://skydrive.live.com/?cid=3732e80b128d016f&id=3732E80B128D016F%213584	
9	TripAdvisor	Reviews	Spanish	http://clic.ub.edu/corpus/es/node/106	18,000 customer reviews on hotels and restaurants from Hopinion
10	[38]	Multi-Domain	English	www2.cs.uic.edu/~liub/FBS/sentiment-analysis.html	6800 opinion words on 10 different products
11	TBOD [144]	Reviews	English		Product Review on Cars, Headphones, Hotels
12	[68]	Product Reviews	English	http://www.lsi.us.es/_fermin/index.php/Datasets	Product Reviews from Epinion.com on headphones 587 reviews, hotels 988 reviews and cars 972 reviews
13	[148]	Movie Reviews	Turkish	http://www.win.tue.nl/~mpechen/projects/smm/#Datasets	5331 positive and 5331 negative reviews on movie
14	[148]	Product Reviews	Turkish	http://www.win.tue.nl/~mpechen/projects/smm/#Datasets	700 +ve &700 –ve reviews on books, DVD, electronics, kitchen appliances
15	ISEAR	English sentences	English	www.affective-sciences.org/system/files/page/2636/ISEAR.zip	The dataset contains 7666 such statements, which include 18,146 sentences, 449,060 running words.
16	[149]	Product Reviews	English	http://www.cs.jhu.edu/~mdredze/datasets/sentiment/	Amazon reviews on 4 domain (books, DVDs, electronics, kitchen appliances)
17	DUC data, NIST	Texts	English	http://www-nlpir.nist.gov/projects/duc/data.html, http://www.nist.gov/tac/data/index.html	Text summarization data
18	[70]	Restaurant and Hotel Reviews	English	http://uilab.kaist.ac.kr/research/WSDM11	Restaurant and Hotel Reviews from Amazon and Yelp
19	[114]	Restaurant Reviews	Cantonese	http://www.openrice.com	Reviews on restaurant
20	[125]	Biographical Articles	Dutch	http://www.iisg.nl/bwsa	574 Biographical articles
21	Spinn3r dataset	Multi-Domain	English	http://www.icwsm.org/2011/data.php	
22	[86]	Ironic Dataset	English	http://users.dsic.upv.es/grupos/nle/	3163 ironic reviews on five products
23	HASH [179]	Tweets	English	http://demeter.inf.ed.ac.uk	31,861 Pos tweets, 64,850 Neg tweets, 125,859 Neu tweets
24	EMOT [179]	Tweets and Emoticons	English	http://twittersentiment.appspot.com	230,811 Pos & 150,570 Neg tweets
25	ISIEVE [179]	Tweets	English	www.i-sieve.com	1520 Pos tweets, 200 Neg tweets, 2295 Neu tweets
26	[177]	Tweets	English	e-mail: apoorv@cs.columbia.edu	11,875 tweets
27	[52]	Opinions	English	http://patientopinion.org.uk	2000 patient opinions
28	[96]	Tweets	English	http://goo.gl/UQvdx	667 tweets
29	[39]	Movie Reviews	English	http://ai.stanford.edu/~amaas/data/sentiment/	50,000 movie reviews
30	[164]	Tweets	English	http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip	
31	[210]	Spam Reviews	English	http://myleott.com/op_spam	400 deceptive and 400 truthful reviews in positive and negative category. Last Accessed by: 12 April, 2015
32	[230]	Sarcasm and nasty reviews	English	https://nlds.soe.ucsc.edu/iac	1000 discussions, ~390,000 posts, and some ~73,000,000 words

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Analysis Datasets

- Product Reviews (PR)
- Movie Reviews (MR)
- Restaurant Reviews (RR)
- Micro-blog (MB)
- Global domain (G)

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Analysis Dictionary

- SenticNet (SN)
- WordNet (WN)
- ConceptNet (CN)
- WordNet-Affect (WNA)
- Bing Liu Opinion Lexicon (OL)

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Summary of reviewed articles

Ref.	Concepts and techniques utilized	P	L	Type of data	Dictionary
[8]	Page rank, Gradient descent, Linear regression	2	E	PR	
[11]	Link mining, Collective classification	NA	E	MB	
[12]	AdaBoost.HM	2	E	G	GI
[13]	DBA	5	E	News Comments	New Lexicon
[18]	DBA, SOFNN, Linear regression	2, 7	E	MB, DJIA data	OF, GPOMS
[21]	Regression, Random walk, SVM	4, 2	E		ANEW, CN
[22]	Cohen's K coefficient	6, 2	I	MB	SN
[23]	Fuzzy clustering, PMI, DBA	6, 2	E	G	WNA, SN, WN.
[24]	DBA	NA	D	G	Dutch WN
[25]	Association Miner CBA, DBA	2	E	PR	WN
[26]	SVM	2	E	PR	
[27]	Markov-Chain Monte Carlo (MCMC)	NA	E	Online discussion	
[29]	SVM with Gaussian Kernel	3, 2			MPQA
[30]	Ontology, K-means	2	E		ReiAction [122],[a] Family Relation[b]
[32]	PMI-IR	2	E	Multi-domain	
[33]	NB, SVM, ME	2	E	MR	
[35]	Ontology, DBA	2	E	MR	SWN
[36]	New Algorithm, DBA	2	E	MR, Book, Mobile	11 dictionaries
[37]	CRF	NA		PR	
[40]	Multinomial inverse regression	3	E	MB	
[41]	FFCA, Lattice	2	E	PR	
[43]	Analytic hierarchy process	NA	C	MB	
[44]	Fisher's discriminant ratio, SVM	2	C	PR	
[45]	Semantic orientation, SVM	3, 2	E	PR	SWN
[46]	MNB, ME, SVM	3, 2	E, D, F	Forum, Blog, PR	
[47]	DBA	2	D, E	News	
[48]	Semantic orientation and BackProp	2	E	Blogs, PR	
[49]	Probabilistic Matrix Factorization	NA	C	MB	
[50]	NB, SVM, NN	2	E	PR	
[51]	SVM, NN	NA	C	MB	
[52]	DNN, CNN, K-medoids, KNN	NA	E	G	CN, WNA, AffectiveSpace
[53]	SVM, NN, MLP, DT, GA, Stepwise LR, RBC	2	E	News	
[54]	NB, ME, SVM	2	E	PR	
[55]	DBA	5, 2	E	MB	
[56]	NB, EM	NA	E	PR	WN
[57]	SVM, NN	5, 2	E	MB	

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Summary of reviewed articles

Ref.	Concepts and techniques utilized	P	L	Type of data	Dictionary
[58]	SVM	NA	E	Suicide Notes	WN, SWN.
[59]	EM	NA	E, S	PR	fullStrengthLexicon[c]
[60]	ME	NA	E	MB	
[61]	Bayesian Model, LDA	2	E	PRMPQA, Appraisal Lexicons[d]	
[62]	Fuzzy Set, Ontology	2	C	PR	
[63]	ME, Bootstrapping, IG	3, 2	C	PR	Hownet, NEUCSP[e]
[64]	DBA	NA	E	e-mail, book	Roget Thesaurus[f]
[66]	NB, ME, DT, KNN, SVM	NA	C, E	PR, Forums	
[67]	SVM, DBA	2	E	PR	GI
[68]	DBA, Random walk algorithm	2	E	PR	
[69]	DBA	2	E	PR	
[70]	Linear Regression	NA	C	PR, social network	
[73]	BayesNet, J48, Jrip, SVM, NB, ZeroR, Random	5, 2	E	News, Magazine	
[74]	Semantic relationships	2	E		SWN, GI
[75]	Multilingual bootstrapping and cross-lingual bootstrapping, linear regression, IG	NA	E, R		WN
[76]	Bootstrapping, DT, MLP, PCA, SLR, SMO-SVM	2	E	Phone Reviews	WN
[77]	LR, SVM, RF	2	B	e-mails	
[78]	Discretionary accrual model	NA	E	Book Reviews	
[80]	Bayes-Nash equilibria	NA	E	MB	
[81]	RF	NA	E	PR	
[85]	DBA	3, 2	E	MB	SWN
[86]	Semantic, NB, SVM, DT	NA		PR	WN, MSOL, WNA
[88]	SVM, LR, CRF	NA	E	PR	
[90]	SVM, NB	NA	E	MB	
[91]	K-means, SVM	NA	C	Forums	
[92]	HMM-LDA	NA	E	PR	
[93]	Two level CRF	NA	E	PR	
[94]	Corpus based approach, SVM, NB, C4.5, BBR	5, 2	E, S	PR	SWN, Tree Tagger
[95]	SVM	NA	E		WNA, LIWC, VerbOcean, CN
[96]	DBA, Ontology	2	E	MB	
[97]	SMO-SVM, DBA	2	E	MR	SWN, WN
[98]	NB and Ontology	2	E	PR, MR	WN
[99]	Cosine similarity, L1 regularized logistic regression	2	E	PR	WN and SWN
[100]	Association miner CBA	NA	C	PR	
[101]	NN, C4.5, CART, SVM, NB	2	E	MB	

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Summary of reviewed articles

Ref.	Concepts and techniques utilized	P	L	Type of data	Dictionary
[102]	SVM	2	C	HR, PR	TU lexicon[g]
[107]	LDA, DBA	2	E	RR, HR	MPQA, SWN
[108]	SVM	2	A	Dialects, MB, Wiki Talks, Forums	
[109]	Rule-based multivariate features, SVM	2	E	MR, PR, Automobile	
[110]	DBA	2	S	MR	BLEL, WN
[111]	NB, SVM	2	E	RR	SWN
[112]	DBA, RBC, SVM	2	E	MR, Product, MySpace texts	WN, GI
[114]	IG, DBA	2	CT	RR	
[115]	SVM, Statistical approach	2	E, C	HR, Mobile	
[116]	DBA, SVM, NB, LR, J48, Jrip, AdaBoost, Decision Table, MLP, NB.	2	E	MySpace	SentiStrength
[117]	DBA	2	E	MB	SWN
[118]	SMO-SVM, LR, AdaBoost, SVR, DT, NB, J48, Jrip	2	E	Social Media	SentiStrength
[121]	Adaptive-NB	NA	C	PR	
[123]	SVR	6, 2	C	Sina-Wiebo	
[124]	NB	2	E	Social & Mass media	
[125]	Lexical features, NB, Linear SVM, Jrip, KNN	2	D	Biographies	Brouwers thesaurus
[126]	DBA	2	E	MB	OL
[127]	DBA	5, 2	E	G	SentiStrength
[130]	SVR, RBF	NA			
[131]	SVM, NB	3	E	MB, PR	
[132]	New Algorithm	NA		PR	
[148]	SVM, NB, ME	2	E, T		
[154]	New algorithm, Lexical features	3	E	PR	
[155]	SP-LSA, AR, EM, ε-SVR	2	E	MR	2030 appraisal words
[156]	Tabu search, MB, NB, SVM, ME	2	E	MR and News	
[157]	PSO and SVM	2	E	MB	
[158]	DBA	3, 2	E	Mobile Reviews	Moreo et al. [13]
[160]	EWGA, SVM, Bootstrapping	2	E, A	Forums	
[162]	Class sequential rules	3	E	MR	SWN
[163]	DBA, SVM, NB, Logistic, NN	2	E	MB	10 dictionaries
[165]	Semantic, GI, Chi-square, SVM	2	E	MR and PR	
[166]	Semantic	2	C	HR	
[167]	NB, SVM, Min.-cut in the graph	2	E	MR	
[168]	Linear classifiers, Clique, MIRA classifier	2	E	PR	
[169]	DBA, SVM, and SMO-SVM	2	E	MR	WN
[170]	DBA	3	J	MR and PR	Yi et al. [7] lexicon

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Summary of reviewed articles

Ref.	Concepts and techniques utilized	P	L	Type of data	Dictionary
[171]	DBA	2	E	Web pages, News	
[172]	SVM, Osgoodian values, PMI	2	E	MR	WN
[173]	Transfer-based machine translation	2	J	Camera Review	
[174]	ME	2	E	MR	
[175]	DBA, Sigmoid scoring	2	C	Blogs	Hownet
[176]	SVM, PMI	2	E	MB	GI
[177]	Convolution kernels [152], SVM, DBA	2, 3	E	MB	WN, DAL [151]
[178]	Statistical method of OASYS [8]	C	E	News articles	OASYS
[179]	Boosting, SVM	3	E	MB	MPQA, NetLingo
[180]	Bipartite graph, Regularization operator	2	E	Blogs	
[182]	LDA, Ontology, MCMC	2	E	Multi-domain	OF
[183]	SVM, TF-IDF	2	E	News headlines, Forex Rate	SWN
[184]	Vector space model	3	E	News articles	Harvard IV
[185]	Modified LDA	5	E	PR	
[186]	Recursive Chinese Restaurant Process	2	E	PR	
[189]	LDA incorporated with domain knowledge	NA	E	Camera and HR	
[190]	CRF, syntactic and semantic features	2	E	PR, Facebook text	
[191]	LDA, Appraisal expression pattern	NA	E	HR, RR, PR	
[192]	PMI, TF-IDF	2	E	PR	GI
[193]	TF-IDF, Domain relevance	2	C	HR, Cellphone	
[194]	Ontology	2	E	Automobile, PR, SW	SWN, GI, OL
[195]	Ontology	2	E	MR	WN
[196]	Ontology, Maximum-Likelihood	2	E	MR	GI
[197]	PCA, SVM, LR, Bayesian Boosting, Bagged SVM	2	E	PR	
[200]	SVM	2	E	PR	
[202]	DBA, Graphical Techniques	2	E	G	CN, DBPedia, WN
[203]	DBA	2	E	MB	CN, WN, JMDict, Verbosity
[205]	Graphical techniques	2	GE	MB	SWN, SN 3
[206]	DBA	8	E	Google n-grams	SN 3, WNANRC, SAT
[207]	Ontology, DBA	4	E	PR, MR	CN
[209]	SVM, NB, J48	3	S	Facebook text	Spanish LIWC
[210]	SVM, RF	3	S	Apontador	
[211]	DBA	2	S	MB	SN 3, WeFeelFine
[212]	NB, SVM, DBA	2	E	PR	LIWC
[213]	Ontology, DBA, ELM	2	E	G	AffectiveSpace
[214]	Ontology, DBA, SVM, FCM	2	E	G	SN 3, WNA, AffectiveSpace
[216]	DBA, Ontology	2	E	PR, MR	WN, CN
[217]	Rule base classifier, NB	2	E	Dialogue	SN 3

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Summary of reviewed articles

Ref.	Concepts and techniques utilized	P	L	Type of data	Dictionary
[218]	Bootstrapping, PMI, DBA	NA	E	PR	
[220]	DBA, Binomial LR	NA	E	PR	LIWC
[221]	Product, Review & Reviewer Information	NA	E	PR	
[222]	Linear Regression	2	E	PR	
[223]	Linear Regression	NA	E	PR	
[224]	Linear Regression	NA	E	PR	
[225]	SVM	NA	E	PR	
[226]	MLP	NA	E	PR	
[227]	RFM, SVR	NA	E	PR	
[228]	RF, NB, SVM	NA	E	PR	
[229]	DBA	2	E	PR	
[231]	Linear Regression	NA	E	PR	
[232]	PU-learning	NA	E	PR	
[240]	LDA, SVM, PMI	NA	C	PR	
[241]	PageRank algorithm, DBA	NA	C	PR	
[243]	PMI-IR, RCut, Apriori Algo.	NA	C	PR	

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

IMDB
Large Movie Review Dataset

- This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets.
- We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing.
- There is additional unlabeled data for use as well.
- Raw text and already processed bag of words formats are provided.
- Large Movie Review Dataset v1.0
  - http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz

Source: http://ai.stanford.edu/~amaas/data/sentiment/

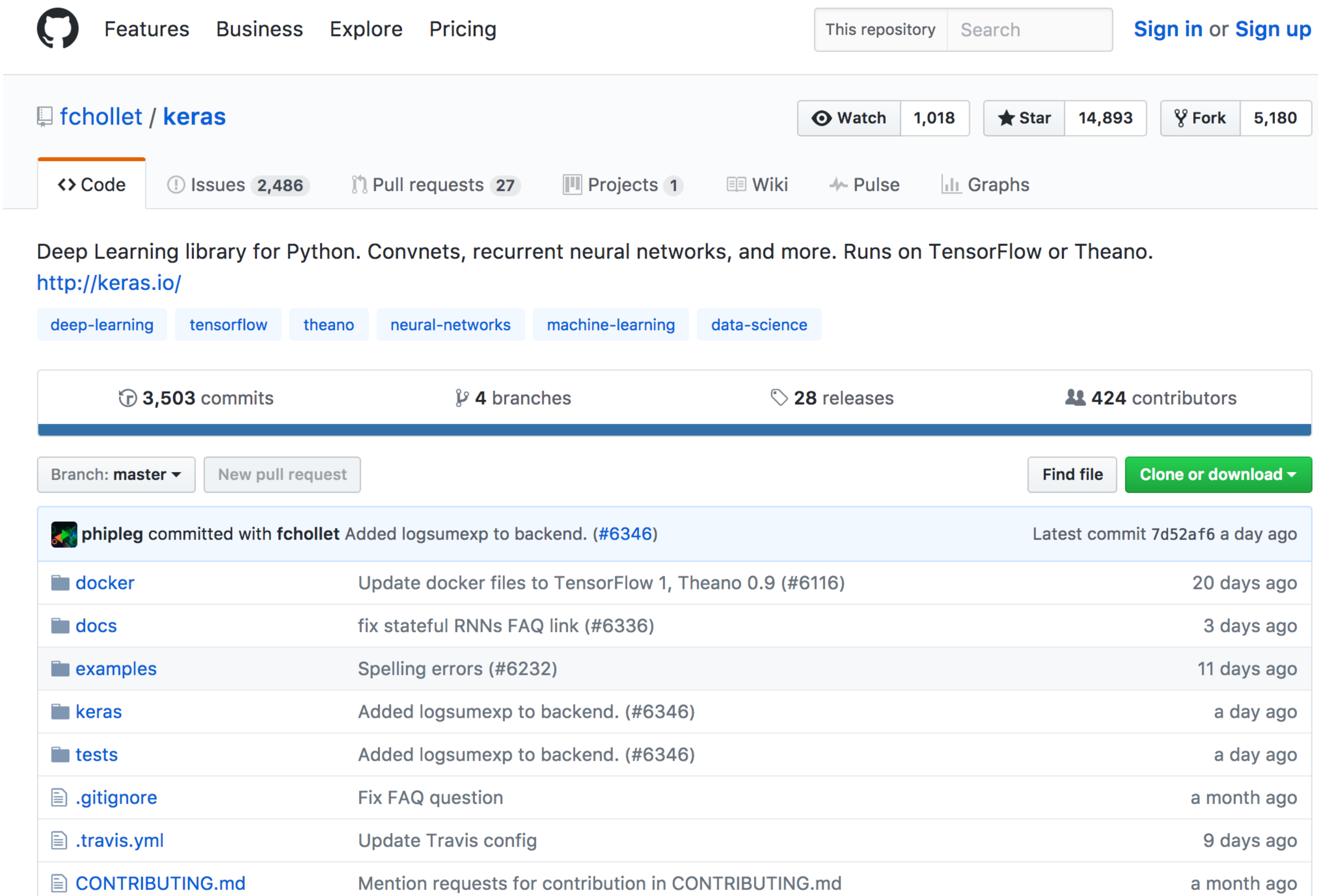
IMDB Dataset (Mass et al., 2011)

Features	PL04	Our Dataset	Subjectivity
Bag of Words (bnc)	85.45	87.80	87.77
Bag of Words (bΔt'c)	85.80	88.23	85.65
LDA	66.70	67.42	66.65
LSA	84.55	83.96	82.82
Our Semantic Only	87.10	87.30	86.65
Our Full	84.65	87.44	86.19
Our Full, Additional Unlabeled	87.05	87.99	87.22
Our Semantic + Bag of Words (bnc)	88.30	88.28	88.58
Our Full + Bag of Words (bnc)	87.85	88.33	88.45
Our Full, Add'l Unlabeled + Bag of Words (bnc)	88.90	88.89	88.13
Bag of Words SVM (Pang and Lee, 2004)	87.15	N/A	90.00
Contextual Valence Shifters (Kennedy and Inkpen, 2006)	86.20	N/A	N/A
tf.Δidf Weighting (Martineau and Finin, 2009)	88.10	N/A	N/A
Appraisal Taxonomy (Whitelaw et al., 2005)	90.20	N/A	N/A

Table 2: Classification accuracy on three tasks. From left to right the datasets are: A collection of 2,000 movie reviews often used as a benchmark of sentiment classification (Pang and Lee, 2004), 50,000 reviews we gathered from IMDB, and the sentence subjectivity dataset also released by (Pang and Lee, 2004). All tasks are balanced two-class problems.

Source: Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*

Keras Github



https://github.com/fchollet/keras

Keras Examples

Features Business Explore Pricing | This repository Search | Sign in or Sign up

fchollet / keras | Watch 1,018 | Star 14,893 | Fork 5,181

Code | Issues 2,486 | Pull requests 27 | Projects 1 | Wiki | Pulse | Graphs

Branch: master | keras / examples / | Create new file | Find file | History

Mohanson committed with fchollet Spelling errors (#6232) — Latest commit 5bd3976 11 days ago

..

README.md	Adding mnist_acgan.py example link in README (#4876)	4 months ago
addition_rnn.py	Spelling errors (#6232)	11 days ago
antirectifier.py	Style fix for examples. (#5980)	28 days ago
babi_memnn.py	Style fixes in example scripts	a month ago
babi_rnn.py	Style fixes in example scripts	a month ago
cifar10_cnn.py	fix rmsprop learning rate for convergence (#6182)	17 days ago
conv_filter_visualization.py	Finish updating examples.	a month ago
conv_lstm.py	Update a number of example scripts.	2 months ago
deep_dream.py	Finish updating examples.	a month ago
image_ocr.py	Fixed URL for wordlist.tgz in image_ocr.py (#6136)	20 days ago
imdb_bidirectional_lstm.py	Finish updating examples.	a month ago
imdb_cnn.py	Finish updating examples.	a month ago
imdb_cnn_lstm.py	Style fix for examples. (#5980)	28 days ago

https://github.com/fchollet/keras/tree/master/examples

Keras Examples

- imdb_bidirectional_lstm.py Trains a Bidirectional LSTM on the IMDB sentiment classification task.
- imdb_cnn.py Demonstrates the use of Convolution1D for text classification.
- imdb_cnn_lstm.py Trains a convolutional stack followed by a recurrent stack network on the IMDB sentiment classification task.
- imdb_fasttext.py Trains a FastText model on the IMDB sentiment classification task.
- imdb_lstm.py Trains a LSTM on the IMDB sentiment classification task.
- lstm_benchmark.py Compares different LSTM implementations on the IMDB sentiment classification task.
- lstm_text_generation.py Generates text from Nietzsche's writings.

Source: https://github.com/fchollet/keras/tree/master/examples

Keras IMDB Movie reviews sentiment classification

- Dataset of 25,000 movies reviews from IMDB, labeled by sentiment (positive/negative).
- Reviews have been preprocessed, and each review is encoded as a sequence of word indexes (integers).
- For convenience, words are indexed by overall frequency in the dataset, so that for instance the integer "3" encodes the 3rd most frequent word in the data.
- This allows for quick filtering operations such as: "only consider the top 10,000 most common words, but eliminate the top 20 most common words".
- As a convention, "0" does not stand for a specific word, but instead is used to encode any unknown word.

Source: https://keras.io/datasets/

Keras IMDB load_data

```
def load_data(path='imdb.npz',
              num_words=None,
              skip_top=0,
              maxlen=None,
              seed=113,
              start_char=1,
              oov_char=2,
              index_from=3):
  path = get_file(
      path, origin='https://s3.amazonaws.com/text-datasets/imdb.npz')
  f = np.load(path)
  x_train = f['x_train']
  labels_train = f['y_train']
  x_test = f['x_test']
  labels_test = f['y_test']
  f.close()
```

Source: keras/datasets/imdb.py

Keras IMDB get_word_index

```
def get_word_index(path='imdb_word_index.json'):
  path = get_file(
      path,
      origin='https://s3.amazonaws.com/text-datasets/imdb_word_index.json')
  f = open(path)
  data = json.load(f)
  f.close()
  return data
```

Source: keras/datasets/imdb.py

Keras IMDB CNN

localhost:8888/notebooks/Documents/SCDBA/DL/Keras_imdb_cnn.ipynb

jupyter Keras_imdb_cnn Last Checkpoint: 15 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Python 3

Code CellToolbar

```
from __future__ import print_function
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.layers import Embedding
from keras.layers import Conv1D, GlobalMaxPooling1D
from keras.datasets import imdb

# set parameters:
max_features = 5000
maxlen = 400
batch_size = 32
embedding_dims = 50
filters = 250
kernel_size = 3
hidden_dims = 250
epochs = 2

print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)

print('Build model...')
model = Sequential()

# we start off with an efficient embedding layer which maps
# our vocab indices into embedding_dims dimensions
model.add(Embedding(max_features,
                    embedding_dims,
```


Source: https://github.com/fchollet/keras/blob/master/examples/imdb_cnn.py

Keras IMDB CNN

localhost:8888/notebooks/Documents/SCDBA/DL/Keras_imdb_cnn.ipynb

jupyter Keras_imdb_cnn Last Checkpoint: 19 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Python 3

CellToolbar

```
model.add(Embedding(max_features,
                    embedding_dims,
                    input_length=maxlen))
model.add(Dropout(0.2))

# we add a Convolution1D, which will learn filters
# word group filters of size filter_length:
model.add(Conv1D(filters,
                 kernel_size,
                 padding='valid',
                 activation='relu',
                 strides=1))
# we use max pooling:
model.add(GlobalMaxPooling1D())

# We add a vanilla hidden layer:
model.add(Dense(hidden_dims))
model.add(Dropout(0.2))
model.add(Activation('relu'))

# We project onto a single unit output layer, and squash it with a sigmoid:
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          validation_data=(x_test, y_test))
```

```
Using TensorFlow backend.
```

```
Loading data...
Downloading data from https://s3.amazonaws.com/text-datasets/imdb.npz
25000 train sequences
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_cnn.py

Keras IMDB CNN

localhost:8888/notebooks/Documents/SCDBA/DL/Keras_imdb_cnn.ipynb

jupyter Keras_imdb_cnn Last Checkpoint: 13 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Python 3

Code CellToolbar

```
# we use max pooling:
model.add(GlobalMaxPooling1D())

# We add a vanilla hidden layer:
model.add(Dense(hidden_dims))
model.add(Dropout(0.2))
model.add(Activation('relu'))

# We project onto a single unit output layer, and squash it with a sigmoid:
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          validation_data=(x_test, y_test))
```

```
Using TensorFlow backend.
```

```
Loading data...
Downloading data from https://s3.amazonaws.com/text-datasets/imdb.npz
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 400)
x_test shape: (25000, 400)
Build model...
Train on 25000 samples, validate on 25000 samples
Epoch 1/2
25000/25000 [==============================] - 266s - loss: 0.4110 - acc: 0.8012 - val_loss: 0.2965 - val_acc: 0.8739
Epoch 2/2
25000/25000 [==============================] - 286s - loss: 0.2429 - acc: 0.9020 - val_loss: 0.2726 - val_acc: 0.8862
```

```
Out[1]: <keras.callbacks.History at 0x11dc37b00>
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_cnn.py

Keras IMDB CNN

```
python imdb_cnn.py
Using TensorFlow backend.
Loading data...
Downloading data from https://s3.amazonaws.com/text-datasets/imdb.npz
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 400)
x_test shape: (25000, 400)
Build model...
Train on 25000 samples, validate on 25000 samples
Epoch 1/2
25000/25000 [==============================] - 157s - loss: 0.4050 - acc: 0.8065 - val_loss: 0.2924 - val_acc: 0.8750
Epoch 2/2
25000/25000 [==============================] - 128s - loss: 0.2433 - acc: 0.9040 - val_loss: 0.2701 - val_acc: 0.8865
Exception ignored in: <bound method BaseSession.__del__ of <tensorflow.python.client.session.Session object at 0x0000019F153C2A20>>
Traceback (most recent call last):
  File "C:\Program Files\Anaconda3\lib\site-packages\tensorflow\python\client\session.py", line 587, in __del__
AttributeError: 'NoneType' object has no attribute 'TF_NewStatus'
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_cnn.py

Keras IMDB LSTM

```
from __future__ import print_function
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.datasets import imdb

max_features = 20000
maxlen = 80  # cut texts after this number of words (among top max_features most common words)
batch_size = 32

print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)

print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

print('Train...')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=15,
          validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test,
                            batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

Keras IMDB LSTM

```
from __future__ import print_function
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.datasets import imdb
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

Keras IMDB LSTM

```
max_features = 20000
maxlen = 80  # cut texts after this number of words (among top
max_features most common words)
batch_size = 32

print('Loading data...')
(x_train, y_train), (x_test, y_test) =
imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

Keras IMDB LSTM

```
print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
                   optimizer='adam',
                   metrics=['accuracy'])
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

Keras IMDB LSTM

```
print('Train...')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=15,
          validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test,

batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)
```

Source: https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

Keras IMDB LSTM

```
python imdb_lstm.py
Using TensorFlow backend.
Loading data...
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 80)
x_test shape: (25000, 80)
Build model...
Train...
Train on 25000 samples, validate on 25000 samples
Epoch 1/15
25000/25000 [==============================] - 111s - loss: 0.4561 - acc: 0.7837 - val_loss: 0.3892 - val_acc: 0.8275
Epoch 2/15
25000/25000 [==============================] - 112s - loss: 0.2947 - acc: 0.8792 - val_loss: 0.4266 - val_acc: 0.8353
Epoch 3/15
25000/25000 [==============================] - 111s - loss: 0.2122 - acc: 0.9178 - val_loss: 0.4133 - val_acc: 0.8284
Epoch 4/15
25000/25000 [==============================] - 112s - loss: 0.1461 - acc: 0.9450 - val_loss: 0.4670 - val_acc: 0.8260
Epoch 5/15
25000/25000 [==============================] - 113s - loss: 0.1038 - acc: 0.9633 - val_loss: 0.5580 - val_acc: 0.8203
Epoch 6/15
25000/25000 [==============================] - 113s - loss: 0.0739 - acc: 0.9749 - val_loss: 0.6738 - val_acc: 0.8174
Epoch 7/15
25000/25000 [==============================] - 113s - loss: 0.0542 - acc: 0.9810 - val_loss: 0.7463 - val_acc: 0.8154
Epoch 8/15
25000/25000 [==============================] - 113s - loss: 0.0428 - acc: 0.9856 - val_loss: 0.8131 - val_acc: 0.8157
Epoch 9/15
25000/25000 [==============================] - 115s - loss: 0.0334 - acc: 0.9889 - val_loss: 0.8566 - val_acc: 0.8165
Epoch 10/15
25000/25000 [==============================] - 114s - loss: 0.0248 - acc: 0.9920 - val_loss: 0.9186 - val_acc: 0.8165
Epoch 11/15
25000/25000 [==============================] - 116s - loss: 0.0156 - acc: 0.9955 - val_loss: 0.9016 - val_acc: 0.8082
Epoch 12/15
25000/25000 [==============================] - 117s - loss: 0.0196 - acc: 0.9942 - val_loss: 0.9720 - val_acc: 0.8124
Epoch 13/15
25000/25000 [==============================] - 120s - loss: 0.0152 - acc: 0.9957 - val_loss: 1.0064 - val_acc: 0.8148
Epoch 14/15
25000/25000 [==============================] - 121s - loss: 0.0128 - acc: 0.9961 - val_loss: 1.1103 - val_acc: 0.8121
Epoch 15/15
25000/25000 [==============================] - 114s - loss: 0.0110 - acc: 0.9970 - val_loss: 1.0173 - val_acc: 0.8132
25000/25000 [==============================] - 23s
Test score: 1.01734088922
Test accuracy: 0.8132
```

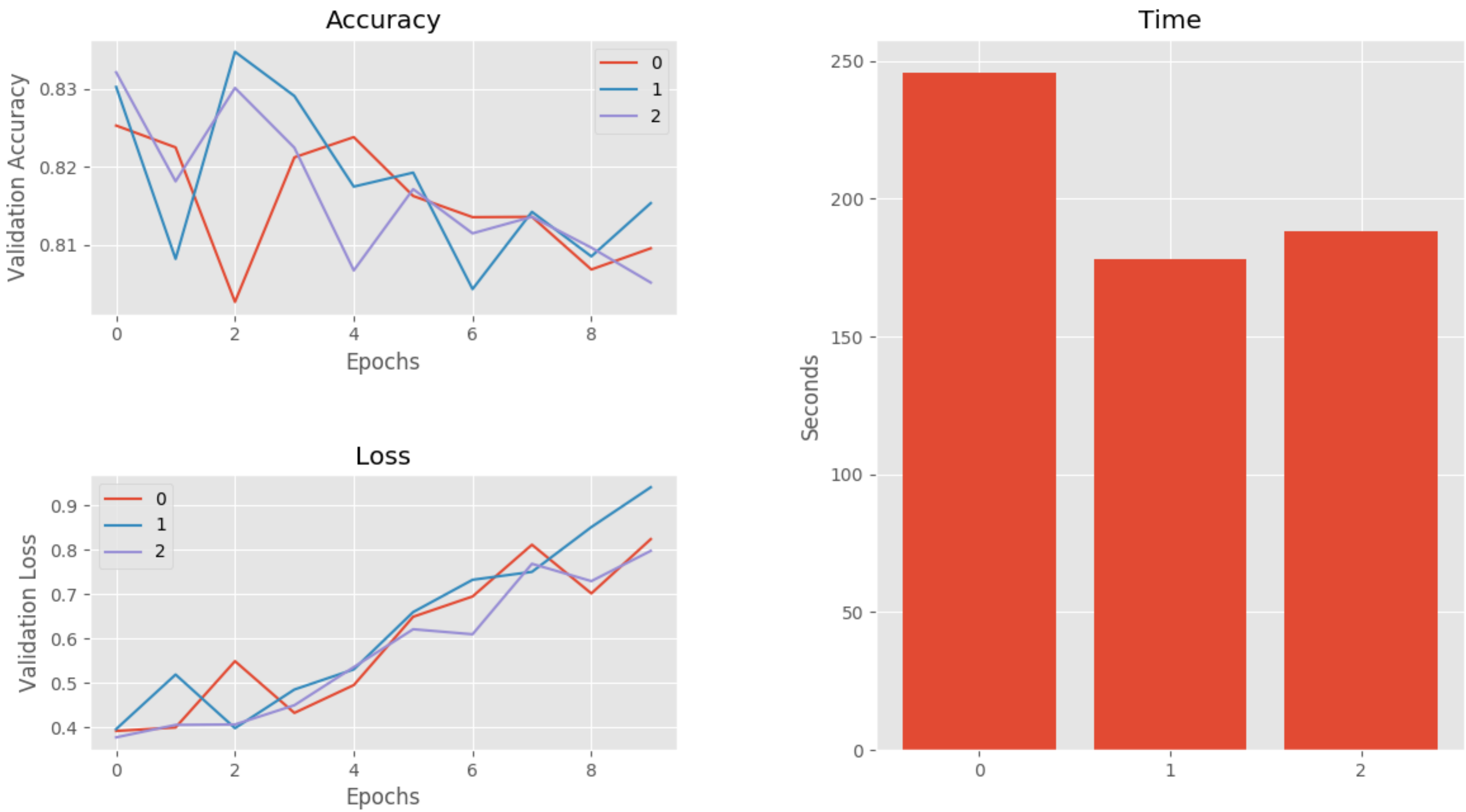
Keras IMDB FastText

```
python imdb_fasttext.py
Using TensorFlow backend.
Loading data...
25000 train sequences
25000 test sequences
Average train sequence length: 238
Average test sequence length: 230
Pad sequences (samples x time)
x_train shape: (25000, 400)
x_test shape: (25000, 400)
Build model...
Train on 25000 samples, validate on 25000 samples
Epoch 1/5
25000/25000 [==============================] - 14s - loss: 0.6102 - acc: 0.7397 - val_loss: 0.5034 - val_acc: 0.8105
Epoch 2/5
25000/25000 [==============================] - 14s - loss: 0.4019 - acc: 0.8656 - val_loss: 0.3697 - val_acc: 0.8654
Epoch 3/5
25000/25000 [==============================] - 14s - loss: 0.3025 - acc: 0.8959 - val_loss: 0.3199 - val_acc: 0.8791
Epoch 4/5
25000/25000 [==============================] - 14s - loss: 0.2521 - acc: 0.9113 - val_loss: 0.2971 - val_acc: 0.8848
Epoch 5/5
25000/25000 [==============================] - 14s - loss: 0.2181 - acc: 0.9249 - val_loss: 0.2899 - val_acc: 0.8855
Exception ignored in: <bound method BaseSession.__del__ of <tensorflow.python.client.session.Session object at
0x000001E3257DB438>>
Traceback (most recent call last):
  File "C:\Program Files\Anaconda3\lib\site-packages\tensorflow\python\client\session.py", line 587, in __del__
AttributeError: 'NoneType' object has no attribute 'TF_NewStatus'
```

Keras IMDB CNN LSTM

```
python imdb_cnn_lstm_2.py
Using TensorFlow backend.
Loading data...
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 100)
x_test shape: (25000, 100)
Build model...
Train...
Train on 25000 samples, validate on 25000 samples
Epoch 1/2
25000/25000 [==============================] - 64s - loss: 0.3824 - acc: 0.8238 - val_loss: 0.3591 - val_acc: 0.8467
Epoch 2/2
25000/25000 [==============================] - 63s - loss: 0.1953 - acc: 0.9261 - val_loss: 0.3827 - val_acc: 0.8488
24990/25000 [============================>.] - ETA: 0s
Test score: 0.382728585386
Test accuracy: 0.848799994493
```

Keras LSTM Benchmark



Source: https://github.com/fchollet/keras/blob/master/examples/lstm_benchmark.py

imdb_lstm_2.py

```
from __future__ import print_function

from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.datasets import imdb

py_filename = 'imdb_lstm_2.py'
max_features = 20000
maxlen = 80  # cut texts after this number of words (among top max_features
most common words)
batch_size = 32
epochs = 20 #60

#%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

import codecs
import datetime
import timeit
timer_start = timeit.default_timer()
#timer_end = timeit.default_timer()
#print('timer_end - timer_start', timer_end - timer_start)
```

imdb_lstm_2.py

```
def getDateTimeNow():
    strnow = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    return strnow

def read_file_utf8(filename):
    with codecs.open(filename,'r',encoding='utf-8') as f:
        text = f.read()
    return text

def write_file_utf8(filename,text):
    with codecs.open(filename, 'w', encoding='utf-8') as f:
        f.write(text)
        f.close()

def log_file_utf8(filename, text):
    with codecs.open(filename, 'a', encoding='utf-8') as f:
        #append file
        f.write(text + '\n')
        f.close()

log_file_utf8("logfile.txt", '***** ' + py_filename + ' *****')
log_file_utf8("logfile.txt", '***** Start DateTime: ' + getDateTimeNow())

print('Start: ', datetime.datetime.now().strftime("%Y%m%d_%H%M%S"))
```

imdb_lstm_2.py

```
print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)

print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

imdb_lstm_2.py

```
print('Train...')
print('model.fit: ', datetime.datetime.now().strftime("%Y%m%d_%H%M%S"))
history = model.fit(x_train, y_train,
          batch_size = batch_size,
          epochs = epochs,
          validation_data = (x_test, y_test))

score, acc = model.evaluate(x_test, y_test,
                            batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)
```

imdb_lstm_2.py

```
timer_end = timeit.default_timer()
print('Timer: ', str(round(timer_end - timer_start, 2)), 's')
print('DateTime: ', datetime.datetime.now().strftime("%Y%m%d_%H%M%S"))
log_file_utf8("logfile.txt", 'Timer: ' + str(round(timer_end - timer_start, 2))
+ ' s')
log_file_utf8("logfile.txt", '***** End Datetime: ' +
datetime.datetime.now().strftime("%Y%m%d_%H%M%S"))

# summarize history for accuracy
#http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/
print('history.history.keys():', history.history.keys())
print('history.history:', history.history)
log_file_utf8("logfile.txt", 'history.history:' + str(history.history))
```

imdb_lstm_2.py

```
# Deep Learning Training Visualization
plt.figure(figsize=(10, 8))  # make separate figure
ax1 = plt.subplot(2, 1, 1)
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
ax1.xaxis.set_major_locator(plt.NullLocator())
#plt.xlabel('epoch')
plt.legend(['train acc', 'test val_acc'], loc='upper left')
#plt.show()
ax2 = plt.subplot(2, 1, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train loss', 'test val_loss'], loc='upper left')
plt.savefig("training_accuacy_loss_" + py_filename + "_" + str(epochs) +
".png", dpi= 300)
```

imdb_lstm_2.py

```
#Log File for Deep Learning Summary Analysis
log_file_utf8("logfile.txt", 'DL_Summary:\tpy_filename\t'  + py_filename +
    '\tepochs\t' + str(epochs) +
    '\tscore\t' + str(score) +
    '\taccuracy\t' + str(acc) +
    '\tTimer\t ' + str(round(timer_end - timer_start, 2)) +
    '\thistory\t' + str(history.history))
#plt.show()
```

python filename.py

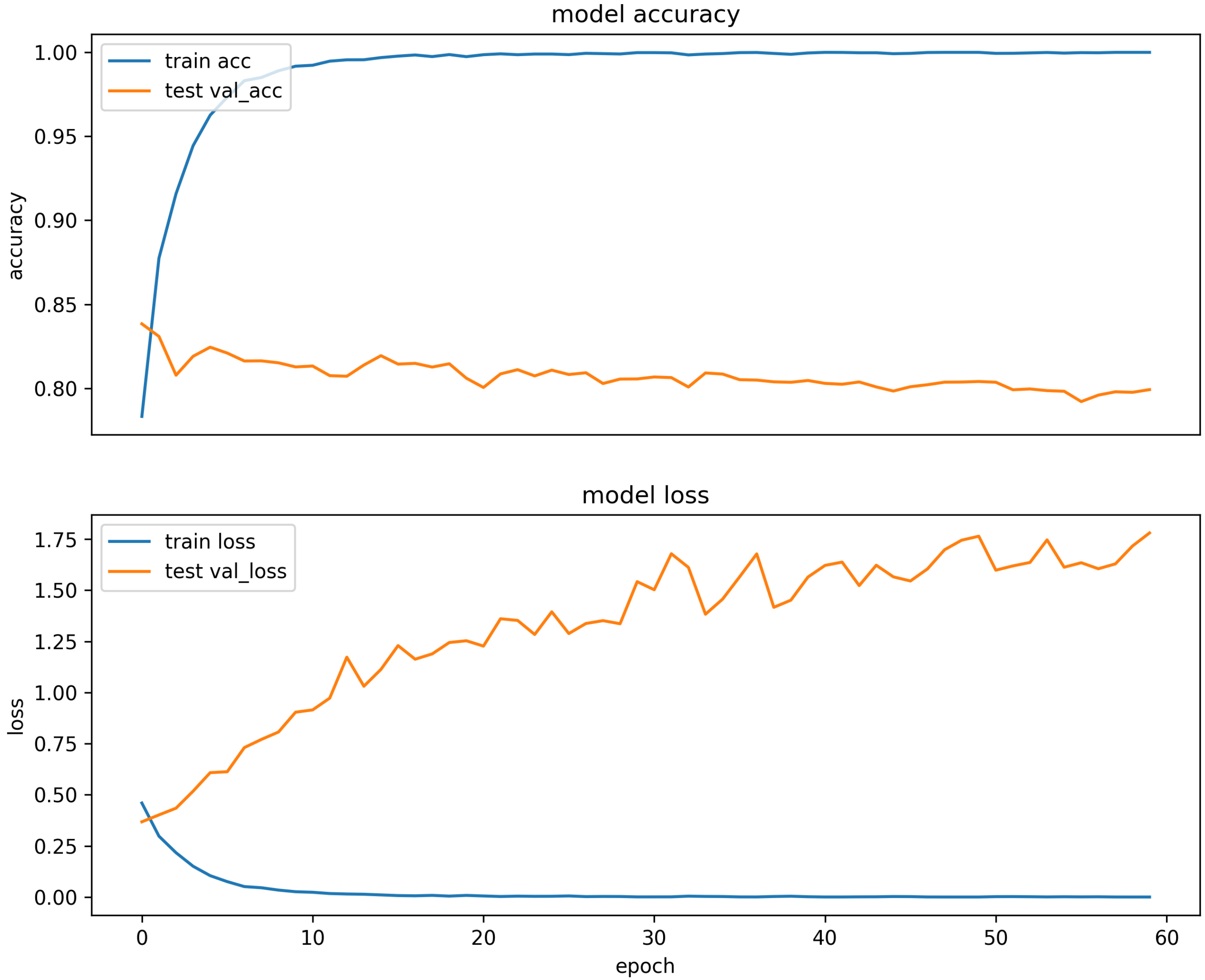
```
python imdb_fasttext_2.py
python imdb_cnn_2.py
python imdb_lstm_2.py
python imdb_cnn_lstm_2.py
python imdb_bidirectional_lstm_2.py
```

Deep Learning Summary

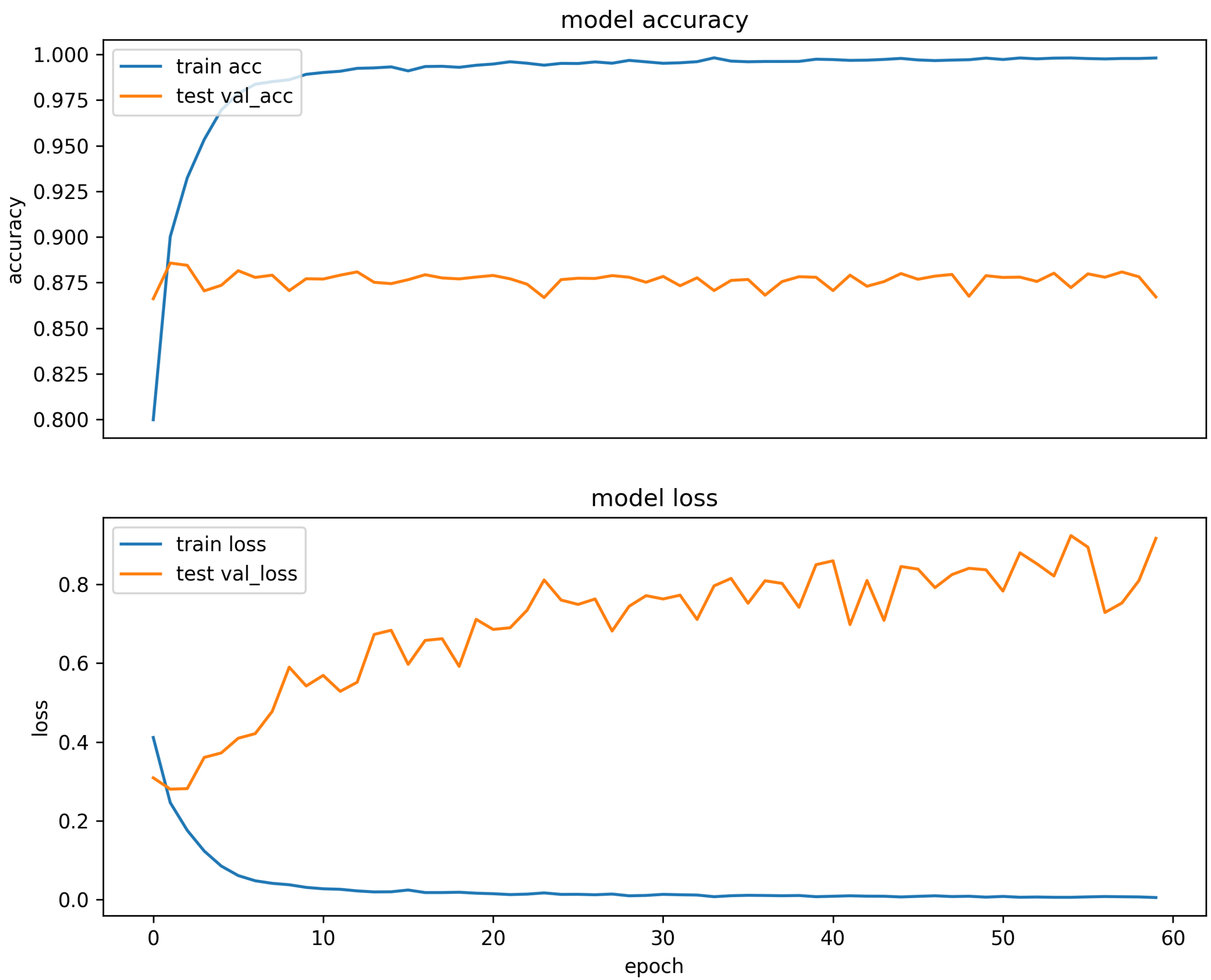
```
#Log File for Deep Learning Summary Analysis
log_file_utf8("logfile.txt", 'DL_Summary:\tpy_filename\t'  + py_filename +
    '\tepochs\t' + str(epochs) +
    '\tscore\t' + str(score) +
    '\taccuracy\t' + str(acc) +
    '\tTimer\t ' + str(round(timer_end - timer_start, 2)) +
    '\thistory\t' + str(history.history))
```

Model	epochs	Score	Accuracy	Timer (s)
imdb_lstm_2.py	30	0.6440	0.8540	**682.57**
imdb_cnn_2.py	30	0.7186	**0.8775**	4320.38
imdb_lstm_2.py	30	1.5716	0.8052	3958.93
imdb_cnn_lstm_2.py	30	1.3105	0.8240	2471.65
imdb_bidirectional_lstm_2.py	30	1.4083	0.8255	4344.36
imdb_fasttext_2.py	30	**0.6439**	0.8540	1117.78
imdb_fasttext_2.py	60	1.2335	0.8407	1297.02
imdb_cnn_2.py	60	0.9170	0.8672	8507.48
imdb_lstm_2.py	60	1.7803	0.7992	8039.67
imdb_cnn_lstm_2.py	60	1.4623	0.8137	4912.25
imdb_bidirectional_lstm_2.py	60	1.8975	0.8138	8589.17

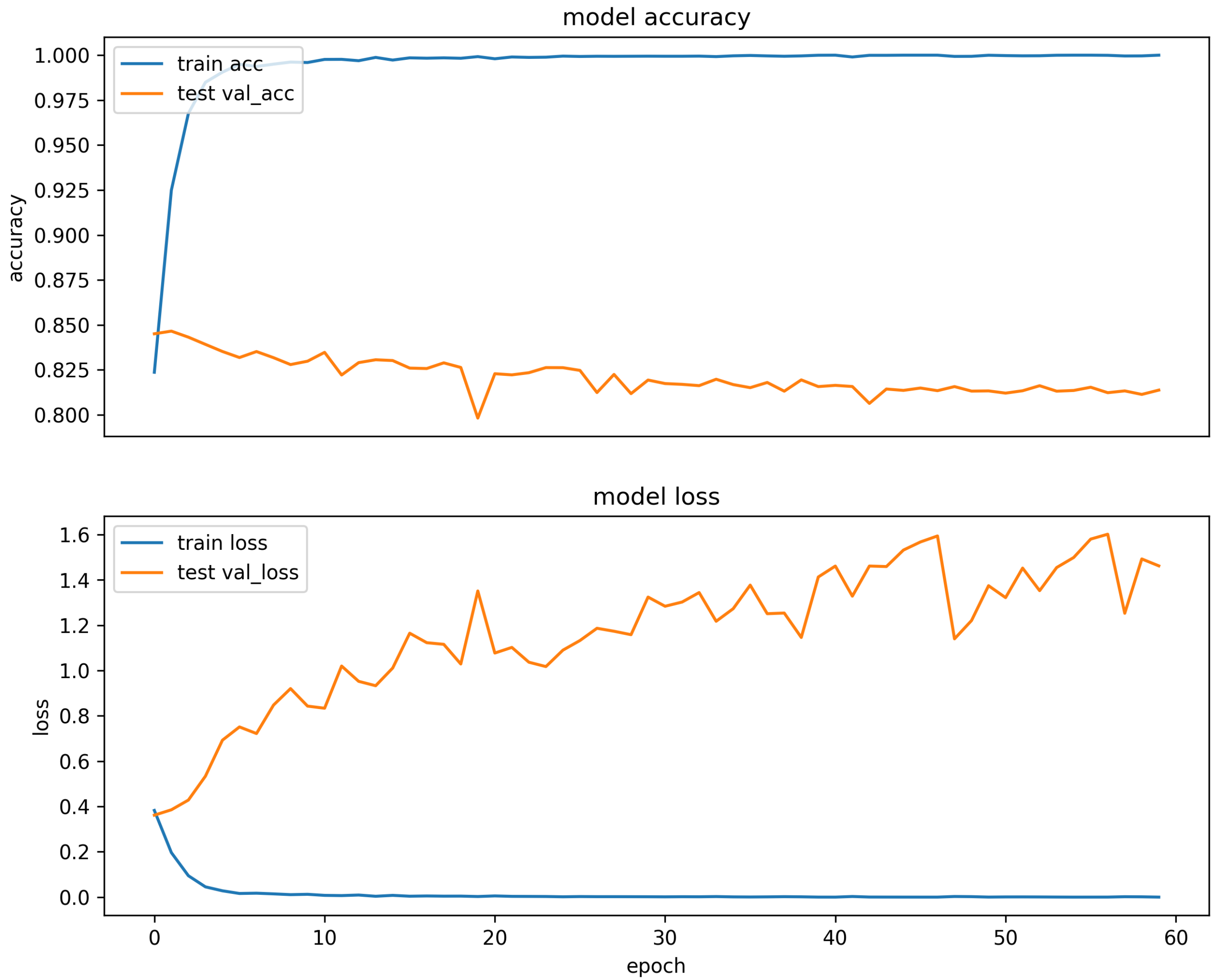
imdb_lstm_2.py



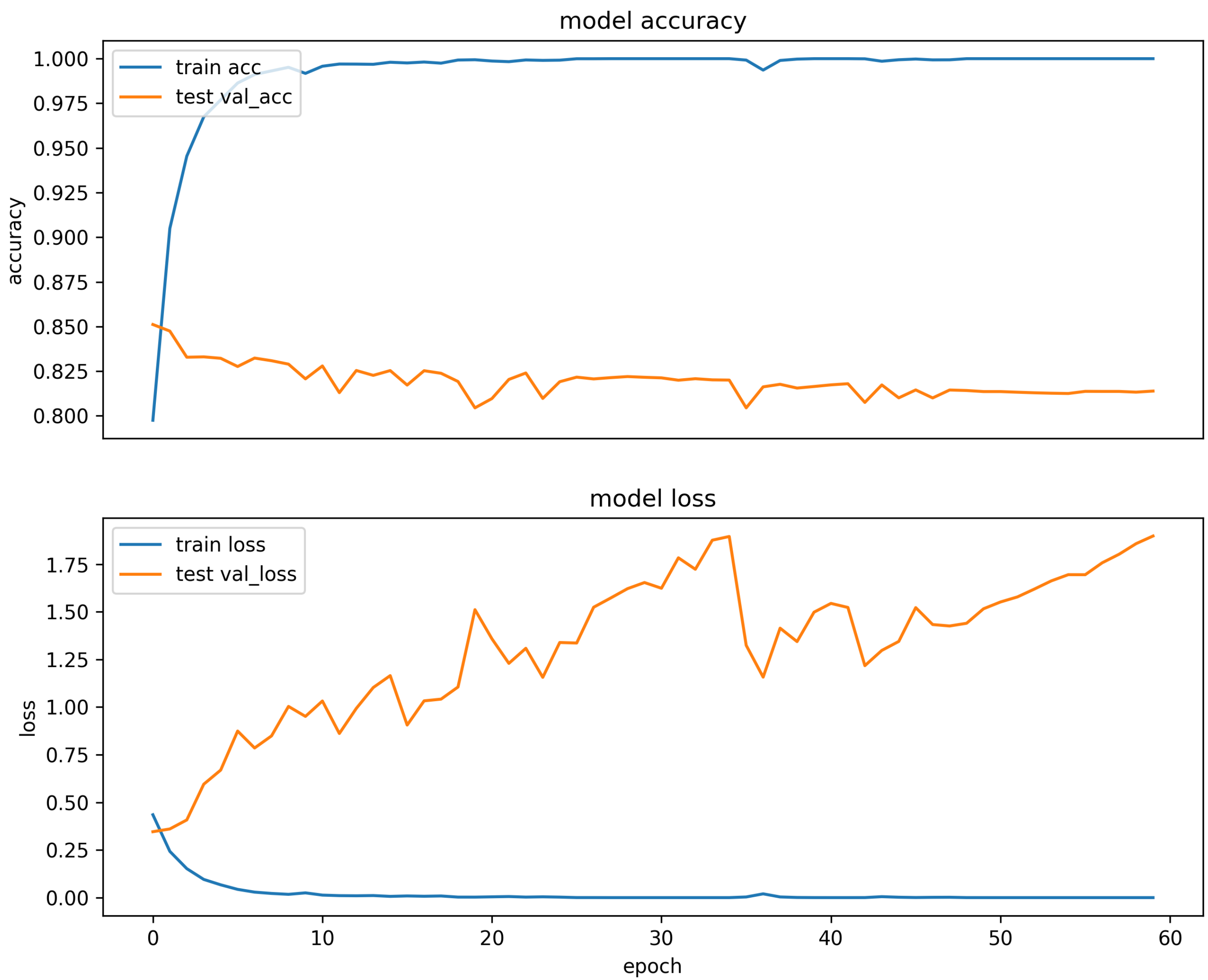
imdb_cnn_2.py


model accuracy
train acc
test val_acc
accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
model loss
train loss
test val_loss
loss
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
epoch

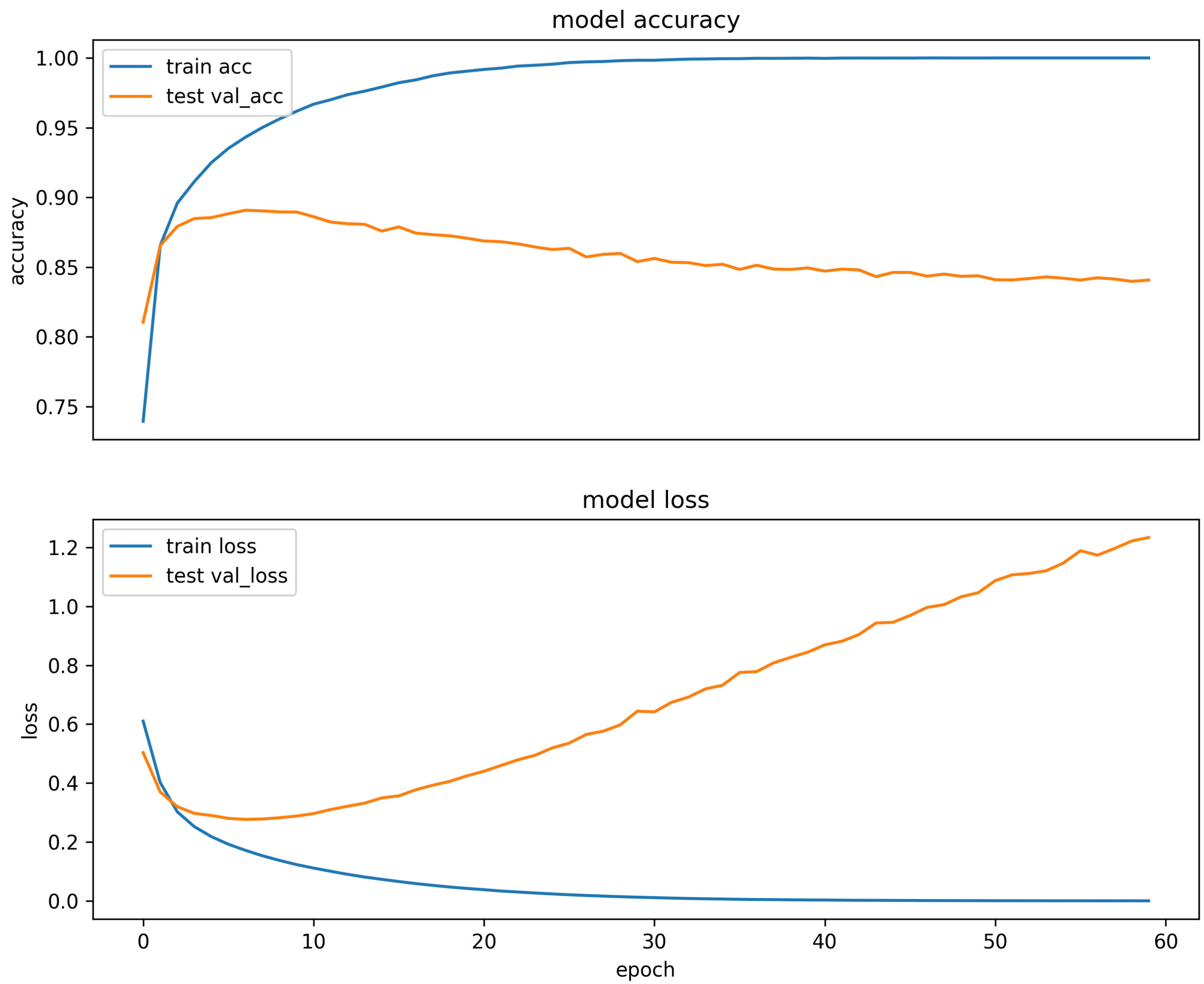
imdb_cnn_lstm_2.py



imdb_bidirectional_lstm_2.py



imdb_fasttext_2.py



References

- Bing Liu (2015), Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press
- Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” 2nd Edition, Springer. http://www.cs.uic.edu/~liub/WebMiningBook.html
- Bing Liu (2013), Opinion Spam Detection: Detecting Fake Reviews and Reviewers, http://www.cs.uic.edu/~liub/FBS/fake-reviews.html
- Bo Pang and Lillian Lee (2008), "Opinion mining and sentiment analysis,” Foundations and Trends in Information Retrieval 2(1-2), pp. 1–135, 2008.
- Wiltrud Kessler (2012), Introduction to Sentiment Analysis, http://www.ims.uni-stuttgart.de/~kesslewd/lehre/sentimentanalysis12s/introduction_sentimentanalysis.pdf
- Z. Zhang, X. Li, and Y. Chen (2012), "Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews," ACM Trans. Manage. Inf. Syst. (3:1) 2012, pp 1-23.
- Yongjun Zhu, Meen Chul Kim, and Chaomei Chen (2017), "An investigation of the intellectual structure of opinion mining research." Information Research 22, no. 1 (2017).
- Erik Cambria (2016), "Affective computing and sentiment analysis." IEEE Intelligent Systems 31, no. 2 (2016): 102-107.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)

References

- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts (2013), "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," In Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631, p. 1642 http://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf
- Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.
- Vishal Kharde and Sheetal Sonawane (2016), "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, vol 139, no. 11, 2016. pp.5-15.
- Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.
- Steven Struhl (2015), Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence (Marketing Science), Kogan Page
- Keras: http://keras.io/
- TensorFlow: https://www.tensorflow.org/