Neural Network for Sentiment Analaysis

a Tutorial at EMNLP 2016

Yue Zhang and Duy Tin Vo
Singapore University of Technology and Design

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Outline

- **Introduction**
  - **Definition**
  - Benchmarks
  - Lexicons
  - Machine learning background
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Definition

- Given a set of data $\mathcal{D}$: $x^{(i)}$ $(i \le N)$ with label $y^{(i)}$ $(y^{(i)} \le \mathcal{C})$, sentiment analysis task can be deemed as a classification task
- Extract subjectivity and sentiment polarity from text data

This is an awesome movie ☺!!!

$$S = \{s_1, s_2, \dots, s_n\}$$
$$s_i = \{w_1, w_2, \dots, w_m\}$$

Chelsea beats **MU** ☹ !?!

$$T = \{t_1, t_2, \dots, t_n\}, t \in s_i$$

Definition

- Document Level Sentiment
  - Input

$$D = \{d_1, d_2, \dots, d_n\}$$

  - Where:
    - $d_i = \{s_1, s_2, \dots, s_m\}$
  - Output

$$senti = \{pos, neg[, neu]\}$$

> This film has everything in it from a jail break, crooked southern politicians, muses, references to what I can only assume are historical figures, riverside baptisms, bank robberies, violence towards animals, singing flocks of religious fanatics, KKK, lynch mobs and so on. There are obviously many references to Homer's Odyssey in here as well, but I wouldn't know that because I have never read Homer's Odyssey or even knew one thing about it. Every other newspaper reviewer seems to know all about it and they think that this cynicism and almost spoof-like quality towards it makes the film that much better. Well coming from a guy who doesn't know anything about it, I can tell you that it is still an entertaining film. There were times when again, as is usual for a Coen film, I wasn't sure why I was entertained or laughing, but I was.
>
> This is a road picture where three men travel along the way to find a hidden treasure that Clooney says he has hidden to his two other cell mates. He has to take them along because they were also chained to him when they had their chance to escape.
>
> I like all the principal actors in the film and many of them are Coen cronies. It was nice to see Goodman again. It was nice to see Hunter and especially Turturro who seems to have a place in every Coen film. It's too bad they didn't find a place for Steve Buscemi but that is a different story all together. But back to Clooney. The man just has charisma. He is a one hell of an actor as well and here he is not quite as zany as the others but even he has his own idiosyncrasies. His work here is quite awesome and I really hope this shows that he is capable of playing any range of character.
>
> Now after heaping all this praise on the film, let me just say this as well. I didn't really enjoy the film at first. I found it to be quite tedious and a little boring. There were too many ideas in here and not enough care went into harnessing them for all what they were worth. But then the film began to grow on me. It took a while but it did grow on me. I don't think this is their best film, but it is still a good one and I am giving it a 8.5. But the reason that I do recommend this film is for one reason only.
>
> Every day you can go look into the paper and look at the films that are playing and say to yourself, seen it, seen it, oh, seen it last year, that is the same as this film and that is the same as that film. Most films have been recycled in some form or another. Not the Coen's films. They have not been recycled and if they have I don't know about it. That is reason enough to see something that they put out. Originality counts for a lot in my books. The Coens are original and they are good. And that is not common in todays cinema. Enjoy them while they are allowed to make films. Because you don't get vision like this in many films, so when you do, enjoy it!

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In Proceedings of ACL:HLT, 142-150.

Definition

- Sentence Level Sentiment
  - Input

$$S = \{s_1, s_2, \dots, s_n\}$$

    - Where:
      - $s_i = \{w_1, w_2, \dots, w_m\}$
  - Output

$$senti = \{pos, neg[, neu]\}$$

I like all the principal actors in the film and many of them are Coen cronies.

Bo Pang and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of* ACL, 115-124.

Definition

- Fine-grained Sentiment
  - Sentiment on target
  - Opinion expression
  - Opinion holder
  - Opinion strength
  - Etc.

It was nice to see **Goodman** again.

I really love Leicester City!! Fantastic!!!

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-Markov conditional random fields. In *Proceedings of* EMNLP:CoNLL, 1335-1345.

Outline

- **Introduction**
  - Definition
  - **Benchmarks**
  - Lexicons
  - Machine learning background
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Benchmarks

- Movie reviews
  - Pang and Lee (2004)
    - Subjectivity vs Objectivity sentences
    - Positive vs Negative document

Sentence-level		
subjective	objective	total
5000	5000	10000
Document-level		
positive	negative	total
1000	1000	2000

Subjective:
works both as an engaging drama and an incisive look at the difficulties facing native americans .

Positive:
kolya is one of the richest films i've seen in some time . zdenek sverak plays a confirmed old bachelor (who's likely to remain so) , who finds his life as a czech cellist increasingly impacted by the five-year old boy that he's taking care of .
though it ends rather abruptly-- and i'm whining , 'cause i wanted to spend more time with these characters-- the acting , writing , and production values are as high as , if not higher than , comparable american dramas .
this father-and-son delight-- sverak also wrote the script , while his son , jan , directed-- won a golden globe for best foreign language film and , a couple days after i saw it , walked away an oscar .in czech and russian , with english subtitles .

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of* ACL.

Benchmarks

- Movie reviews
  - Pang and Lee (2005)
  - Sentence-level

Sentence-level		
positive	negative	total
5331	5331	10662

Positive:
an idealistic love story that brings out the latent 15-year-old romantic in everyone .

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, 115-124.

Benchmarks

- Movie reviews
  - Mass et al. (2011)
  - Document-level

	pos	neg	total
Train	12500	12500	25000
Test	12500	12500	25000
unsup			50000

Positive:

This film has everything in it from a jail break, crooked southern politicians, muses, references to what I can only assume are historical figures, riverside baptisms, bank robberies, violence towards animals, singing flocks of religious fanatics, KKK, lynch mobs and so on. There are obviously many references to Homer's Odyssey in here as well, but I wouldn't know that because I have never read Homer's Odyssey or even knew one thing about it. Every other newspaper reviewer seems to know all about it and they think that this cynicism and almost spoof-like quality towards it makes the film that much better. Well coming from a guy who doesn't know anything about it, I can tell you that it is still an entertaining film. There were times when again, as is usual for a Coen film, I wasn't sure why I was entertained or laughing, but I was.

This is a road picture where three men travel along the way to find a hidden treasure that Clooney says he has hidden to his two other cell mates. He has to take them along because they were also chained to him when they had their chance to escape.

I like all the principal actors in the film and many of them are Coen cronies. It was nice to see Goodman again. It was nice to see Hunter and especially Turturro who seems to have a place in every Coen film. It's too bad they didn't find a place for Steve Buscemi but that is a different story all together. But back to Clooney. The man just has charisma. He is a one hell of an actor as well and here he is not quite as zany as the others but even he has his own idiosyncrasies. His work here is quite awesome and I really hope this shows that he is capable of playing any range of character.

Now after heaping all this praise on the film, let me just say this as well. I didn't really enjoy the film at first. I found it to be quite tedious and a little boring. There were too many ideas in here and not enough care went into harnessing them for all what they were worth. But then the film began to grow on me. It took a while but it did grow on me. I don't think this is their best film, but it is still a good one and I am giving it a 8.5. But the reason that I do recommend this film is for one reason only.

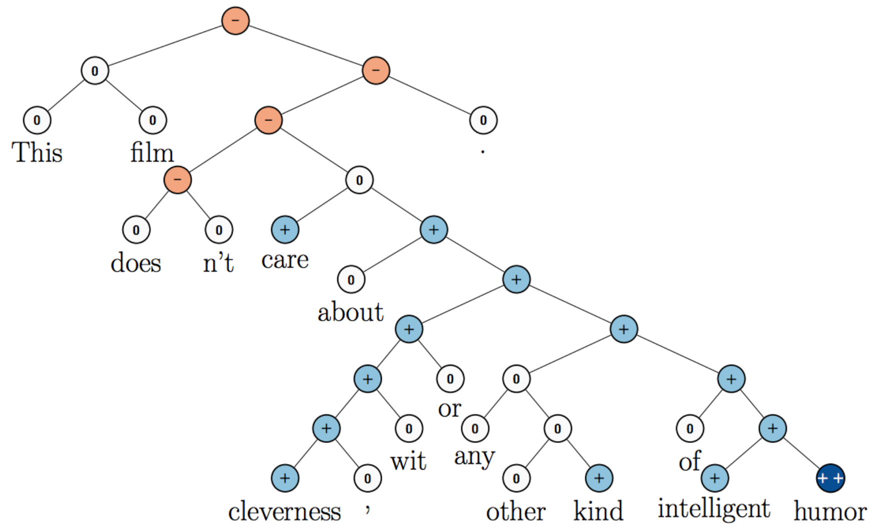
Every day you can go look into the paper and look at the films that are playing and say to yourself, seen it, seen it, oh, seen it last year, that is the same as this film and that is the same as that film. Most films have been recycled in some form or another. Not the Coen's films. They have not been recycled and if they have I don't know about it. That is reason enough to see something that they put out. Originality counts for a lot in my books. The Coens are original and they are good. And that is not common in todays cinema. Enjoy them while they are allowed to make films. Because you don't get vision like this in many films, so when you do, enjoy it!

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL:*HLT, 142-150.

Benchmarks

- Movie reviews
  - Socher et al. (2013), which is induced from Pang and Lee (2005)
  - Phrase-level

	Train	**Valid**	**Test**
Binary	6920	872	1821
Fine-grained	8544	1101	2210



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 EMNLP*, 1631-1642.

Bo Pang and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, 115-124.

Benchmarks

- Product reviews
  - Hu and Liu (2004): 5 products
  - Ding et al (2008): 9 products, which is induced from Hu and Liu (2004)
  - Fine-grained

[t]
feature[+2]##just received this camera two days ago and already love the features it has .
photo[+2]##takes excellent photos .
night mode[+2]##night mode is clear as day .
use[+1][u]##i have not played with all the features yet , but the camera is easy to use once you get used to it .
viewfinder[-1]##the only drawback is the viewfinder is slightly blocked by the lens .
##however , using the lcd seems to eliminate this minor problem .
camera[+3]##overall it is the best camera on the market .
##i give it 10 stars !

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD* KDD, 168-177.
Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of* WSDM, 231-240.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD* KDD, 168-177.

Benchmarks

- Twitter
  - Go et. al. (2009)
  - Sentence-level

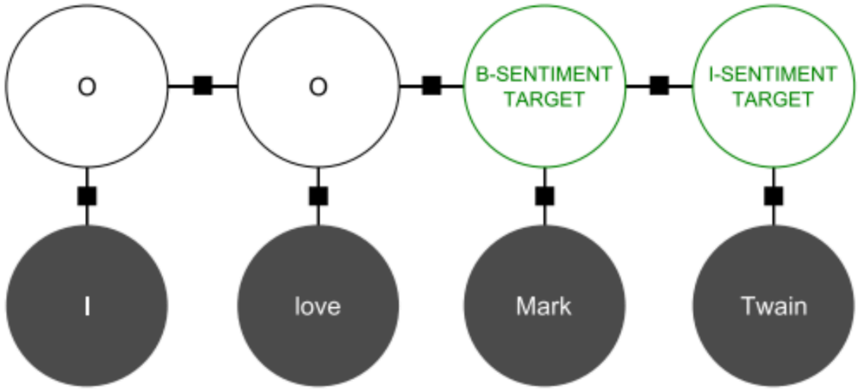
	pos	neg	total
Train	800k	800k	1.6m
Test	182	177	359

Positive: how can you not love Obama? he makes jokes about himself.
Negative: Naive Bayes using EM for Text Classification. Really Frustrating...

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 12.

Benchmarks

- Twitter
  - Mitchell et. al. (2013)
  - Open domain



Domain	pos	neg	neu	#Sent	#Entities
English	707	275	2,306	2,350	3,288
Spanish	1,555	1,007	4,096	5,145	6,658

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In Proceedings of EMNLP, 1643–1654.

Benchmarks

- Twitter
  - Dong et. al. (2014)
  - Targeted

	pos	neg	neu	total
Train	1561	1560	3127	6248
Test	173	173	346	692

Neutral:
i hate that i haven't had time for #zbrush in the past two days… we need #zspheres on the **[iphone]** so i can still sculpt on the go.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In Proceedings of ACL, 49-51.

Benchmarks

- Twitter
  - SemEval13 (Nakov et. al., 2013)
  - Sentence-level

	pos	neg	neu	total
Train	3662	1466	4600	9729
Valid	575	340	739	1654
Test	1573	601	1640	3814

Positive: OMG Saturday at 8, p.s. I love you premieres on abc family.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In Proceddings of SemEval, 312–320.

Outline

- **Introduction**
  - Introduction
  - Benchmarks
  - **Lexicons**
  - Machine learning background
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Lexicons

- Manual methods
  - MPQA lexicon (Wilson et. al., 2005) contains 8222 words

	Strength	Length	Word	Part-of-speech	Stemmed	Polarity
1.	type=weaksubj	len=1	word1=abandoned	pos1=adj	stemmed1=n	priorpolarity=negative
2.	type=weaksubj	len=1	word1=abandonment	pos1=noun	stemmed1=n	priorpolarity=negative
3.	type=weaksubj	len=1	word1=abandon	pos1=verb	stemmed1=y	priorpolarity=negative
4.	type=strongsubj	len=1	word1=abase	pos1=verb	stemmed1=y	priorpolarity=negative
5.	type=strongsubj	len=1	word1=abasement	pos1=anypos	stemmed1=y	priorpolarity=negative
6.	type=strongsubj	len=1	word1=abash	pos1=verb	stemmed1=y	priorpolarity=negative
7.	type=weaksubj	len=1	word1=abate	pos1=verb	stemmed1=y	priorpolarity=negative
8.	type=weaksubj	len=1	word1=abdicate	pos1=verb	stemmed1=y	priorpolarity=negative
9.	type=strongsubj	len=1	word1=aberration	pos1=adj	stemmed1=n	priorpolarity=negative
10.	type=strongsubj	len=1	word1=aberration	pos1=noun	stemmed1=n	priorpolarity=negative
...						
8221.	type=strongsubj	len=1	word1=zest	pos1=noun	stemmed1=n	priorpolarity=positive

Source: http://sentiment.christopherpotts.net/lexicons.html#mpqa

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of* HLT:EMNLP, 347-354.

Lexicons

- Manual methods
  - Hu and Liu (2004) lexicon contains 2006 positive words and 4783 negative words.

positive	negative
a+	2-faced
abound	2-faces
abounds	abnormal
abundance	abolish
abundant	abominable
access	abominably
able	abominate
accessible	abomination
acclaim	abort
acclaimed	aborted

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD* KDD, 168-177.

Lexicons

- Manual methods
  - Mohammad and Turney (2013) Lexicon contains 14182 words with 10 labels (8 emoticons and 2 sentiments)

hate	anger	1	hateful	anger	1
hate	anticipation	0	hateful	anticipation	0
hate	disgust	1	hateful	disgust	1
hate	fear	1	hateful	fear	1
hate	joy	0	hateful	joy	0
hate	negative	1	hateful	negative	1
hate	positive	0	hateful	positive	0
hate	sadness	1	hateful	sadness	1
hate	surprise	0	hateful	surprise	0
hate	trust	0	hateful	trust	0

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. In *Proceedings of NAACL:HLT 2010 Workshop on* CAAGET, 26-34.

Lexicons

- Automatic methods
  - SentiWordNet (Esuli and Fabrizio, 2006) learns positive and negative sentiment scores for synsets in WordNet

POS	ID	PosScore	NegScore	SynsetTerms	Gloss
a	00001740	0.125	0	able#1	(usually followed by `to') having the necessary means or [...]
a	00002098	0	0.75	unable#1	(usually followed by `to') not having the necessary means or [...]
a	00002312	0	0	dorsal#2 abaxial#1	facing away from the axis of an organ or organism; [...]
a	00002527	0	0	ventral#2 adaxial#1	nearest to or facing toward the axis of an organ or organism; [...]
a	00002730	0	0	acroscopic#1	facing or on the side toward the apex
a	00002843	0	0	basiscopic#1	facing or on the side toward the base
a	00002956	0	0	abducting#1 abducent#1	especially of muscles; [...]
a	00003131	0	0	adductive#1 adducting#1 adducent#1	especially of muscles; [...]
a	00003356	0	0	nascent#1	being born or beginning; [...]
a	00003553	0	0	emerging#2 emergent#2	coming into existence; [...]

Source: http://sentiment.christopherpotts.net/lexicons.html#sentiwordnet

Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, 417-422.

Lexicons

- Automatic methods
  - Tang et. al. (2014) consists of 178,781 positive words/phrases and 168,845 negative words/phrases

follow me ... but	-0.592651
#society	-0.592650
i can't view	-0.592650
producer's	-0.592646
now , i'm	-0.592637
#although	-0.592631
twitter like	-0.592629
a wizard	-0.592627

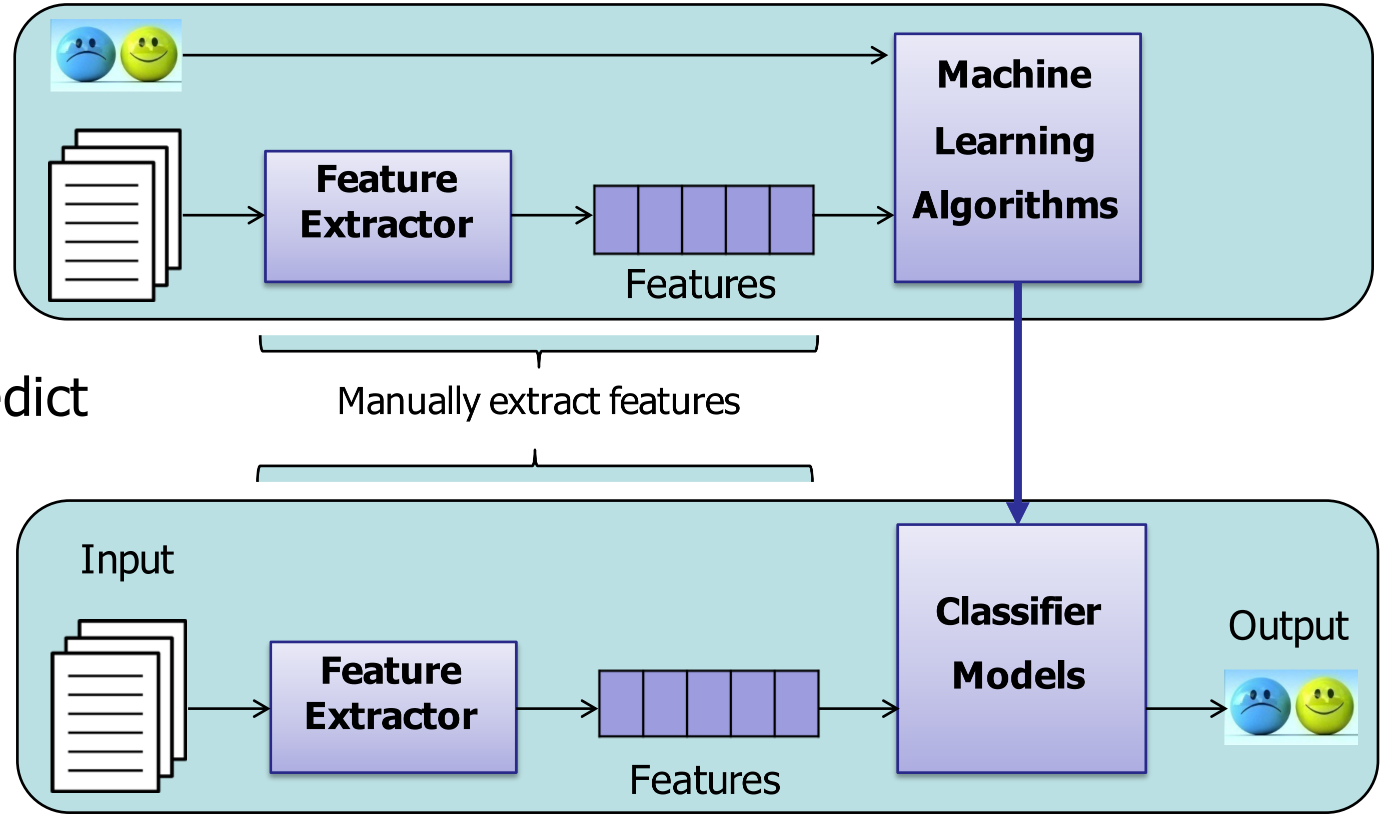
Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In Proceedings of *COLING*, 172-182.

Outline

- **Introduction**
  - Introduction
  - Benchmarks
  - Lexicons
  - **Machine learning background**
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Machine Learning Background

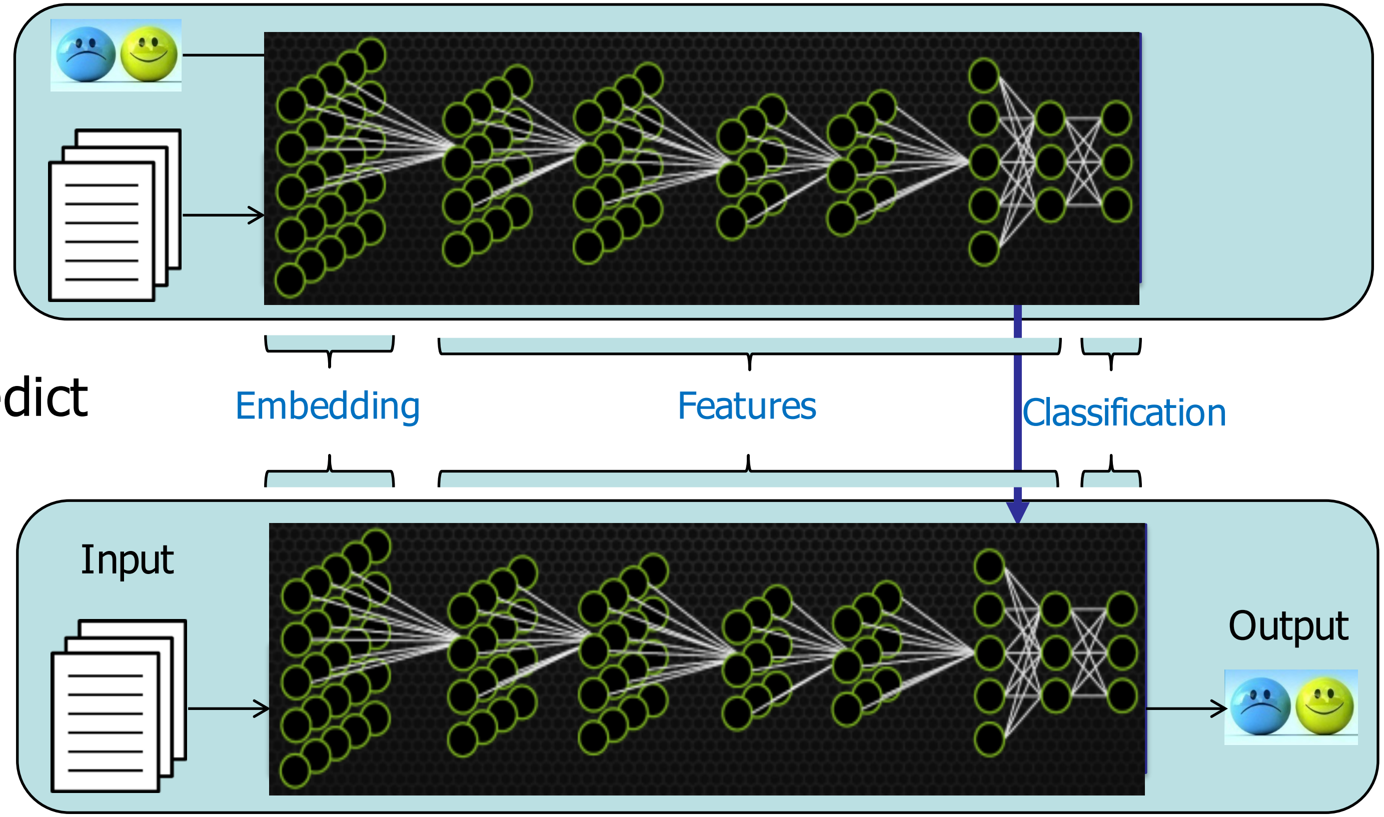
- General model:
  - Train
  - Predict



- One-hot vector
- N-grams
- Brown Clustering
- Lexicons
- Patterns
- POS
- …

Machine Learning Background

- Neural Network: a sub-area of machine learning
  - Train
  - Predict

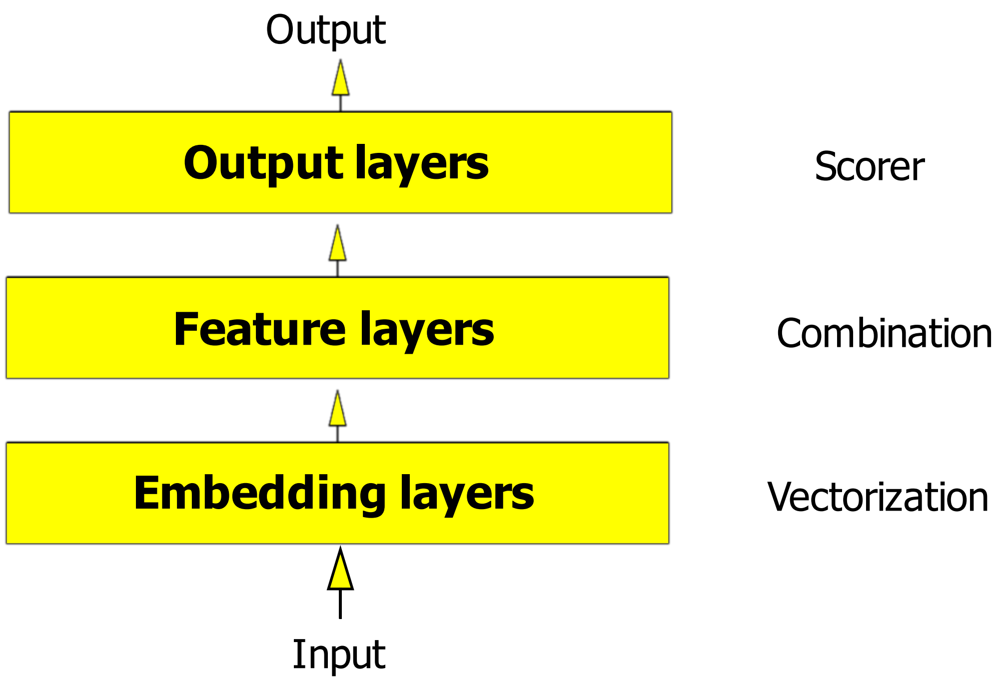


Outline

- Introduction and Background
- **Neural Network Background**
  - **Overview**
  - Typical Feature Layers
  - Training
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Overview

- General model:

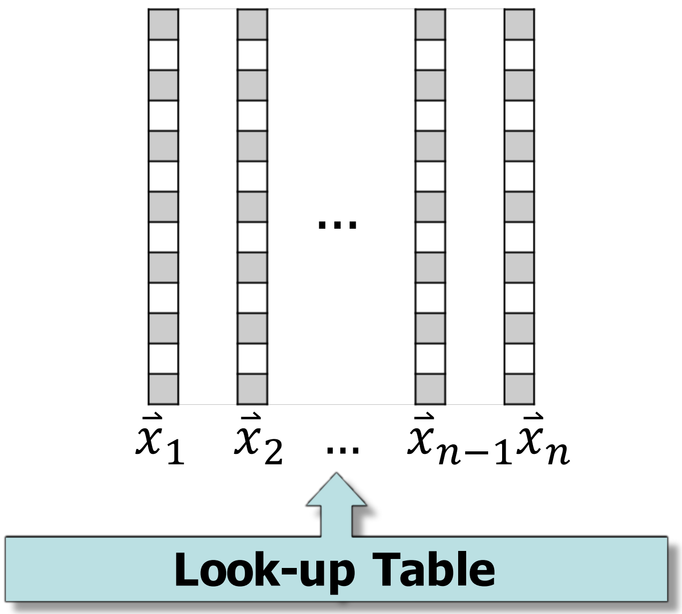


Overview

- Embedding Layer
  - Word to vector
  - Look up table

$$\vec{x}_i = \boldsymbol{W}_{|V|} \times \vec{I}_i$$

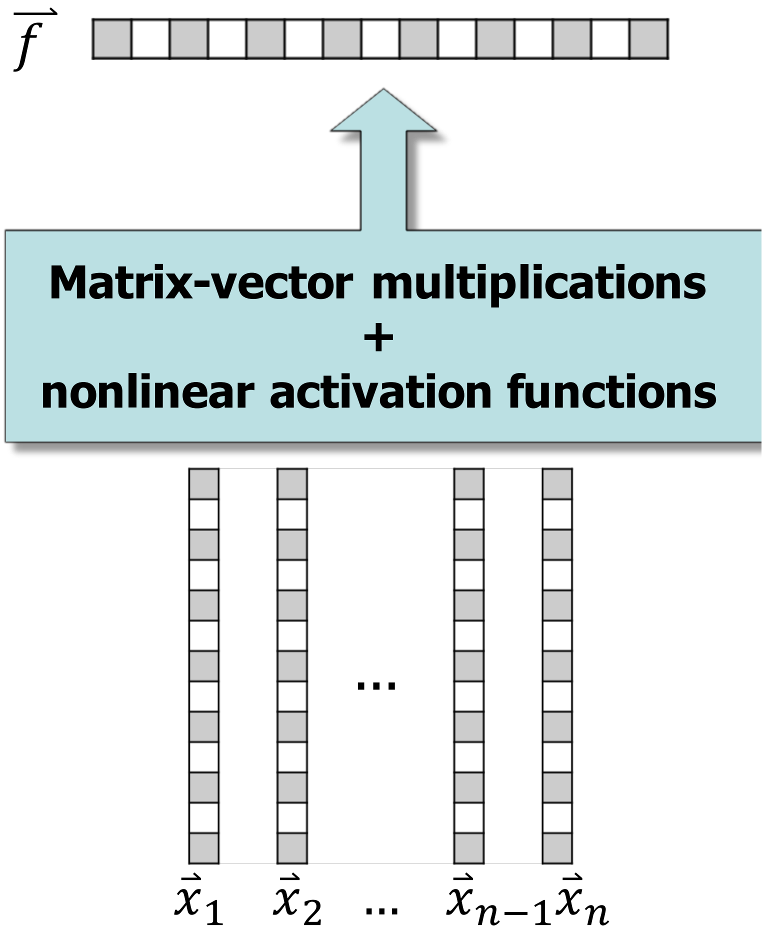
  - Where:
    - $\vec{x}_i \epsilon R^d$: word embedding
    - $\boldsymbol{W}_{|V|} \epsilon R^{|V| \times d}$: embedding matrix
    - $\vec{I}_i \epsilon R^{|V|}$: one-hot vector of word w_i
    - d: embedding dimension



$$s = w_1 \; w_2 \; \dots \; w_{n-1} w_n$$

Overview

- Feature Layer
  - Automatically learn the representation of inputs
  - Matrix-vector multiplication
  - Element-wise composition
  - Non-linear transformation

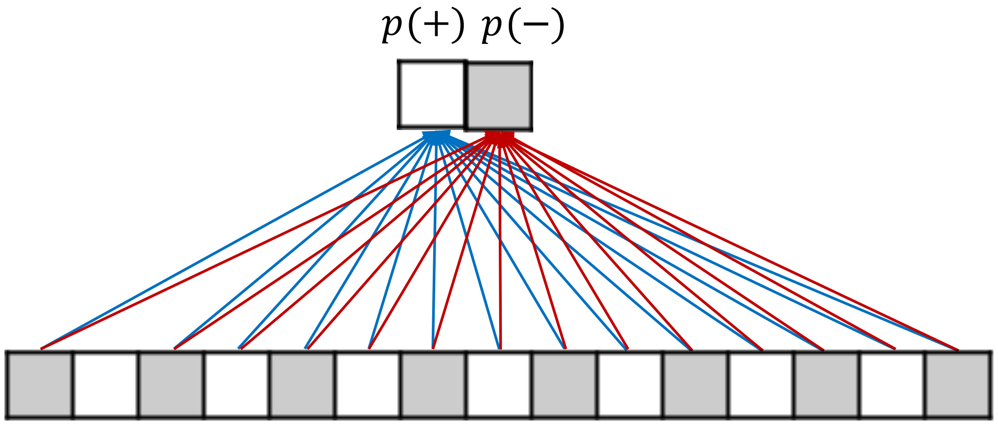


Overview

- Output Layer
  - *Margin output:* $f_{score} = \boldsymbol{W}_O \vec{f} + \vec{b}_O$
  - *Probability output*

$$O_c^{(i)} = P\left(Y = c \middle| x^{(i)}, \theta\right) = softmax_c(f_{score}) = \frac{e^{\vec{w}_c \vec{f} + b_c}}{\sum_{c'} e^{\vec{w}_{c'} \vec{f} + b_{c'}}}$$

  - Predicted label: $\bar{y}^{(i)} = argmax(O^{(i)})$
  - *Where:*
    - θ*: set of parameters*
    - $\boldsymbol{W}_O$, $\vec{b}_O$*: weight and bias parameters of output layer*



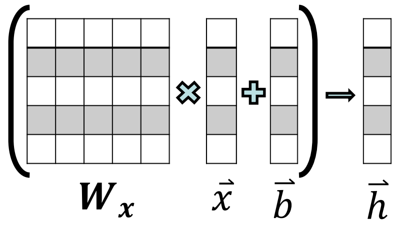
Outline

- Introduction
- **Neural Network Background**
  - Overview
  - **Typical Feature Layers**
  - Training
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Typical Feature Layers

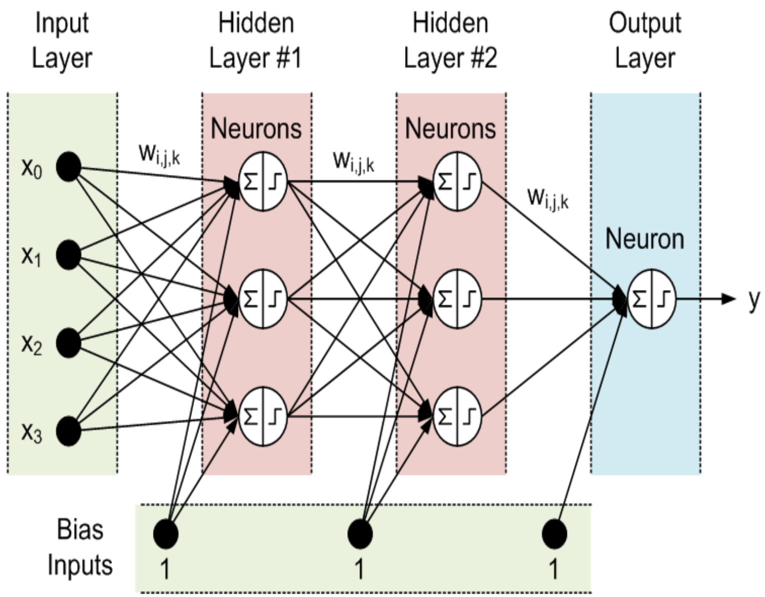
❖ Feed Forward (MLP)

$$\vec{h}_i = f(\boldsymbol{W}_x \vec{x}_i + \vec{b}_x)$$



– Where:

- $\vec{h}_i$: hidden features
- $f(z)$: activation function
- $\boldsymbol{W}_x$, $\vec{b}_x$: *weight and bias parameters of MLP*
- $\vec{x}_i$: input vector



Source: https://www.mql5.com/pt/code/9002

Typical Feature Layers

- $Acitivation\, functions\, f(z)$
  - $sigmoid(z) = \frac{1}{1+e^{-z}}$
  - $tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
  - $softmax_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, \ldots, K$
  - $relu(z) = \max(0, z)$
  - $iden(z) = z$

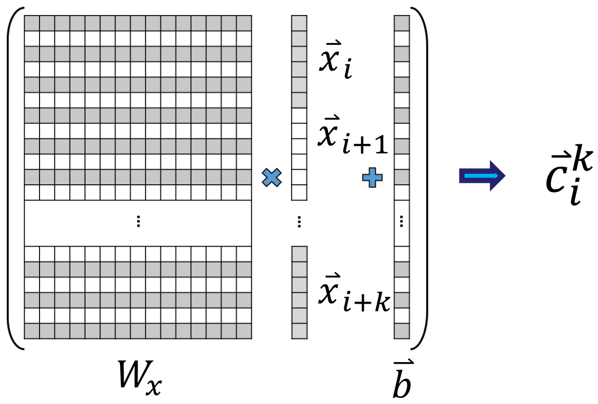
Typical Feature Layers

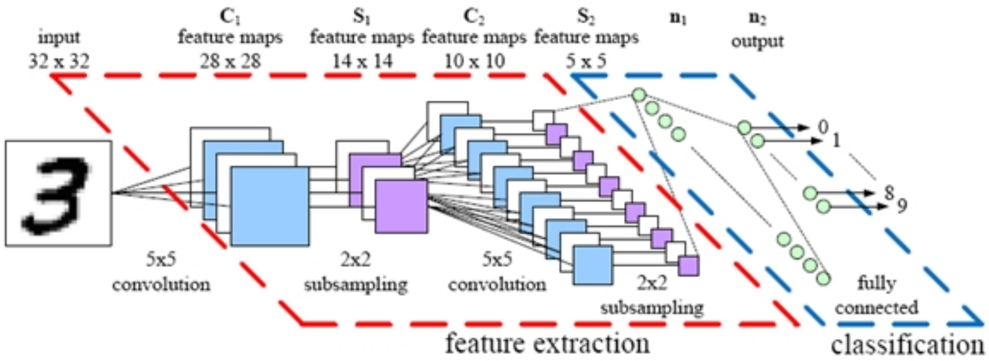
❖ Convolutional neural network (CNN)

$$\vec{c}_i^{\,k} = f(\boldsymbol{W}_c(\vec{x}_i \oplus \vec{x}_{i+1} \oplus \ldots \oplus \vec{x}_{i+k}) + \vec{b}_c)$$

– Where:

- $\vec{c}_i^{\,k}$: convolutional features
- $f(z)$: activation function
- $\boldsymbol{W}_c$, $\vec{b}_c$: *weight and bias parameters of CNN*
- $\vec{x}_i$: input vectors
- k: window size (2,3 in common)
- $\oplus$: concatenation





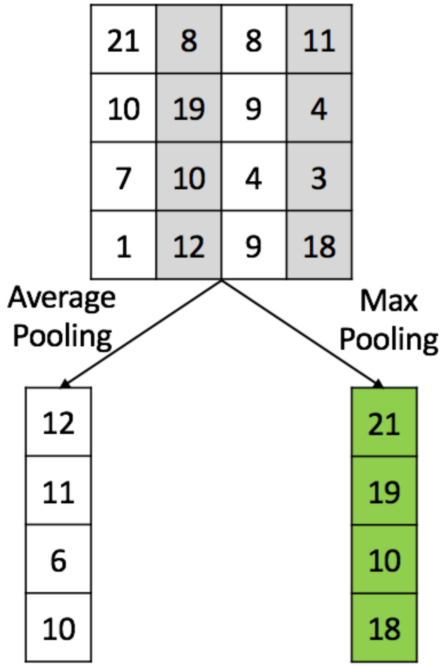
Source: http://parse.ele.tue.nl/education/cluster2

Typical Feature Layers

- Pooling

$$\vec{h}_i = pool(\boldsymbol{C}_i)$$

  - Where:
    - $\vec{h}_i$: hidden features
    - $pool$ is element-wise operations (max, average, min,...)
    - $\boldsymbol{C}_i$: input matrix

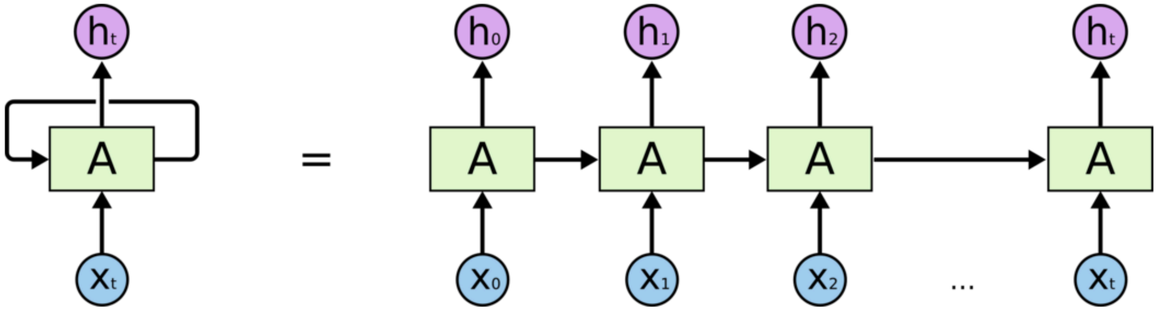


Typical Feature Layers

- Recurrent Neural Network (RNN)

$$\vec{h}_i = f(\boldsymbol{W}_h \vec{h}_{i-1} + \boldsymbol{W}_x \vec{x}_i + \vec{b}_x)$$

  - Where:
    - $\vec{h}_i$: hidden features at time i
    - $f(z)$: activation function
    - $\boldsymbol{W}_h, \boldsymbol{W}_x, \vec{b}_x$: *weight and bias parameters of RNN*
    - $\vec{x}_i$: input vector



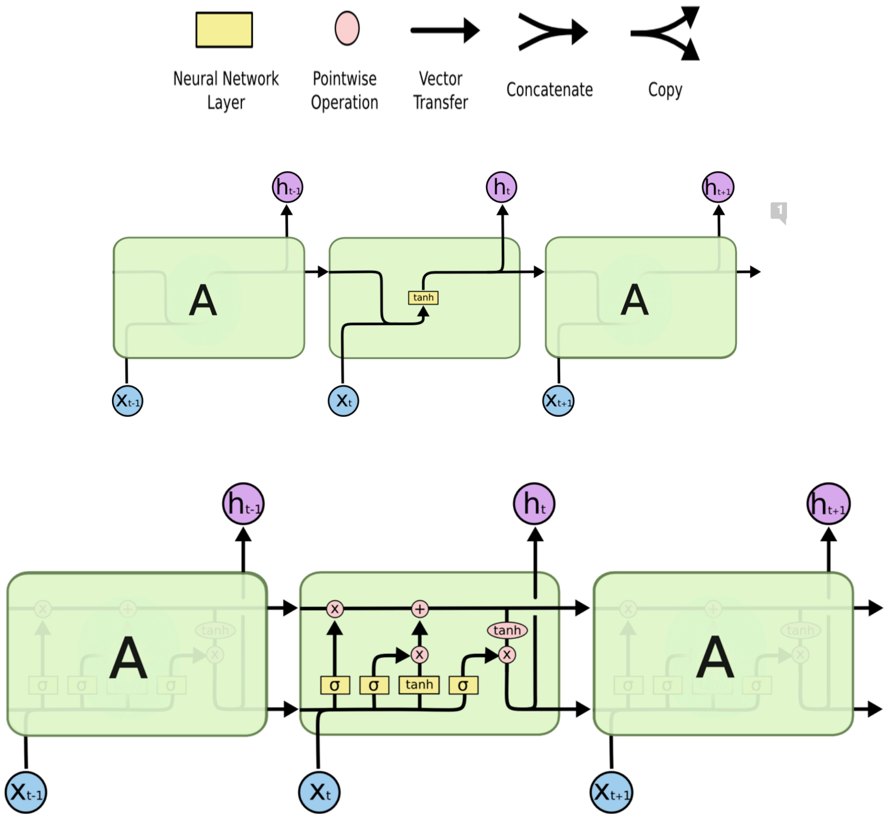
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Typical Feature Layers

- Long Short Term Memory (LSTM)

$$\vec{f}_t = \sigma(\boldsymbol{W}_f \vec{x}_t + \boldsymbol{U}_f \vec{h}_{t-1} + \vec{b}_f)$$
$$\vec{i}_t = \sigma(\boldsymbol{W}_i \vec{x}_t + \boldsymbol{U}_i \vec{h}_{t-1} + \vec{b}_i)$$
$$\vec{u}_t = tanh\left(\boldsymbol{W}_u \vec{x}_t + \boldsymbol{U}_u \vec{h}_{t-1} + \vec{b}_u\right)$$
$$\vec{c}_t = \vec{i}_t \odot \vec{u}_t + \vec{f}_t \odot \vec{c}_{t-1}$$
$$\vec{o}_t = \sigma(\boldsymbol{W}_o \vec{x}_t + \boldsymbol{U}_o \vec{h}_{t-1} + \vec{b}_o)$$
$$\vec{h}_t = \vec{o}_t \tanh \odot (\vec{c}_t)$$

  - Where:
    - $\vec{f}_t, \vec{i}_t, \vec{u}_t, \vec{c}_t, \vec{o}_t$: forget, input, update, control, output gate layers, respectively
    - $\boldsymbol{W}_*, \boldsymbol{U}_*, \vec{b}_*$: *weight and bias parameters of LSTM*



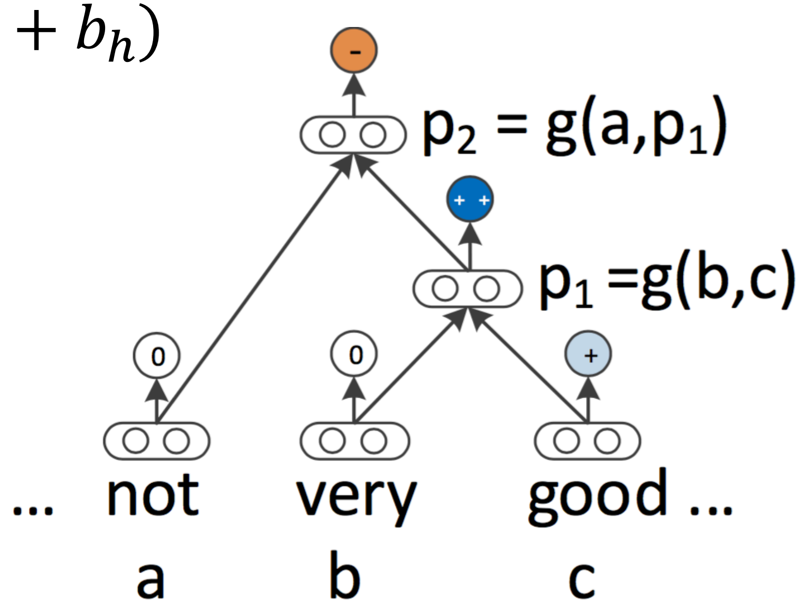
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Typical Feature Layers

- Recursive Neural Network (RecNN)

$$\vec{h}_i = f(\boldsymbol{W}_l \vec{h}_{i-1}^{l} + \boldsymbol{W}_r \vec{h}_{i-1}^{r} + \vec{b}_h)$$

  - Where:
    - $\vec{h}_i$: hidden features at time i
    - $f(z)$: activation function
    - $\boldsymbol{W}_l, \boldsymbol{W}_r, \vec{b}_h$: *weight and bias parameters of RecNN*



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 EMNLP*, 1631-1642.

Outline

- Introduction
- **Neural Network Background**
  - Overview
  - Typical Feature Layers
  - **Training**
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Training

- Supervised Learning
- Randomly initialized model
- Compare model output with manual reference

Training

- Loss functions
  - Cross Entropy Loss (Maximum Likelihood)

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i} p_i \log(q_i) = -\frac{1}{N}\sum_{i=1}^{N} I_{y^{(i)}} \log(O^{(i)})$$

    - Where:
      - θ: *set of parameters*
      - N: *number of samples*
      - $I_{y^{(i)}}$: one-hot vector corresponding to label $y^{(i)}$
      - $O^{(i)}$: *probability output of sample* $x^{(i)}$

Training

- Loss functions
  - Hinge loss (maximum-margin)
    - Binary classification:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y^{(i)} f_{score}{}^{(i)})$$

    - Multiclass classification

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \max(0, 1 + \max_{c' \neq c} f_{score\,c'} - f_{score\,c}))$$

    - Where:
      - θ: *set of parameters*
      - N: *number of samples*
      - $y^{(i)} \epsilon \{-1, 1\}$
      - f_{score}: margin output

Training

- Loss functions
  - 0/1 Loss (large margin)

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} I_{y_i \neq \hat{y}_i}$$

- Where:
  - θ*: set of parameters*
  - N*: number of samples*
  - I: indication function
  - y: ground-true labeled vector
  - $\hat{y}$: predicted vector

Training

- Loss functions
  - MSE Loss (regression)

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

    - Where:
      - θ*: set of parameters*
      - N*: number of samples*
      - y is a ground-true labeled vector
      - $\hat{y}$ is a predicted vector

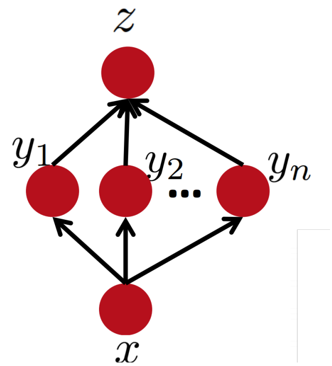
Training

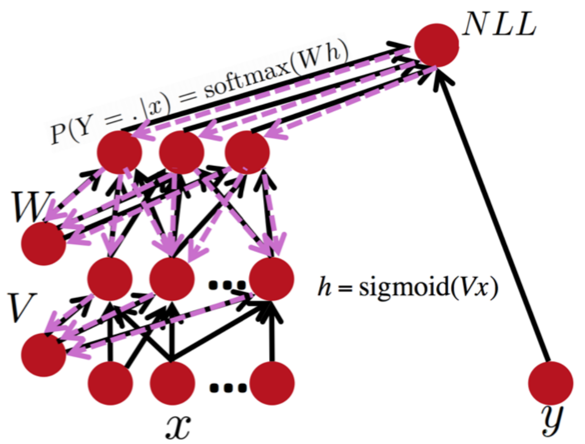
- Back Propagation
  - Goal
    - Find $\frac{\partial \mathcal{L}}{\partial \theta}$ for all parameters
    - Adjust parameters accordingly
  - Derivation
    - Chain Rule: if z= $f(y)$ and y= $g(x)$, then

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$$

    - Layer-wise calculation

$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n}\frac{\partial z}{\partial y_i}\frac{\partial y_i}{\partial x}$$





Training

- Batch gradient descent is an algorithm in which we repeatedly make small steps downward on an error surface defined by a loss function of a set of parameters over the full training set (N samples)

$$\theta^{k+1} = \theta^{k} - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

  - Where
    - *θ: set of parameters*
    - η: learning rate

➔ Problem: N is a very large number

Training

- SGD: Stochastic gradient descent works according to the same principles as batch gradient descent, but proceeds more quickly by estimating the gradient from just one example at a time instead of the entire training set

$$\theta^{k+1} = \theta^k - \eta \frac{\partial \mathcal{L}(\theta, x^{(i)}, y^{(i)})}{\partial \theta}$$

- Mini-batch SGD (MSGD) works identically to SGD, except that we use more than one training example to make each estimate of the gradient

$$\theta^{k+1} = \theta^k - \eta \frac{\partial \mathcal{L}\left(\theta, x^{(i:i+n)}, y^{(i:i+n)}\right)}{\partial \theta}$$

→ Problem: manually adjust learning rate

Training

- Momentum: helps to accelerate SGD in the relevant direction by adding a fraction γ of the update vector of the past time step to the current update vector

$$v_k = \gamma v_{k-1} - \eta \frac{\partial \mathcal{L}(\theta, x^{(i)}, y^{(i)})}{\partial \theta}$$
$$\theta^{k+1} = \theta^k - v_k$$

Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. In Proceedings of Neural Networks, 145-151.

Training

- AdaGrad: adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters

$$\theta^{k+1} = \theta^k - \eta^k g^k$$

  - Where:
    - g^k: the gradient of $\mathcal{L}$ w.r.t θ at k
    - $\eta^k = \frac{\eta}{\sqrt{\sum_{\tau=1}^{k} g_\tau^2 + \varepsilon}}$
    - ε: a smoothing term that avoids division by zero

➔ Problem: learning rate need to be initialized and gradually shrunk to an infinitesimally small number

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. In Proceeding of *The Journal of Machine Learning Research* 12, 2121-2159.

Training

- RMSprop*: adjusts the Adagrad method in a very simple way in an attempt to reduce its aggressive, monotonically decreasing learning rate. In particular, it uses a moving average of squared gradients instead

$$\theta^{k+1} = \theta^k + \Delta\theta^k,$$
$$\Delta\theta^k = -\frac{\eta}{RMS[g]_k} g_k$$

  - Where:
    - RMS: root mean square
    - $RMS[g]_k = \sqrt{E[g^2]_k + \varepsilon}$, $E[g^2]_k = \rho E[g^2]_{k-1} + (1-\rho) g_k^2$

*currently unpublished adaptive learning rate method. However, it is usually to cite slide 29 of Lecture 6 of Geoff Hinton's Coursera class.

Training

- AdaDelta: is an extension of Adagrad to handle the problem of continual decay of learning rates. Instead of accumulating all past squared gradients, it restricts the window of accumulated past gradients to some fixed size w

$$\theta^{k+1} = \theta^k + \Delta\theta^k,$$

$$\Delta\theta^k = -\frac{RMS[\Delta\theta]_{k-1}}{RMS[g]_k} g_k$$

  - Where:
    - RMS: root mean square
    - $RMS[\Delta\theta]_{k-1} = \sqrt{E[\Delta\theta^2]_{k-1} + \varepsilon}$, $E[\Delta\theta^2]_{k-1} = \rho E[\Delta\theta^2]_{k-2} + (1-\rho)\Delta\theta_{k-1}^2$
    - $RMS[g]_k = \sqrt{E[g^2]_k + \varepsilon}$, $E[g^2]_k = \rho E[g^2]_{k-1} + (1-\rho) g_k^2$

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Training

- Adaptive Moment Estimation (ADAM): is another method that computes adaptive learning rates for each parameter. It is similar to RMSProp with momentum. The simplified ADAM update looks as follows

$$m_k = \beta_1 m_{k-1} - (1 - \beta_1) g_k$$

$$v_k = \beta_2 v_{k-1} - (1 - \beta_2) g_k^2$$

$$\theta^{k+1} = \theta^k - \eta \frac{m_k}{\sqrt{v_k} + \varepsilon}$$

Diederik Kingma, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Training

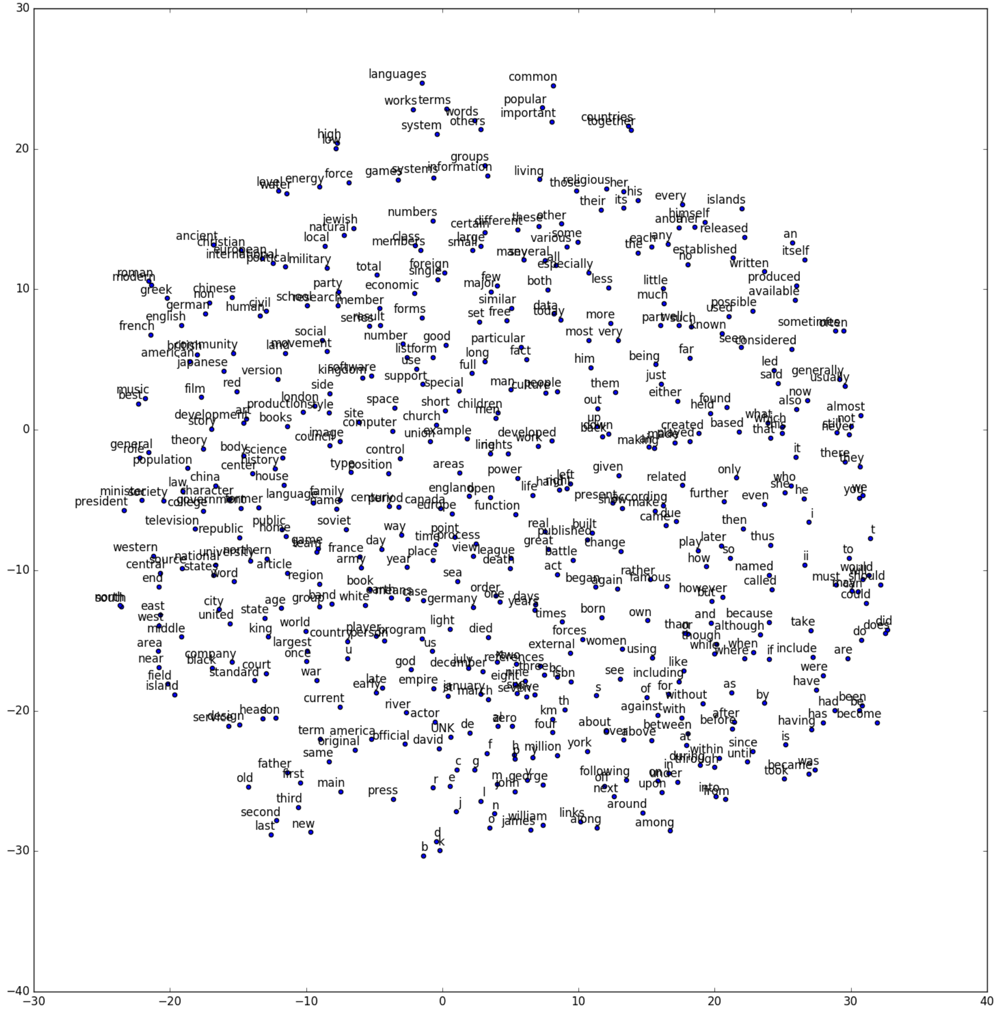
- Regularization
  - L2: $J(\theta) = \mathcal{L}(\theta) + \lambda||\theta||$
    - Where:
      - λ: decay rate
  - Dropout: $\vec{f}' = \vec{f} \circ r$
    - Where:
      - r: a masking vector of Bernoulli random variables with probability p of being 1
  - Batch normalization
  - Rescaling parameters θ when L2 exceeds a threshold
- Experimental tricks
  - OOV: randomly initialization
  - Fine-tune: slightly improve the performance

Outline

- Introduction
- Neural Network Background
- **Sentiment-oriented Word Embedding**
  - **Overview**
  - Traditional word embedding
  - Sentimental-oriented word embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Overview

- Traditional embedding is syntactically and semantically similar, but cannot distinguish sentimental differences.
- How to integrate sentiment information into word embedding
  - Use NN language model to learn syntactic and semantic information
  - Apply labeled data to augment sentiment orientation into word embedding



Outline

- Introduction
- Neural Network Background
- **Sentiment-oriented Word Embedding**
  - Overview
  - **Traditional word embedding**
  - Sentimental-oriented word embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

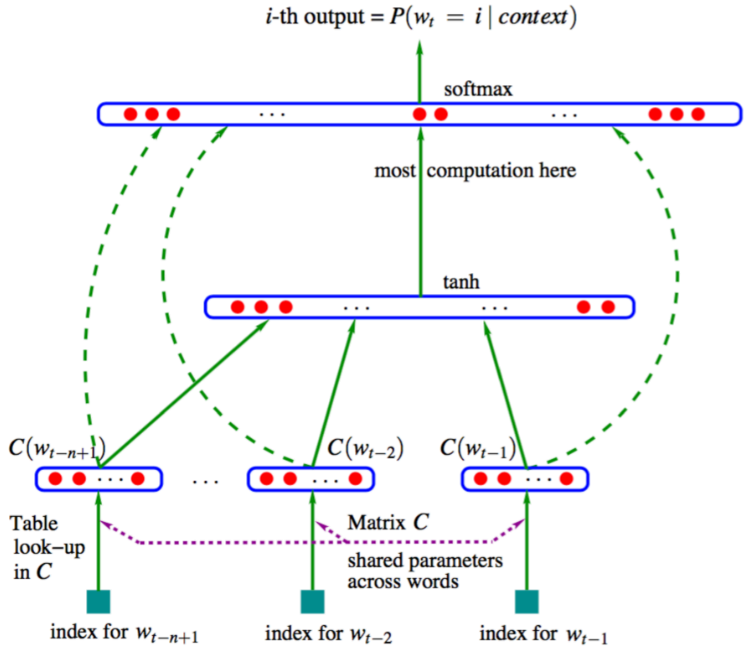
Traditional Word Embedding

- Unsupervised Learning
  - Basic neural network language models:
    - Input:
      - n-grams
    - Output:
      - probability score of the word given previous words
  - Objective function

$$P\left(w_t \middle| w_1^{t-1}\right) \approx P\left(w_t \middle| w_{t-n+1}^{t-1}\right)$$

➔ Problem: probability score

➔ High computation



Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3, 1137-1155.

Traditional Word Embedding

- Unsupervised Learning
  - Pairwise-ranking neural network language models
    - Input: a pair of
      - n-grams:

$$t = w_{i-k} w_{i-k+1} \dots \boldsymbol{w_i} \dots w_{i+k-1} w_{i+k}$$

      - corrupted n-grams:

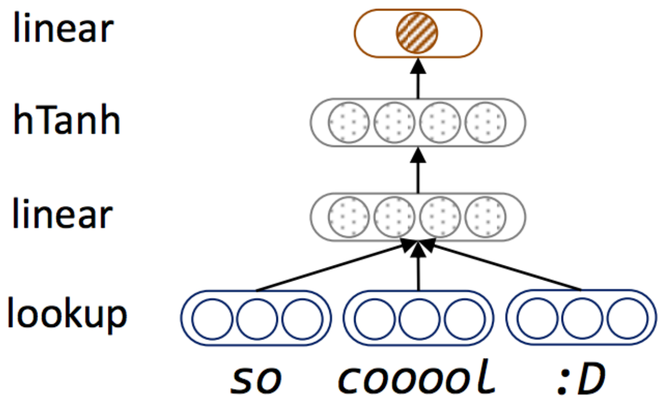
$$t^r = w_{i-k} w_{i-k+1} \dots \boldsymbol{w_i^r} \dots w_{i+k-1} w_{i+k}$$

    - Output:
      - margin scores $f(t), f(t^r)$
  - Objective function

$$\boldsymbol{loss_{cw}(t, t^r) = \max(0, 1 + f(t^r) - f(t))}$$

➔ Problem: Deep structure

➔ Still high computation



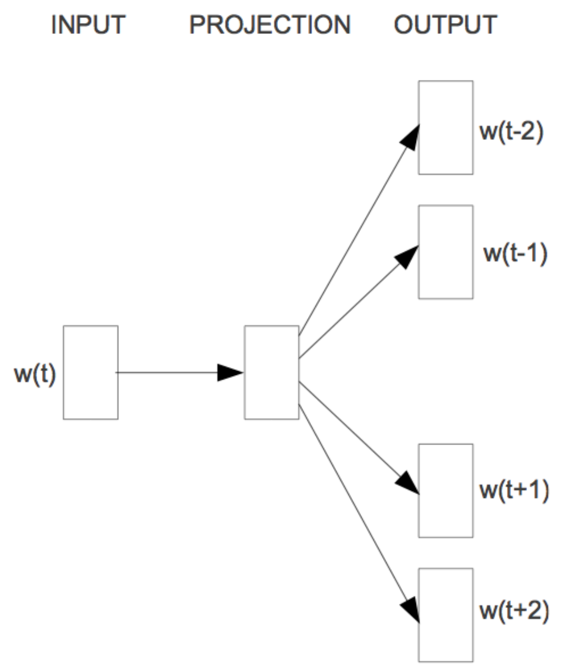
Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* 12, 2493-2537.

Traditional Word Embedding

- Unsupervised Learning
  - Simple neural network language models
    - Input:
      - n-grams
    - Output:
      - probability score of the context words given a word or vice versa
  - Objective function

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, c\neq 0} \log P\left(w_{t+j}\middle|w_t\right)$$

  - Optimization
    - Hierarchical softmax
    - Negative sampling



Skip-gram

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of *NIPS*, 3111-3119.

Outline

- Introduction
- Neural Network Background
- **Sentiment-oriented Word Embedding**
  - Overview
  - Traditional word embedding
  - **Sentimental-oriented word embedding**
- Sentence-level Models
- Document-level Models
- Fine-grained models
- Conclusion

Sentimental-Oriented word embedding

- Semi-Supervised Learning
  - Maas et al. (2011) combine an unsupervised probabilistic model and a supervised sentiment component to learn word embedding
  - Objective function

$$\boldsymbol{\upsilon}\|\boldsymbol{R}\|_F^2 + \sum_{\mathbf{k}=1}^{|\boldsymbol{D}|} \boldsymbol{\lambda}\|\widehat{\boldsymbol{\theta}}_{\boldsymbol{k}}\|_2^2 + \sum_{\mathbf{i}=1}^{\boldsymbol{N_k}} \boldsymbol{log}\, \mathbf{p}(\boldsymbol{w_i}|\widehat{\boldsymbol{\theta}}_{\boldsymbol{k}}; \mathbf{R}, \mathbf{b}) + \sum_{\mathbf{k}=1}^{|\boldsymbol{D}|} \frac{\mathbf{1}}{|\boldsymbol{S_k}|} \sum_{\mathbf{i}=1}^{\boldsymbol{N_k}} \boldsymbol{log}\, \mathbf{p}(\boldsymbol{s_k}|\boldsymbol{w_i}; \mathbf{R}, \boldsymbol{\psi}, \boldsymbol{b_c})$$

    - Where:
      - $p(w_i|\theta; R, b) = softmax(\theta^T \phi_{w_i} + b)$ ➔ maximum a posteriori (MAP)
      - $p(s = 1|w_i; R, \psi, \mathrm{b_c}) = \sigma(\psi^T \phi_{w_i} + b)$
      - $R \in \mathbb{R}^{\beta \times V}$: word embedding matrix with size of β
      - ϕ_{w_i} is embedding of w_i
      - θ, ψ, b, b_c: weight parameters and bias
      - υ, λ: hyper-parameters

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL:HLT*, 142-150.

Sentimental-Oriented word embedding

- Supervised Learning
  - Labutov and Lipson (2013) employ pre-trained embedding and labeled data to learn re-embedding words.
  - Objective function

$$\sum_{d_j \in D} \sum_{w_i \in d_j} \boldsymbol{log\, p(s_j | w_i; \Phi_T) - \lambda \|\Delta\Phi\|_F^2}$$

    - Where:
      - Φ_T, Φ_S: *embedding matrices of source and target words*
      - $p\left(s_j = 1 \middle| w_i; \Phi_T\right) = \sigma(\psi^T \phi_{w_i} + b)$
      - $\Delta\Phi = \Phi_T$-Φ_S
      - λ: hyper-parameter

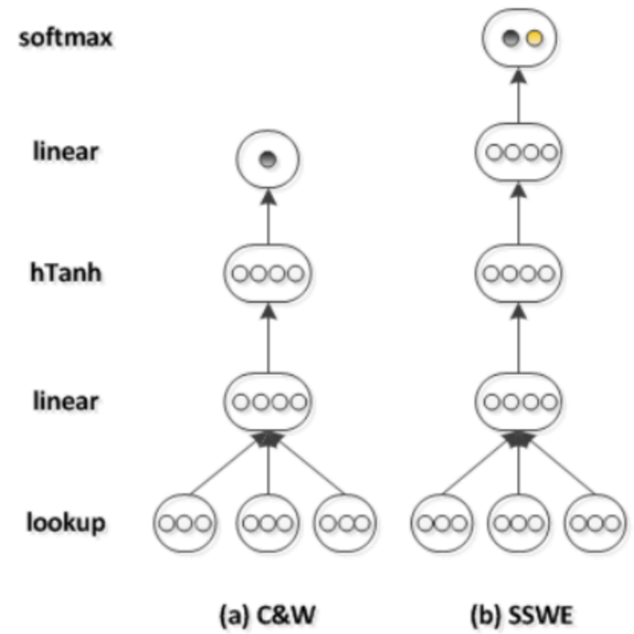
Igor Labutov, and Hod Lipson. 2013. Re-embedding words. In *Proceedings of ACL:HLT*, 489-493

Sentimental-Oriented word embeddings

- SSWE model (Tang et al.,2014)
  - Motivation: $x_{good} \approx x_{bad}$
  - Extend Collobert and Weston (2011) model
  - Adding sentimental information
  - Objective function

$$\boldsymbol{loss_{sswe}(t, t^r)} = \boldsymbol{\alpha \times loss_{cw}(t, t^r) + (1-\alpha) \times loss_s(t, t^r)}$$

    - Where
      - $loss_{cw}(t, t^r) = \max(0, 1 + f_0(t^r) - f_0(t))$
      - $loss_s(t, t^r) = \max(0, 1 + \delta_s(t) f_1(t^r) - \delta_s(t) f_1(t))$
      - $\delta_s(t) = \begin{cases} 1 & if \quad f^g(t) = [1,0] \\ -1 & if \quad f^g(t) = [0,1] \end{cases}$

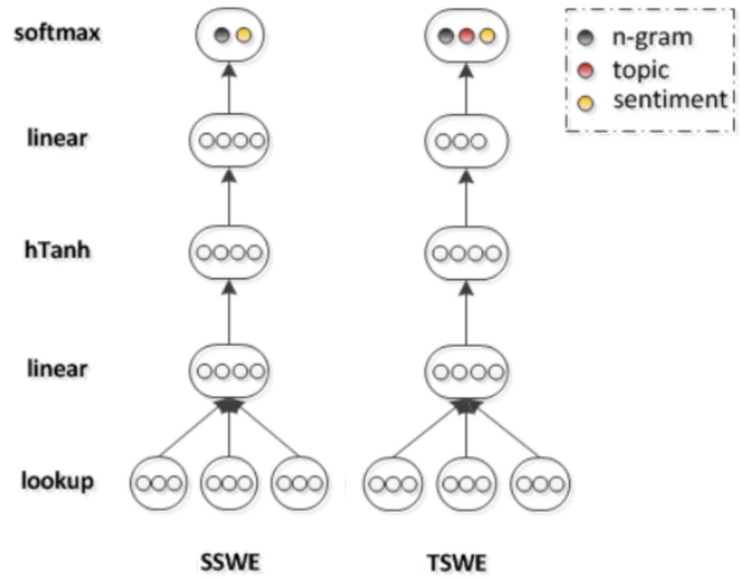


Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In Proceedings of *ACL*, 1555-1565.

Sentimental-Oriented word embeddings

- TSWE model (Ren et al., 2016)
  - Motivation
    - Different topics: offensive message vs offensive player
    - Multi-prototype embedding
  - An extension of Tang et al. (2014)
  - Augmenting topical information
  - Objective function

$$\begin{aligned} \boldsymbol{loss_{Tswe}(t, t^r)} &= \boldsymbol{\alpha \times loss_{cw}(t, t^r)} \\ &+ \boldsymbol{\beta \times loss_t(t, t^r)} \\ &+ \boldsymbol{(1-\alpha-\beta) \times loss_s(t, t^r)} \end{aligned}$$



- Where
  - $loss_{cw}(t, t^r) = \max(0, 1 + f_0(t^r) - f_0(t))$
  - $loss_t(t) = -\frac{f_t^g(t)\log(softmax(f_{1 \ldots N}(t)))}{N}$
  - $loss_s(t) = -\frac{f_s^g(t)\log(softmax(f_{N+1 \ldots (N+M)}(t)))}{M}$

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- **Sentence-level Models**
  - **Overview**
  - Bag-of-word methods
  - CNN
  - RecNN
  - RNN
- Document-level Models
- Fine-grained models
- Conclusion

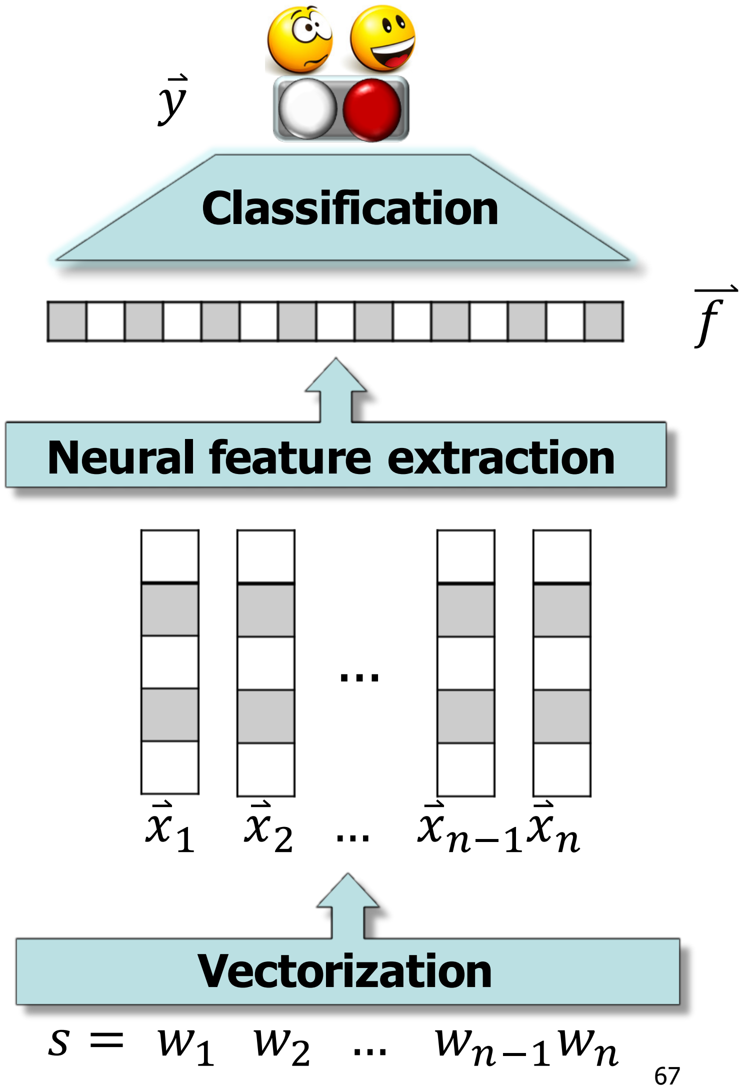
Overview

- Input: a sentence consists of n words
- Output: polarity or fine-grained sentiment

➔ Classification problem

- Classification layer

$$\vec{y} = softmax(\boldsymbol{W}_O\vec{f} + \vec{b}_o)$$

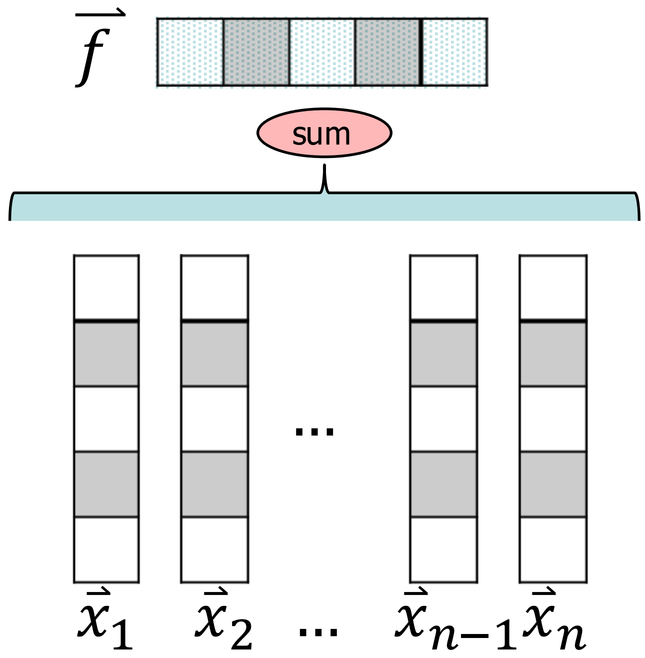


Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- **Sentence-level Models**
  - Overview
  - **Bag-of-words CNN**
  - RecNN
  - RNN
- Document-level Models
- Fine-grained models
- Conclusion

Bag-of-words

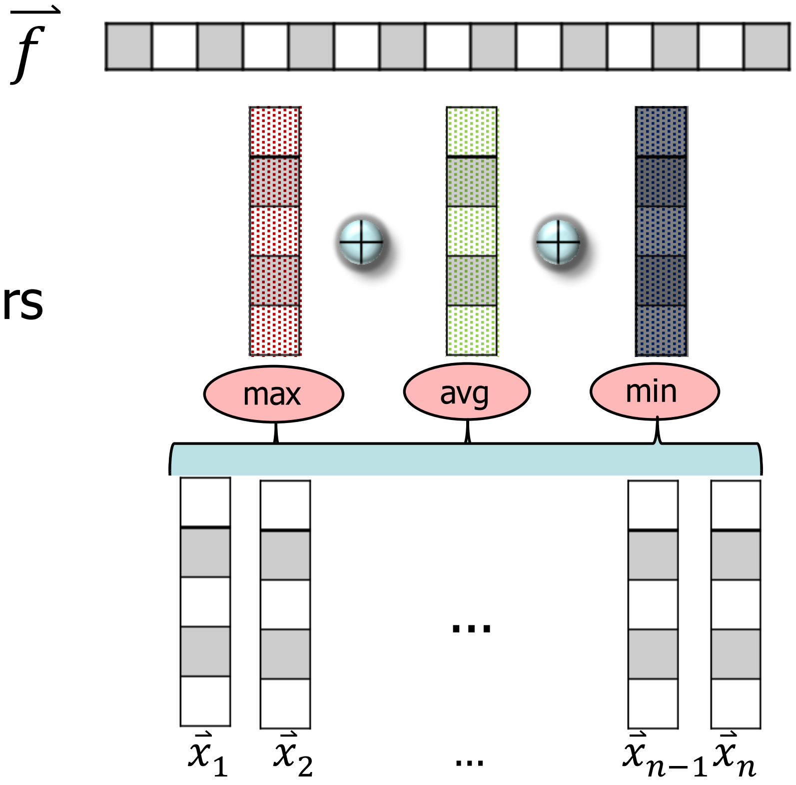
- Bag-of-words (Kalchbrenner et al., 2014)
  - Simply element-wise summing embedding
  - Learning embeddings by back-propagation



Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *In Proceedings of ACL, 655-665.*

Bag-of-words

- Pooling (Tang et. al.,2014; Vo and Zhang, 2015)
  - Make use of Pre-trained word embeddings
  - Extract salient features for traditional classifiers



Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In Proceedings of *ACL*, 1555-1565.

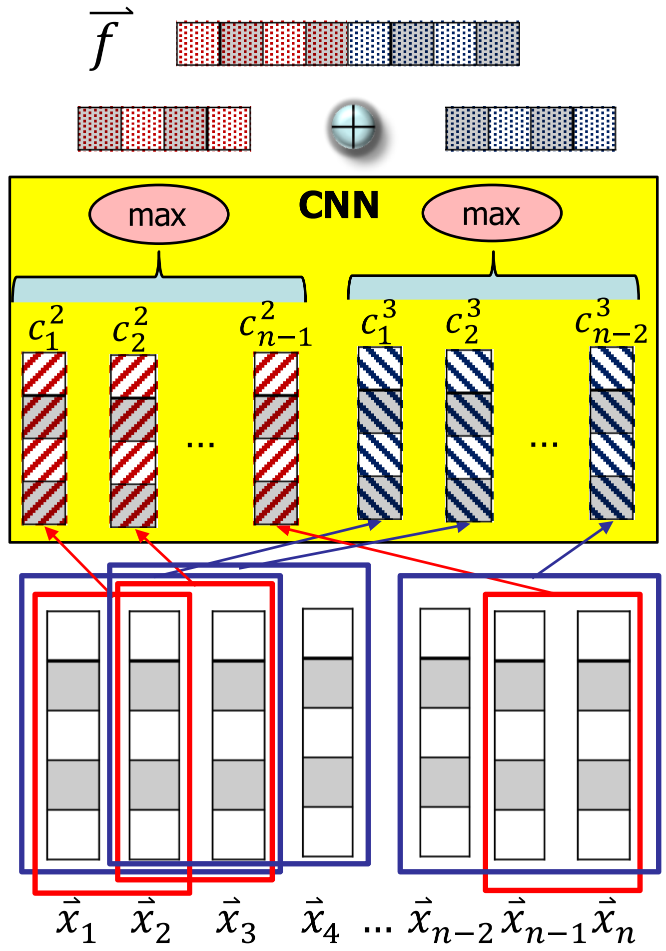
Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of* IJCAI, 1347-1353.

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- **Sentence-level Models**
  - Overview
  - Bag-of-word Methods
  - **CNN**
  - RecNN
  - RNN
- Document-level Models
- Fine-grained models
- Conclusion

Convolutional Neural Network

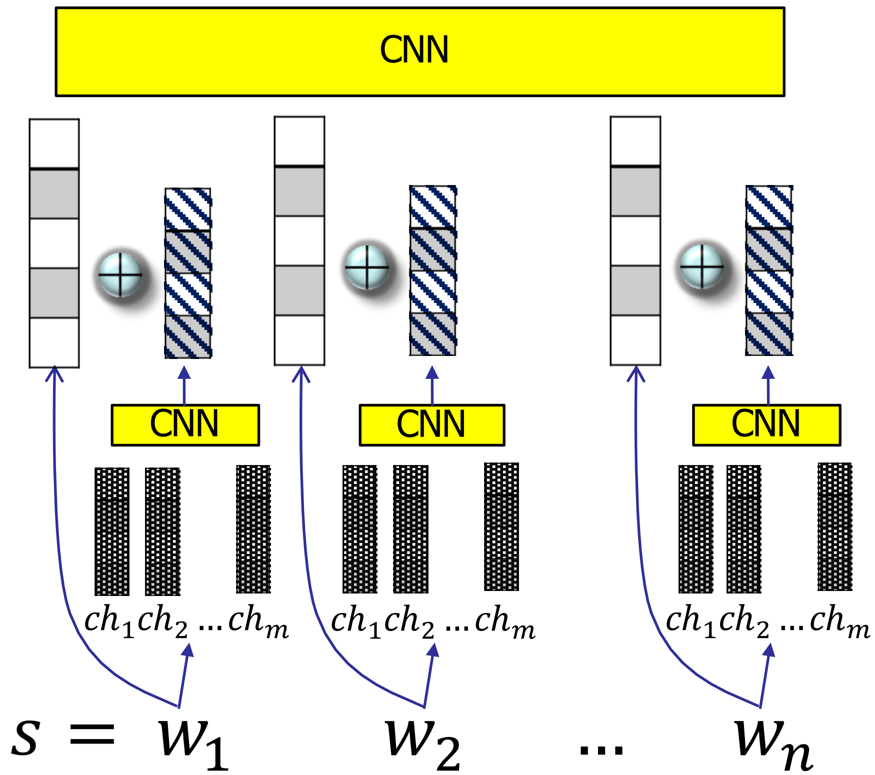
- CNN (Kim, 2014)
  - Feature combinations
  - Single CNN layer
  - Varied-window-size convolutional filters
  - Multichannel (1 static+ 1 nonstatic)



Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of EMNLP, 1746-1751.

Convolutional Neural Network

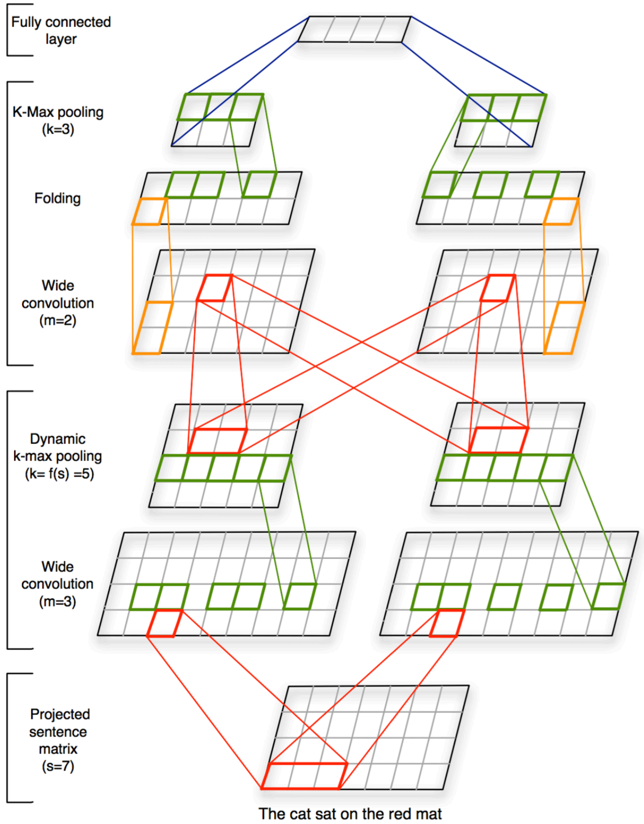
- Variations
  - dos Santos et al. (2014)
    - Add character information



Cícero Nogueira dos Santos, and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In Proceedings of *COLING*, 69-78.

Convolutional Neural Network

- Variations
  - Kalchbrenner et al. (2014)
    - Fixed-window-size convolutional filters
    - Multiple feature maps
    - K-max, with k dynamically decided
    - Stack multiple convolutional layers



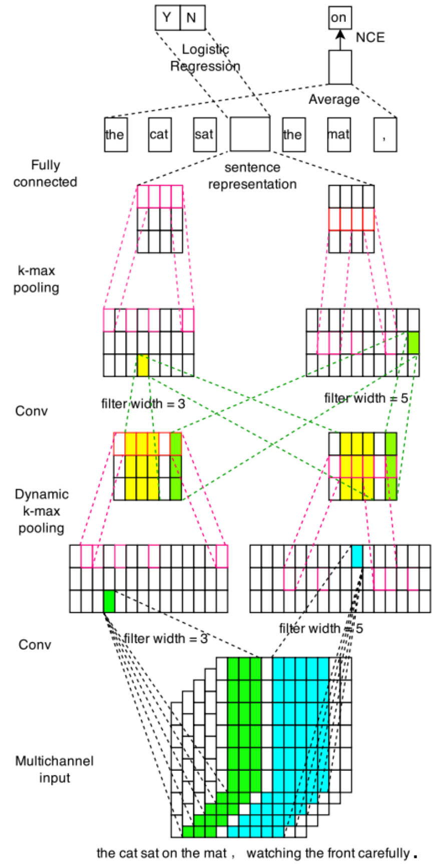
Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *In Proceedings of ACL, 655-665.*

Convolutional Neural Network

- Variations
  - Yin and Schütze (2015)
    - Inspired by CNN for RGB kernels in images
    - Employ different kinds of pre-trained embeddings as multichannel
    - Varied-window-size convolutional filters
    - K-max, with k dynamically decided
  - *Feature map* $F_{i,l}^{j}$:

$$F_{i,l}^{j} = \sum_{k=1}^{n} V_{i,l}^{j,k} * F_{i-1}^{k}$$

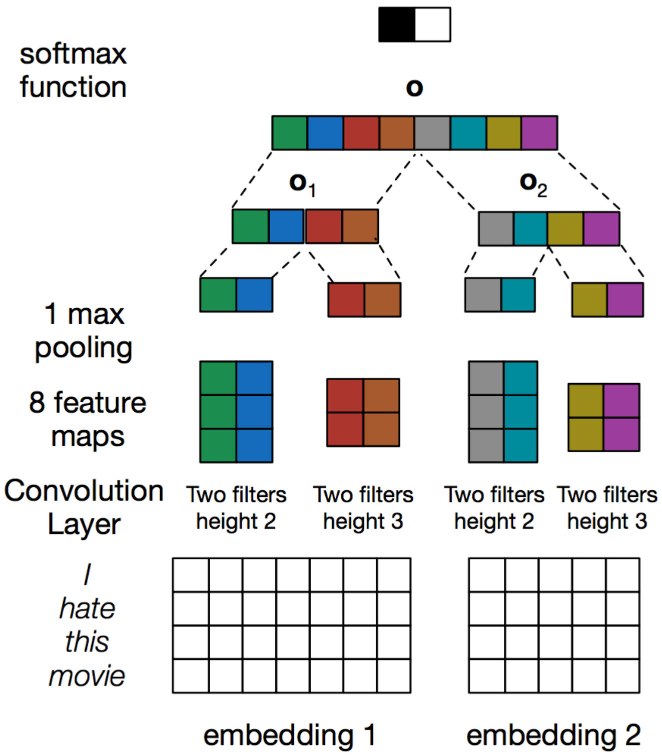
    - Where:
      - $*$: the convolution operation
      - j: the index of a feature map in layer i.
      - V: a rank 4 tensor weights



Convolutional Neural Network

- Variations
  - Zhang et al. (2016)
    - Make use of different sources of pre-trained embedding with different sizes
    - Employ different sets of convolutional filters

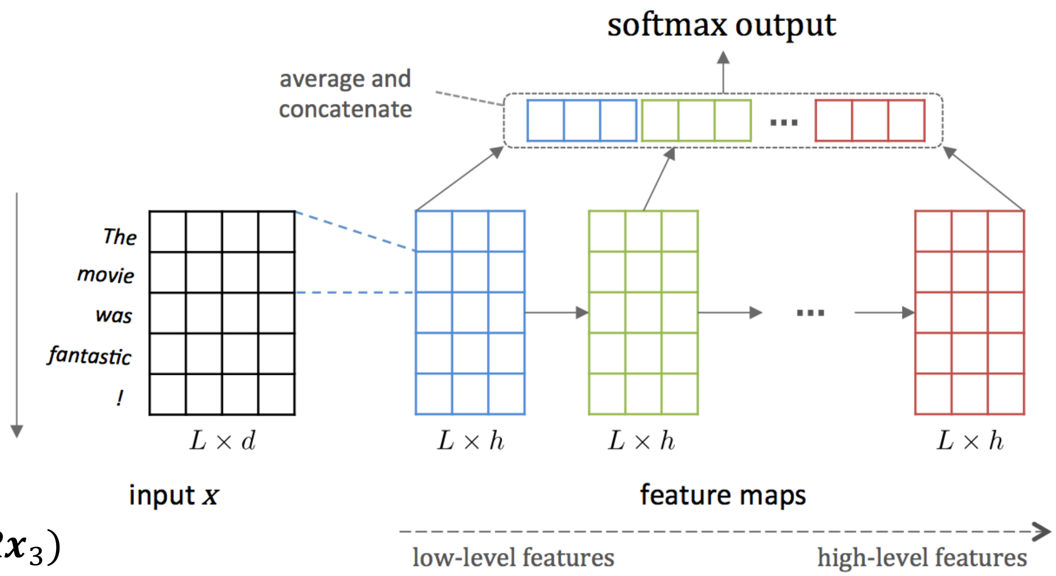
$$\vec{c}_i^{\,jk} = f(\boldsymbol{W}_c^j (\vec{x}_i^{\,j} \oplus \vec{x}_{i+1}^{\,j} \oplus \dots \oplus \vec{x}_{i+k}^{\,j}) + \vec{b}_c^{\,j})$$
$$\vec{o}_i^{\,j} = pool(\boldsymbol{C}_i^j)$$



Ye Zhang, Stephen Roller, and Byron Wallace. 2016. Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. In Proceedings of NAACL-HLT, 1522–1527 .

Convolutional Neural Network

- Variations
  - Lei et al. (2015)
    - N-gram tensor
    - Tensor-based feature mapping
    - Non-local
    - Non-linear



$$z = O^T(Px_1 \odot Qx_2 \odot Rx_3)$$
$$z[i,j,k] = O^T(Px_i \odot Qx_j \odot Rx_k)$$

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding CNNs for text: non-linear, non-consecutive convolutions. In Proceedings of EMNLP, 1565–1575 .

Outline

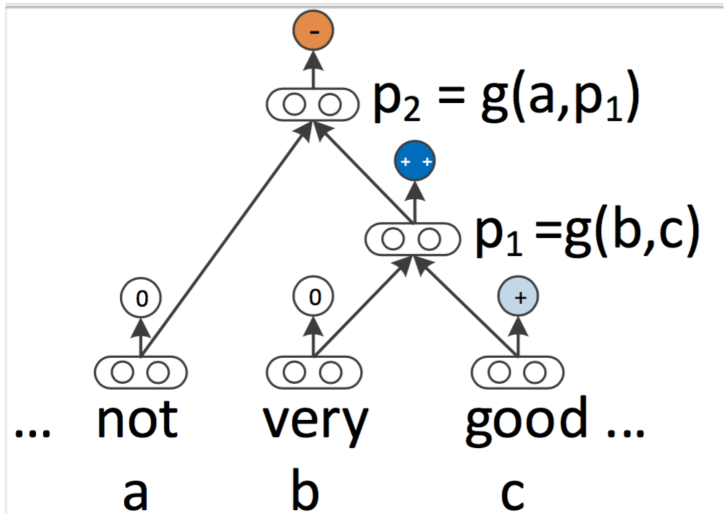
- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- **Sentence-level Models**
  - Overview
  - Bag-of-word methods
  - CNN
  - **RecNN**
  - RNN
- Document-level Models
- Fine-grained models
- Conclusion

Recursive Neural Network

- RecNN (Socher et al., 2013)

$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right)$$

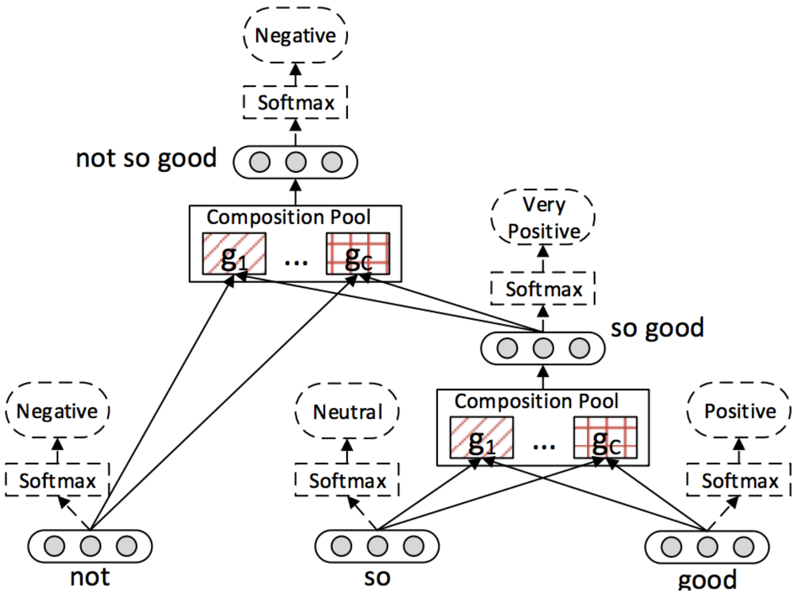
$$p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$



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 EMNLP*, 1642. 2013.

Recursive Neural Network

- Variations
  - Adaptive Multi-Compositionality RecNN (Dong et al., 2014)
    - Employ a set of composition functions



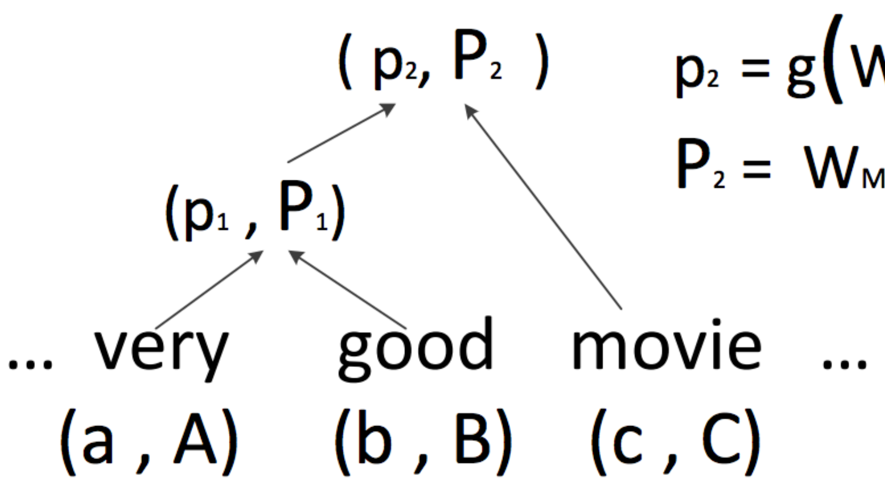
$$\boldsymbol{v}^i = f\left(\sum_{h=1}^{C} P(g_h | v_l^i, v_r^i) g_h(v_l^i, v_r^i)\right)$$

$$\begin{bmatrix} P(g_1 | v_l^i, v_r^i) \\ \ldots \\ P(g_C | v_l^i, v_r^i) \end{bmatrix} = \beta - softmax\left(\boldsymbol{S} \begin{bmatrix} \boldsymbol{v}_l^i \\ \boldsymbol{v}_r^i \end{bmatrix}\right)$$

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2014. Adaptive Multi-Compositionality for Recursive Neural Models with Applications to Sentiment Analysis. In Proceedings of *AAAI*, 1537-1543.

Recursive Neural Network

- Variations
  - Matrix-Vector RecNN (Socher et al., 2012)
    - Both matrix and vector
    - More composition interaction (Cross-way composition)
    - More features



$$p_2 = g\left(W\begin{bmatrix} Cp_1 \\ P_1c \end{bmatrix}\right)$$

$$P_2 = W_M \begin{bmatrix} P_1 \\ C \end{bmatrix}$$

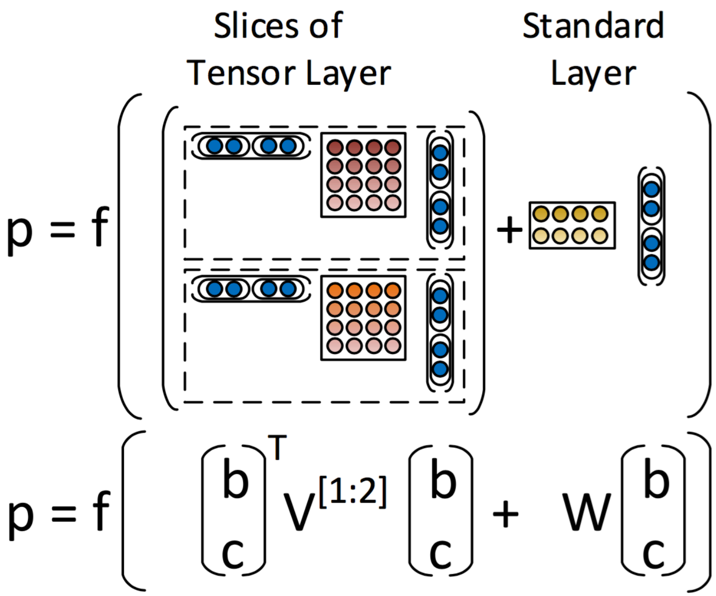
Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, 1201-1211.

Recursive Neural Network

- Variations
  - Recursive Neural Tensor Network (Socher et al.,2013)
    - Also more composition
    - Less parameters (embeddings)

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$



➔ **Problem:**

- Extracts non-local features
- Relies on external syntactic parsers for tree structure.

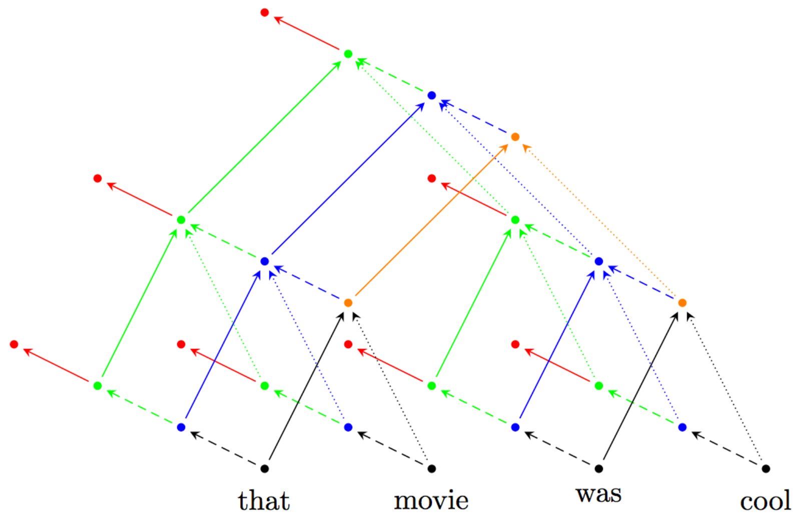
Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*,1642. 2013.

Recursive Neural Network

- Variations
  - Deep RecNN (Irsoy and Cardie 2014)
    - Stack multiple RecNN layers

$$h_{\eta}^{(i)} = f\left(\boldsymbol{W}_{L}^{(i)} h_{l(\eta)}^{(i)} + \boldsymbol{W}_{R}^{(i)} h_{r(\eta)}^{(i)} + \boldsymbol{V}^{(i)} h_{\eta}^{(i-1)} + b^{(i)}\right)$$

    - Where:
      - i: stacked layer index
      - $W_L^{(i)}, W_R^{(i)}, V^{(i)}, b^{(i)}$: weight and bias parameters
      - $l(\eta), \mathrm{r}(\eta)$: left and right children of η



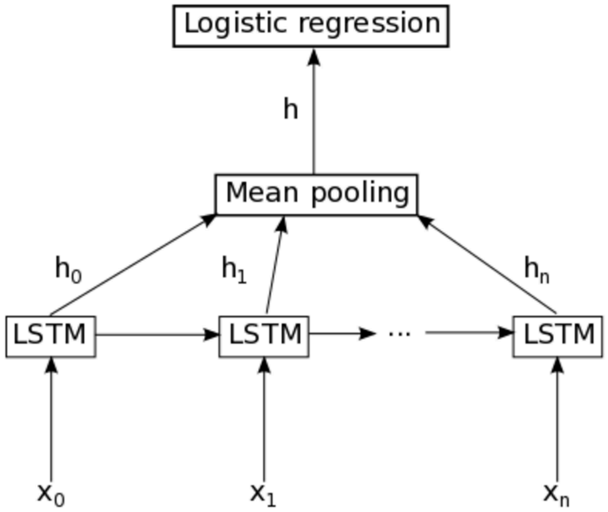
Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- **Sentence-level Models**
  - Overview
  - Bag-of-word methods
  - CNN
  - RecNN
  - **RNN**
- Document-level Models
- Fine-grained models
- Conclusion

Recurrent Neural Network

- LSTM (Wang et al., 2015)
  - Use a standard LSTM
  - Fine-tune word embeddings

$$\vec{f}_{t} = \sigma(\boldsymbol{W}_f \vec{x}_t + \boldsymbol{U}_f \vec{h}_{t-1} + \vec{b}_f)$$
$$\vec{i}_{t} = \sigma(\boldsymbol{W}_i \vec{x}_t + \boldsymbol{U}_i \vec{h}_{t-1} + \vec{b}_i)$$
$$\vec{u}_{t} = tanh(\boldsymbol{W}_u \vec{x}_t + \boldsymbol{U}_u \vec{h}_{t-1} + \vec{b}_u)$$
$$\vec{c}_{t} = \vec{i}_{t} \odot \vec{u}_{t} + \vec{f}_{t} \odot \vec{c}_{t-1}$$
$$\vec{o}_{t} = \sigma(\boldsymbol{W}_o \vec{x}_t + \boldsymbol{U}_o \vec{h}_{t-1} + \vec{b}_o)$$
$$\vec{h}_{t} = \vec{o}_{t} \tanh \odot (\vec{c}_{t})$$



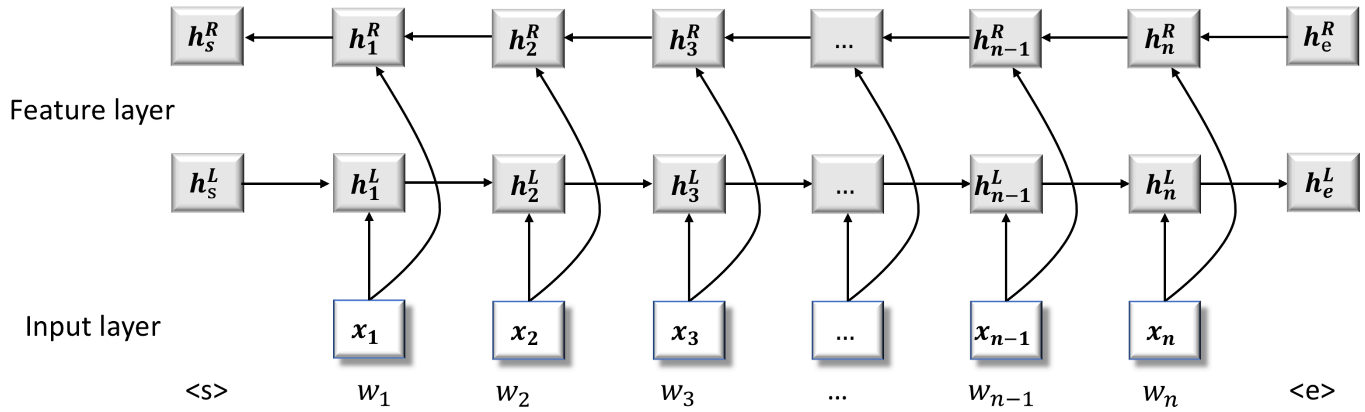
Source: http://deeplearning.net/tutorial/lstm.html

Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. "Predicting polarities of tweets by composing word embeddings with long short-term memory. 2015. In *Proceedings of ACL*, 1343-1353.

Recurrent Neural Network

- Variations
  - Bi-directional LSTM: Tai et al. (2015), Li et al. (2015), Teng et al. (2016)

$$f = h_s^R \oplus h_e^L$$



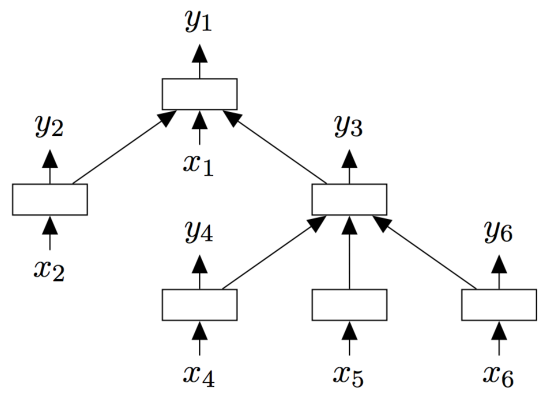
Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In Proceedings of ACL, 1556–1566.

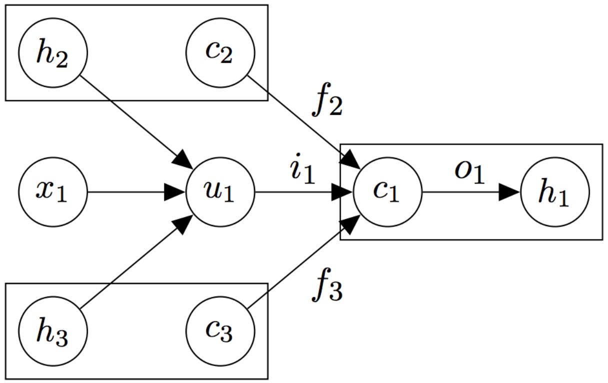
Jiwei Li, Minh-Thang Luong, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations?. In Proceedings of EMNLP, 2304–2314.

Zhiyang Teng, Duy Tin Vo and Yue Zhang.Context-Sensitive Lexicon Features for Neural Sentiment Analysis. In Proceeddings of EMNLP 2016. Austin, Texas, USA, November.

Recurrent Neural Network

- Variations
  - Tree Structured LSTM: Tai et al. (2015); Li et al. (2015); Zhu et al. (2015)
    - Child-sum tree ➔ Dependency tree
    - N-ary tree ➔ Constituency tree





Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In Proceedings of ACL, 1556–1566 .

Jiwei Li, Minh-Thang Luong, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations?. In Proceedings of EMNLP, 2304–2314.

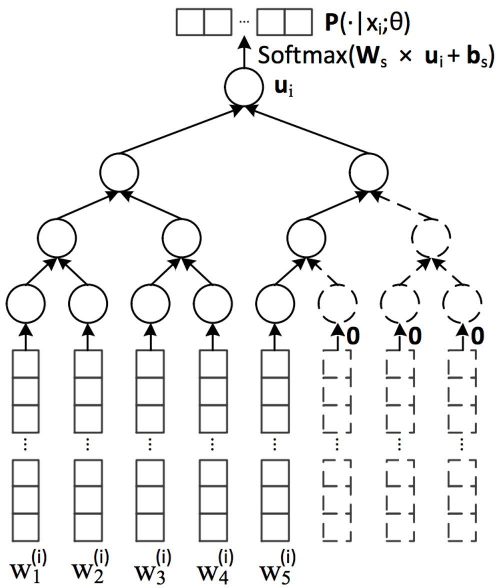
Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proceedings of ICML*,1604-1612.

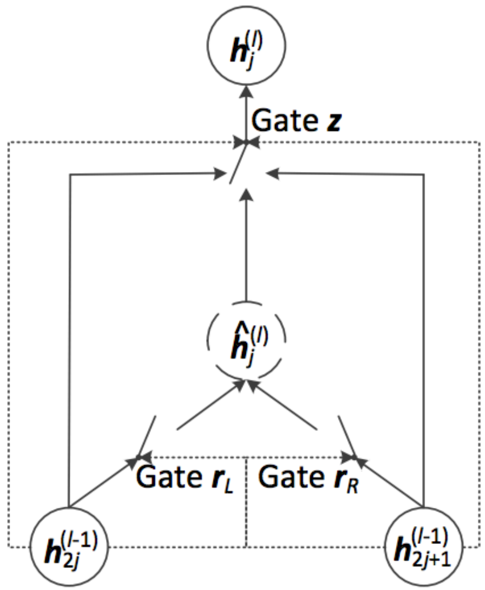
Recurrent Neural Network

❖ Variations

– Gated RecNN (Chen et al., 2015)

- Build a gated structure on the full binary tree





Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Shiyu Wu, and Xuanjing Huang. 2015. Sentence modeling with gated recursive neural network." In *Proceedings of EMNLP*, 793-798.

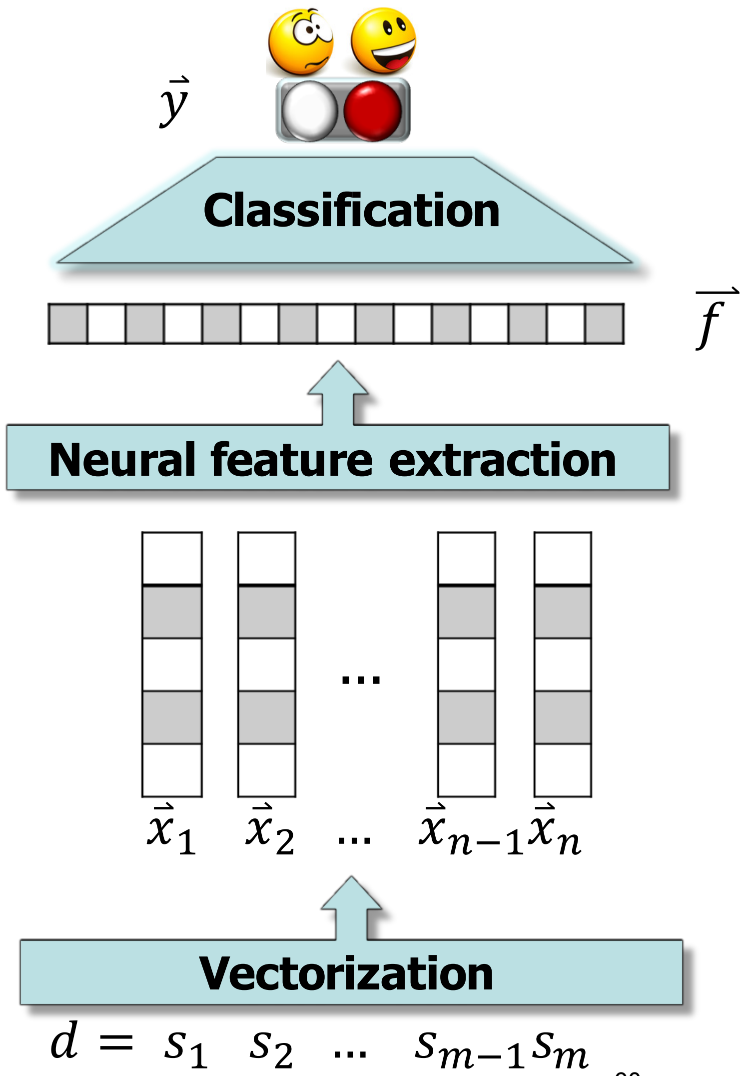
Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- **Document-level Models**
  - **Overview**
  - Document Embedding
  - Flat Models
  - Hierarchical Learning
- Fine-grained models

Overview

- Input: a document consists of m sentences
- Output: polarity or fine-grained sentiment
- ➔ Classification problem
- Classification layer

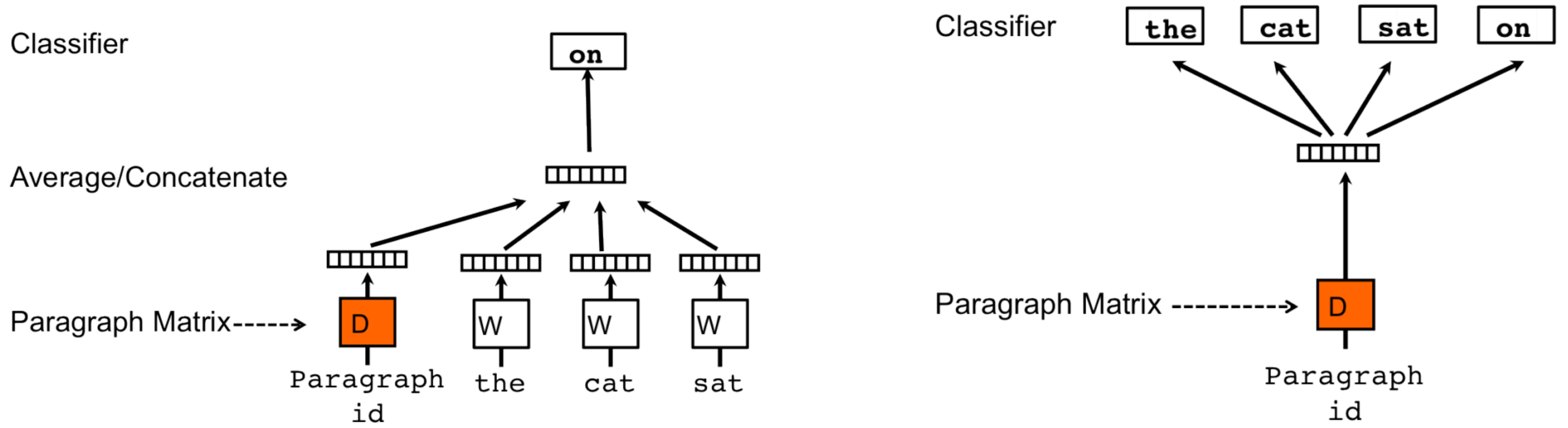
$$\vec{y} = softmax(\boldsymbol{W}_O \vec{f} + \vec{b}_o)$$



Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- **Document-level Models**
  - Overview
  - **Document Embedding**
  - Flat Models
  - Hierarchical Learning
- Fine-grained models

Document Embedding



- Extend Word2vec models (Mikolov et al., 2013) to learn document representations
- Utilize document representation as features for MLP classification

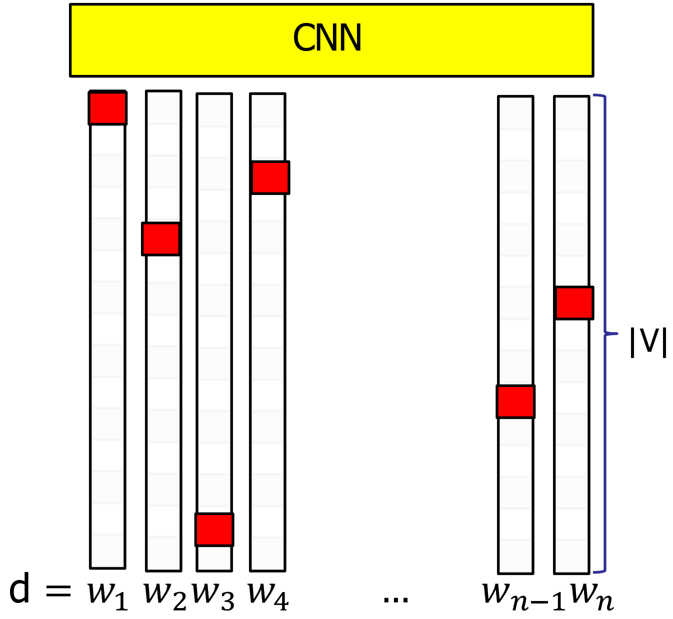
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of *NIPS*, 3111-3119.
Quoc V. Le, and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In Proceedings of *ICML*, 1188-1196.

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- **Document-level Models**
  - Overview
  - Document Embedding
  - **Flat Models**
  - Hierarchical Learning
- Fine-grained models

Flat Models

- **Sentence-level-based models**
- CNN Variations
  - Jonhson and Zhang (2015a)
    - seq-CNN: use one-hot inputs for a word
    - bow-CNN: use one-hot inputs for n-grams
  - Jonhson and Zhang (2015b)
    - Augment inputs by CNN-based region embeddings
- LSTM Variations
  - Jonhson and Zhang (2016):
    - Extend Jonhson and Zhang (2015b) model by applying LSTM



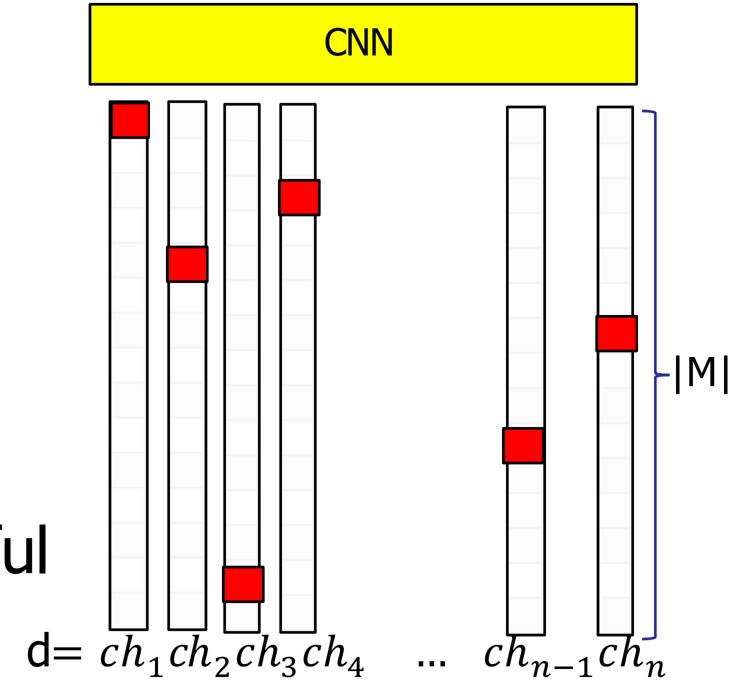
➔ One-hot encoding is efficient to represent variable-sized document

Rie Johnson and Tong Zhang. 2015a. Effective use of word order for text categorization with convolutional neural networks. In Proceedings of NAACL:HLT, 103-112.
Rie Johnson, and Tong Zhang. 2015b. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Proceedings of NIPS,* 919-927.
Rie Johnson, and Tong Zhang. 2016. Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings. In *Proceedings of ICML,* 526-534.

Flat Models

- Deep CNN Variations
  - Zhang et al. (2015)
    - Use one-hot character-level inputs
    - Stack 6 convolutional layers
  - Conneau et al. (2016)
    - Employ character embeddings
    - Build up to 49 CNN layers

➔ Character-level representation is also helpful



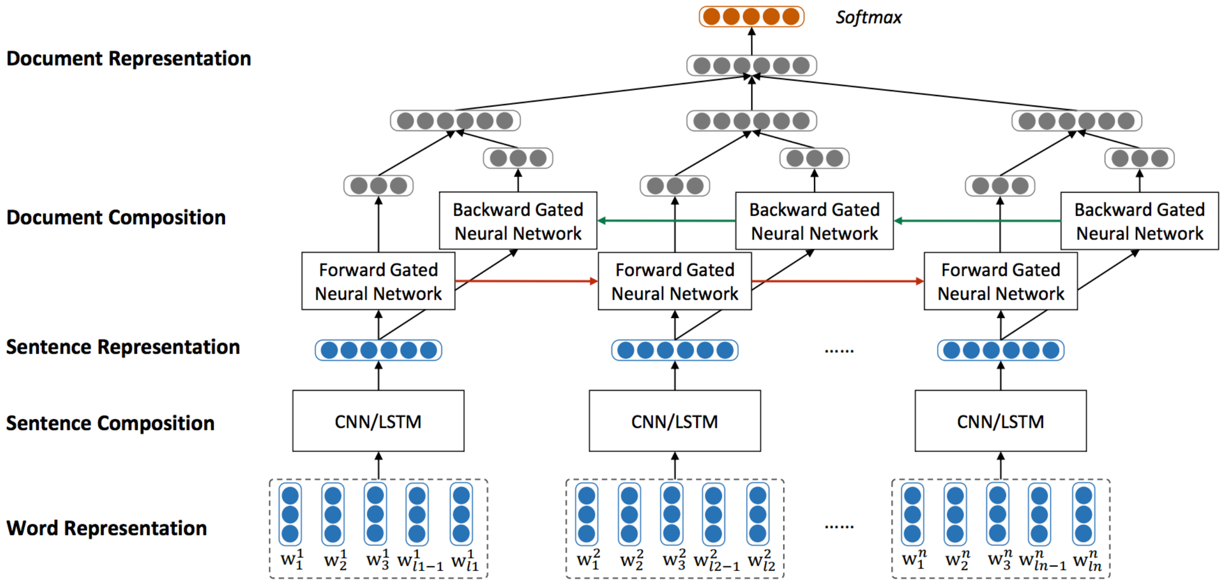
Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of NIPS,* 649-657.
Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very Deep Convolutional Networks for Natural Language Processing. *arXiv preprint arXiv:1606.01781*.

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- **Document-level Models**
  - Overview
  - Document Embedding
  - Flat Models
  - **Hierarchical Learning**
- Fine-grained models

Hierarchical Learning

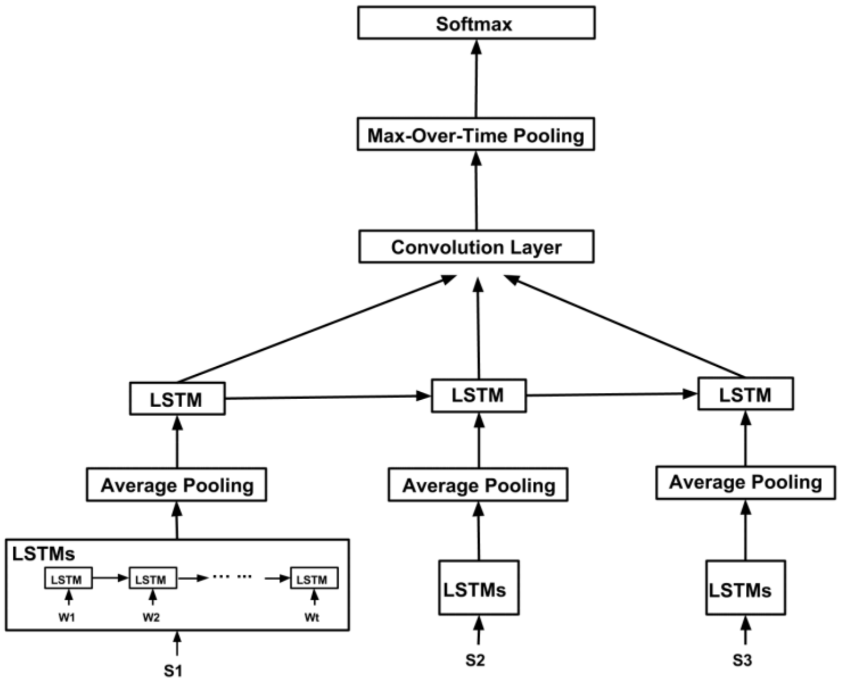
- Pooling (Tang et al., 2015a)
  - Average pooling sentence representations as document representation
- LSTM/CNN-GRU (Tang et al., 2015b)



Duyu Tang, Bing Qin, and Ting Liu. 2015a. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of ACL*.
Duyu Tang, Bing Qin, and Ting Liu. 2015b. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of EMNLP*, 1422-1432.

Hierarchical Learning

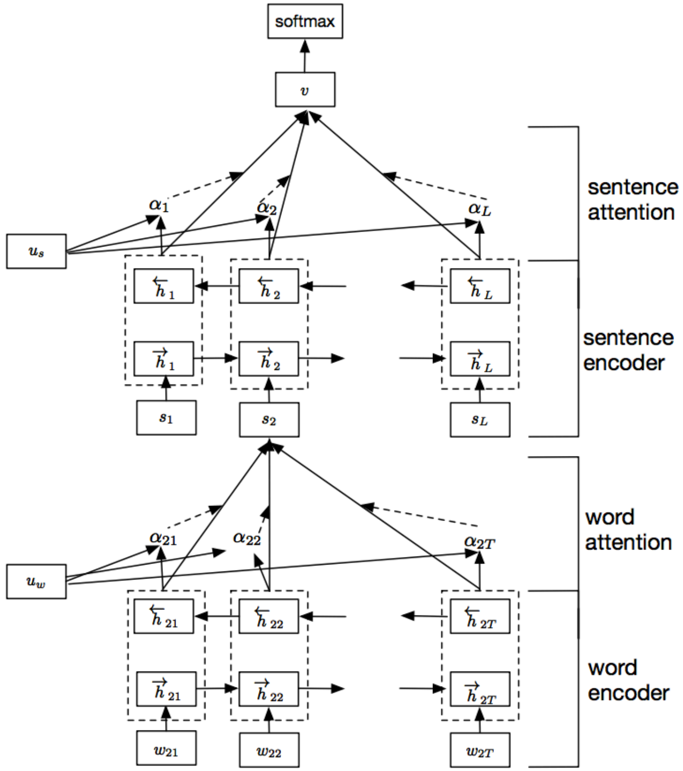
- Variations
  - LSTM-CNN (Zhang et al., 2016)



Rui Zhang, Honglak Lee, and Dragomir Radev. 2016. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In *Proceedings of NAACL:HLT*, 1512-1521.

Hierarchical Learning

- Variations
  - GRU-GRU Attention networks (Yang et al., 2016)



Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL:HLT.*

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- **Fine-grained models**
  - **Overview**
  - Targeted Sentiment
  - Open-domain Targeted Sentiment
  - Opinion Expression Detection
- Conclusion

Overview

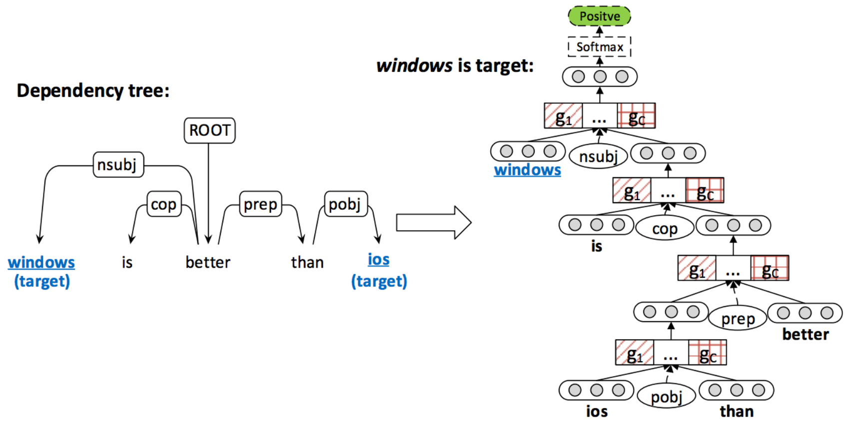
- Inputs:
  - A sentence consists of n words.
    - With a given target → **Classification problem**
    - Without a given target → **Sequence labeler**
- Output:
  - **[Who]** holds **[which opinions]** towards **[whom]**

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- **Fine-grained models**
  - Overview
  - **Targeted Sentiment**
  - Open-domain Targeted Sentiment
  - Opinion Expression Detection
- Conclusion

Targeted Sentiment

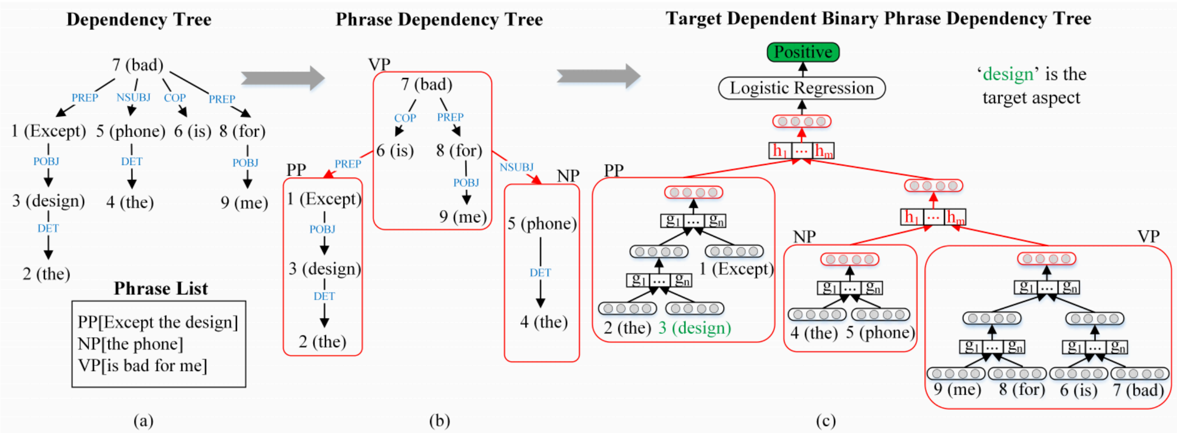
- Tree-structure-based
  - Dong et al. (2014)
    - Variant RecNN
    - Dependency tree



Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In Proceedings of *ACL*, 49-54.

Targeted Sentiment

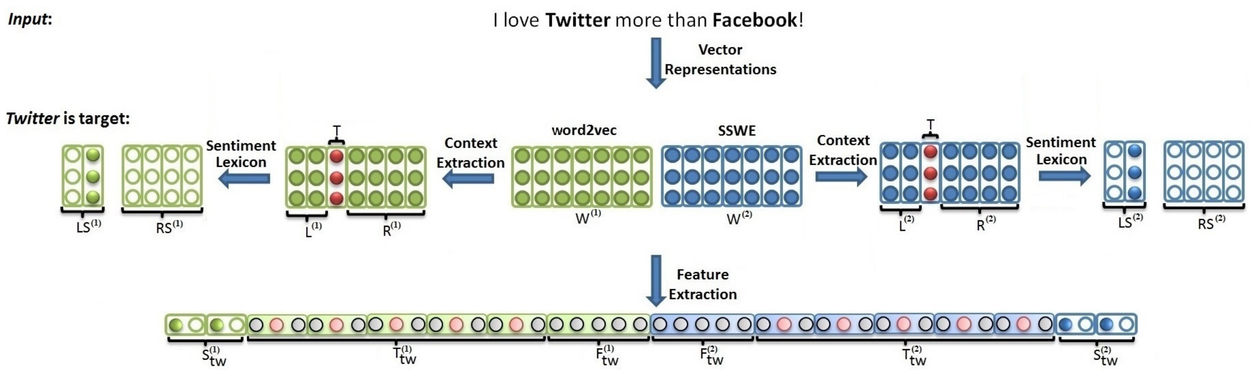
- Tree-structure-based
  - Nguyen and Shirai (2015)
    - Variant RecNN
    - Dependency+Constituent trees



Thien Hai Nguyen, and Kiyoaki Shirai. 2015. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In *Proceedings of EMNLP*, 2509-2514.

Targeted Sentiment

- Pattern-based
  - Vo and Zhang (2015)
    - Pooling mechanisms



$$P_{tw} = [F_{tw}^{(1)}, T_{tw}^{(1)}, S_{tw}^{(1)}, F_{tw}^{(2)}, T_{tw}^{(2)}, S_{tw}^{(2)}]$$

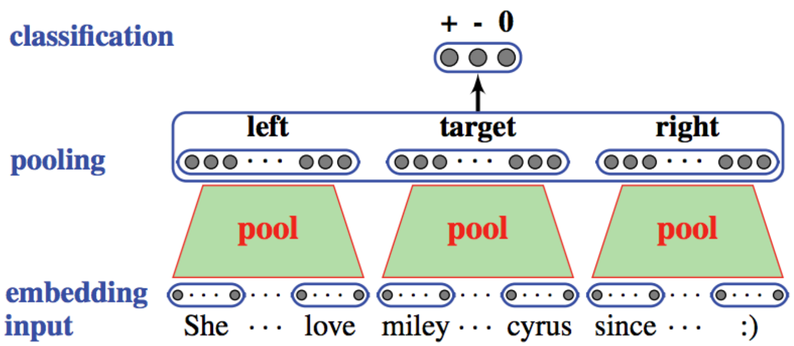
Where:

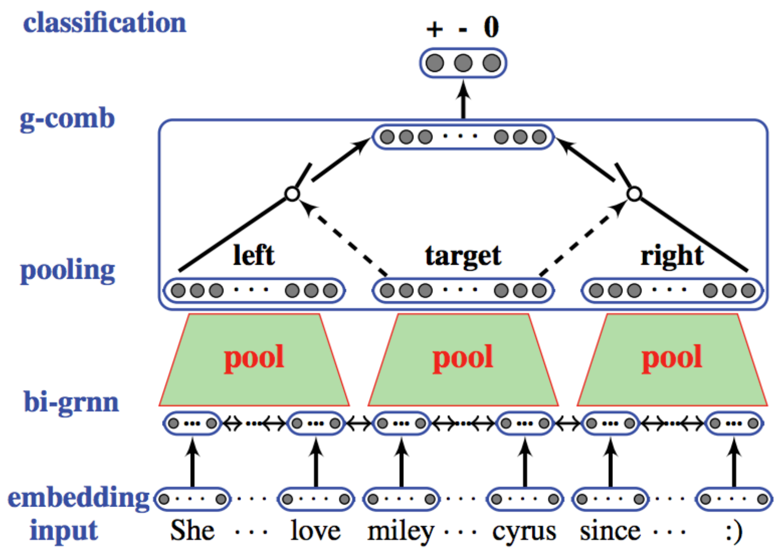
- $F_{tw}^{(i)} = P(W^{(i)})$
- $T_{tw}^{(i)} = [P(L^{(i)}), P(T^{(i)}), P(R^{(i)})]$
- $S_{tw}^{(i)} = [P(LS^{(i)}), P(RS^{(i)})]$
- $P(X) = [f_1(X), \dots, f_k(X)]$
- f_k: *pooling functions*

Duy-Tin Vo, and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of IJCAI*.

Targeted Sentiment

- Pattern-based
  - Zhang et al. (2016)
    - Gated mechanisms





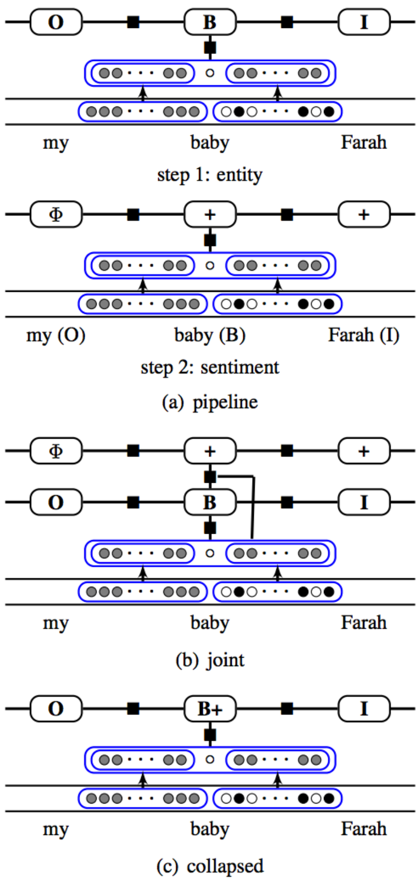
Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated Neural Networks for Targeted Sentiment Analysis. In *Proceedings of AAAI.*

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- **Fine-grained models**
  - Overview
  - Targeted Sentiment
  - **Open-domain Targeted Sentiment**
  - Opinion Expression Detection
- Conclusion

Open-domain Targeted Sentiment

- Open domain (detect target and its sentiment)
  - Zhang et. al.(2015)
    - Neural CRF
    - Discrete features



(a) pipeline

(b) joint

(c) collapsed

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of EMNLP*, 612-621.

Outline

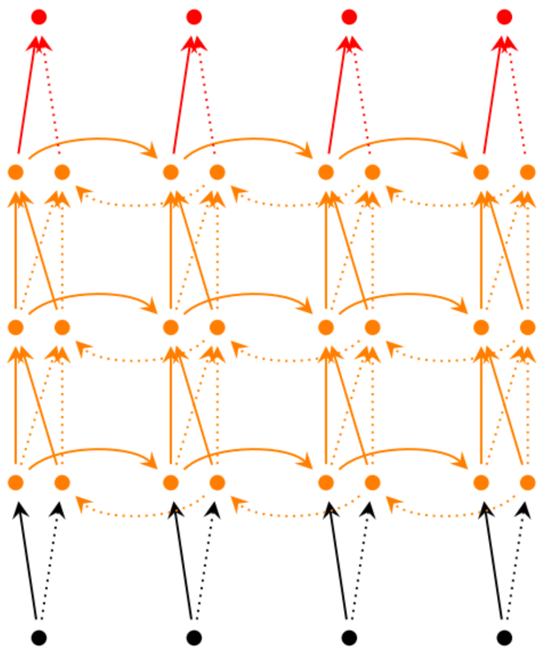
- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- **Fine-grained models**
  - Overview
  - Targeted Sentiment
  - Open-domain Targeted Sentiment
  - **Opinion Expression Detection**
- Conclusion

Opinion Expression Detection

- Detect opinion expression
  - Irsoy and Cardie (2014)
    - Deep biRNN

The	committee	,	as	usual	,	has
O	O	O	B_ESE	I_ESE	O	B_DSE

refused	to	make	any	statements	.
I_DSE	I_DSE	I_DSE	I_DSE	I_DSE	O

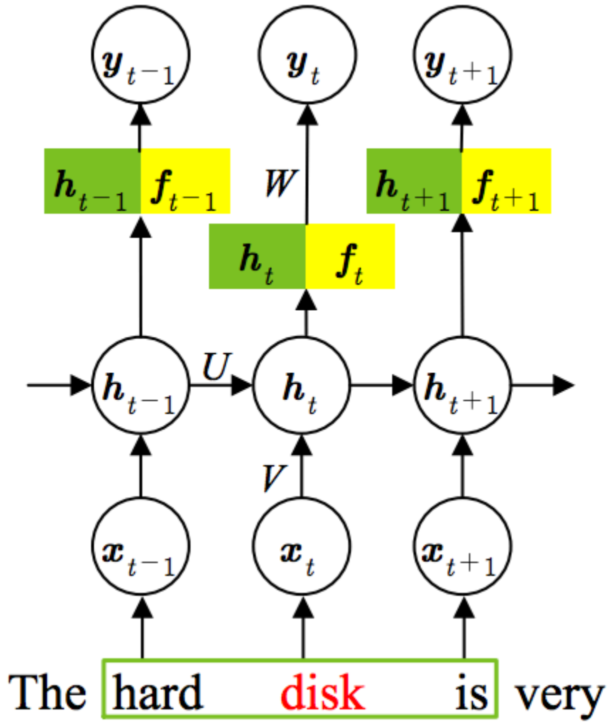


Ozan Irsoy, and Claire Cardie. 2014. Opinion Mining with Deep Recurrent Neural Networks. In Proceedings of *EMNLP*, 720-728.

Opinion Expression Detection

- Opinion expression and detect target
  - Liu et al. (2015)
    - LSTM
    - Discrete features

The	**hard**	**disk**	is	*very*	*noisy*
O	B-TARG	I-TARG	O	O	O
O	O	O	O	B-EXPR	I-EXPR



Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of EMNLP*.

Outline

- Introduction
- Neural Network Background
- Sentiment-oriented Word Embedding
- Sentence-level Models
- Document-level Models
- Fine-grained models
- **Conclusion**

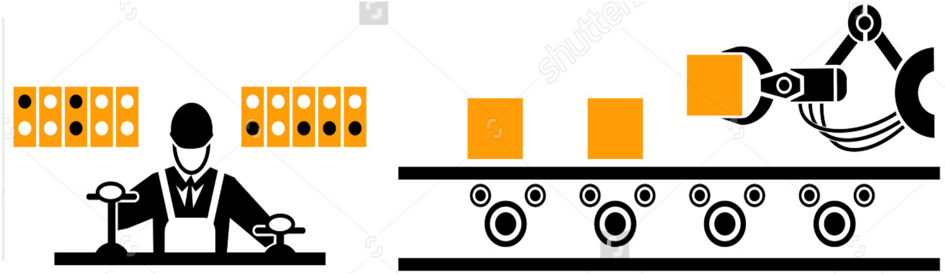
Conclusion

Raw data

Feature Engineering models

Conclusion

Raw data



Neural network models

Thank you!!!