Sentiment Analysis with Neural Networks

Duyu Tang
Associate Researcher
Natural Language Computing Group
Microsoft Research

Meishan Zhang
Associate Professor
School of Computer Science and Technology
Heilongjiang University

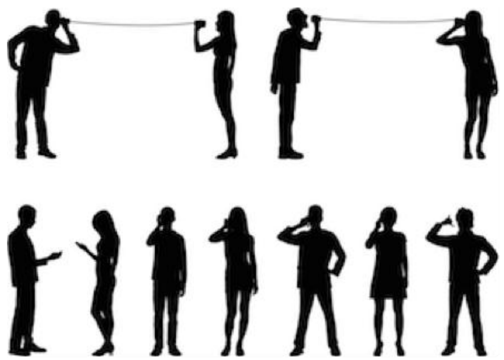
前沿技术讲习班
Advanced Technology Tutorial

Outline

- Definition of sentiment analysis
- Sentiment-specific word embedding (Duyu)
- Sentence composition (Meishan)
- Document composition (Duyu)
- Fine-grained sentiment classification/extraction (Meishan)

Sentiment/Opinion

- Why are sentiments/opinions so important?
  - Sentiments are key influencers of our behaviors.
  - Our beliefs and perceptions of reality are conditioned on how others see the world.
  - Whenever we need to make a decision we often seek out others' opinions.
    - True for both individuals and organizations



- It is simply the "human nature"
  - We want to express our opinions
  - We also want to hear others' opinions

Sentiment Analysis

- Computational study of opinions, sentiments, appraisal, and emotions expressed in text.
  - Reviews of movies, hotels, restaurants, etc.
  - Reviews of products
  - Comments for news
  - Tweets

  - Yelp/Dianping/TripAdvisor/RT/IMDB, etc.
  - Amazon/Taobao
  - Twitter/FB/Weibo

Bing Liu. 2012. **Sentiment analysis and opinion mining**. In *Synthesis lectures on human language technologies*, 1-167.

hp printer

HP Officejet 6500A Plus e-All-in-One E710n Color Ink-jet - Fax / copier / printer / scanner - English, French, Spanish / Canada, United States

$152 online

★★★★☆ 1,251 reviews

September 2010 - Printer - Copier - Fax - Scanner - HP - Inkjet - Office - Wireless - Color - Duplex - 250 sheet - 64 MB memory

Want an all-in-one that's fast, economical, and eco-conscious? The Officejet 6500 All-in-One will save your color page as well as let you use less energy. You'll reduce intervention with the 35-page automatic document feeder. And you can network this versatile machine for group use via built-in Ethernet networking.

Summary - Based on 1,251 reviews

1 | 2 | 3 | 4 stars | 5 stars

What people are saying

ease of use	"Easy to set up, makes great copies, works very well"
setup	"Setup, including wireless was quick."
value	"Great value and super easy set up."
design/style	"Nice sleek look, nice features."
customer service	"customer service was unable to help resolve the issue."
picture/video	"Good quality prints and photos."
size	"Pretty Paper weight."

Great Printer!

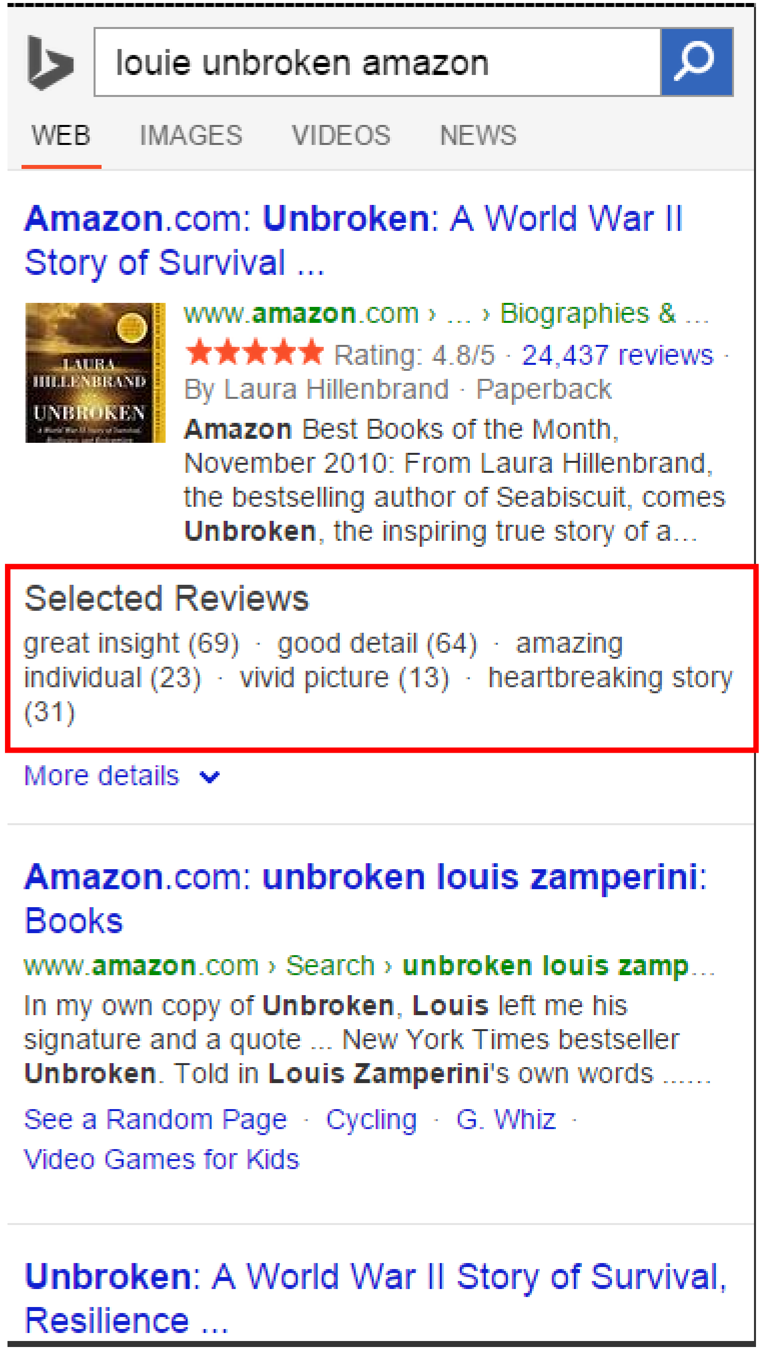
★★★★★ By ttmae12 - Jun 7, 2011 - Staples

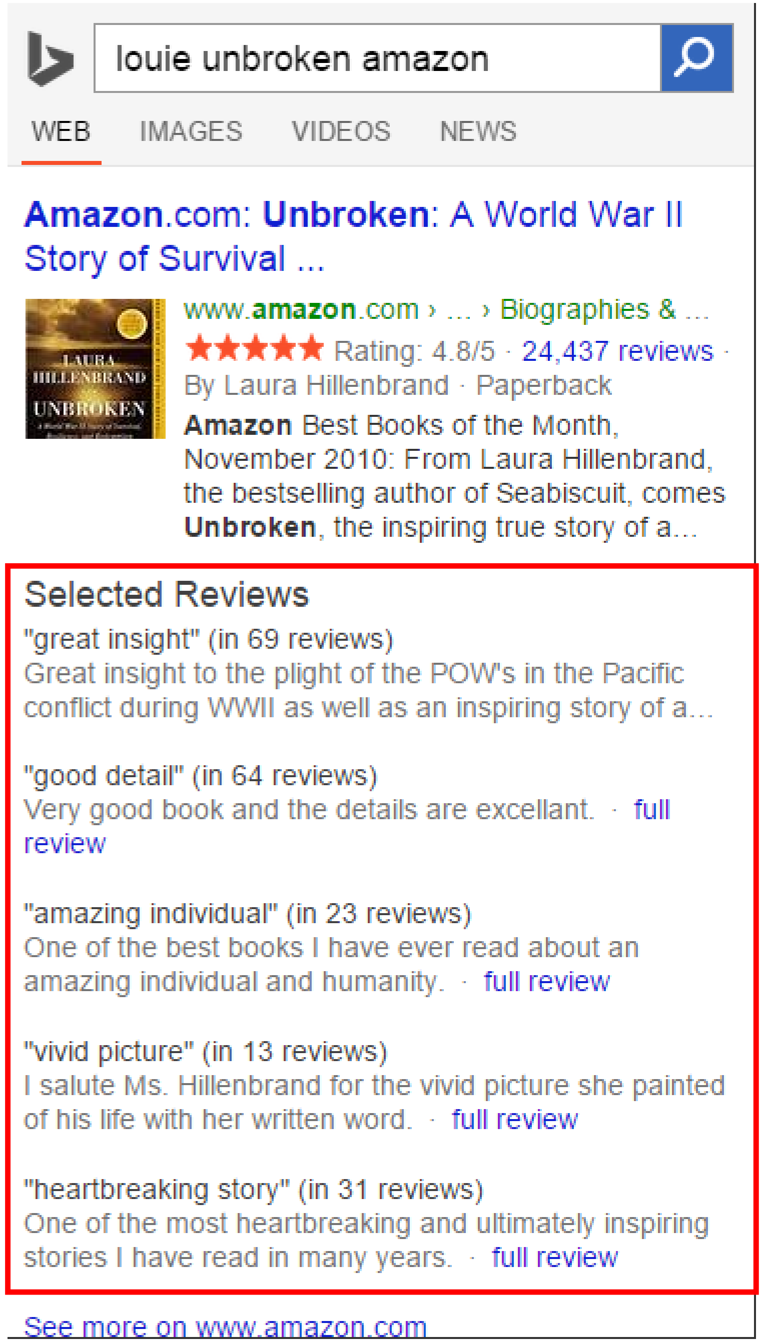
Pros: Great Print Quality; Fast Operation; Easy To Use; Quiet; Easy To Set Up; Compact Design; Reliable

This is an excellent printer for the price. I am amazed at how nice it prints. Machine was easy to setup. Printer is wired to my desktop and wireless to my iPad. Prints beautifully!
The only problem I encountered was printing photos. Once the correct paper quality is selected, there were no issues. I have always had good luck with HP products and the company has good customer support as well. In today's economy customer service and a great product means a lot.
Thanks

Full review provided by: STAPLES

Google Product Search (10/01/2012)


louie unbroken amazon
WEB
IMAGES
VIDEOS
NEWS
Amazon.com: Unbroken: A World War II Story of Survival ...
LAURA HILLENBRAND
UNBROKEN
www.amazon.com › ... › Biographies & ...
Rating: 4.8/5 · 24,437 reviews ·
By Laura Hillenbrand · Paperback
Amazon Best Books of the Month, November 2010: From Laura Hillenbrand, the bestselling author of Seabiscuit, comes Unbroken, the inspiring true story of a...
Selected Reviews
great insight (69) · good detail (64) · amazing individual (23) · vivid picture (13) · heartbreaking story (31)
More details
Amazon.com: unbroken louis zamperini: Books
www.amazon.com › Search › unbroken louis zamp...
In my own copy of Unbroken, Louis left me his signature and a quote ... New York Times bestseller Unbroken. Told in Louis Zamperini's own words ......
See a Random Page · Cycling · G. Whiz ·
Video Games for Kids
Unbroken: A World War II Story of Survival, Resilience ...


louie unbroken amazon
WEB
IMAGES
VIDEOS
NEWS
Amazon.com: Unbroken: A World War II Story of Survival ...
LAURA HILLENBRAND
UNBROKEN
www.amazon.com › ... › Biographies & ...
Rating: 4.8/5 · 24,437 reviews ·
By Laura Hillenbrand · Paperback
Amazon Best Books of the Month, November 2010: From Laura Hillenbrand, the bestselling author of Seabiscuit, comes Unbroken, the inspiring true story of a...
Selected Reviews
"great insight" (in 69 reviews)
Great insight to the plight of the POW's in the Pacific conflict during WWII as well as an inspiring story of a...
"good detail" (in 64 reviews)
Very good book and the details are excellant. · full review
"amazing individual" (in 23 reviews)
One of the best books I have ever read about an amazing individual and humanity. · full review
"vivid picture" (in 13 reviews)
I salute Ms. Hillenbrand for the vivid picture she painted of his life with her written word. · full review
"heartbreaking story" (in 31 reviews)
One of the most heartbreaking and ultimately inspiring stories I have read in many years. · full review
See more on www.amazon.com

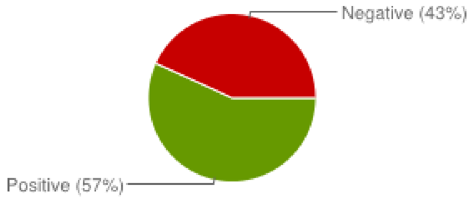
Sentiment140

Tweet 353 | Like 140 | +1 74

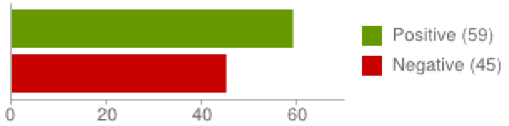
microsoft | English | Search | Save this search

Sentiment analysis for microsoft

Sentiment by Percent



Sentiment by Count



Tweets about: microsoft

Isaydumb: @Youporn, in my humble opinion you have nothing to do on the @Xbox Live. What the fuck is **@Microsoft** doing?!
Posted 46 seconds ago

Megan_Maracle: I hate this class. **#Microsoft** #die
Posted 2 minutes ago

dilwortha: @carasmith10 oh okay, you'll have to explain when i see you as i dont understand this disk haha. is it for **microsoft** project do you

Posted 5 minutes ago

jlebrech: @rsslldnphy it happens to be **microsoft** this time, but a superset is the next best thing from a compiled bytecode, as valid JS is also

http://www.sentiment140.com/

The results for this query are: Accurate

Sentiment Analysis

- Definition: A sentiment is a quadruple
  - Opinion targets: entities/aspects to be evaluated
  - Sentiments: positive and negative
  - Opinion holders: persons who hold opinions
  - Time: when opinions are given

  - Id: Alice on 1-May-2014 “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. However, the price is a little high“

Target	Sentiment	Holder	Time
iPhone	positive	Alice	1-May-2014
touch screen	positive	Alice	1-May-2014
price	negative	Alice	1-May-2014

Bing Liu. 2012. **Sentiment analysis and opinion mining**. In *Synthesis lectures on human language technologies*, 1-167.

Sentiment Analysis Tasks

- Objective: Given an opinion document
  - Discover all/parts of sentiment quadruples (t, s, h, time)
  - Unstructured text → Structured data

- Tasks
  - Word level sentiment analysis
  - Sentiment/Document level sentiment classification
  - Target/Aspect level sentiment classification
  - Aspect extraction

Sentiment Classification

- Input
  - Text (sentences, reviews, tweets, etc.)
  - Target/Aspect
- Output
  - Label: positive, negative or neutral
  - Score

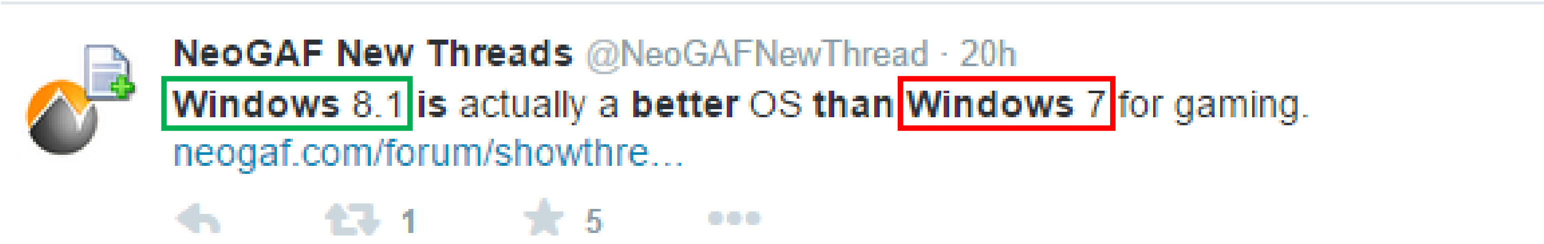
Sentiment Classification

Positive

Sentiment/Document level

The price is great and the service even better

Positive



Positive

Negative

Target/Aspect level

Representation Learning is Important for Sentiment Analysis

- Inferring the sentiment of text requires us to deeply understand the semantic meanings of text.

- Domin machir on the ... end

- It is de ... age the kn engine

Feature group	Examples
word ngrams	*grrreat, show, grrreat_show, miss_NEG, miss_NEG_the*
character ngrams	*grr, grrr, grrre, rrr, rrre, rrrea*
all-caps	all-caps:1
POS	POS_N:1 (nouns), POS_V:2 (verbs), POS_E:1 (emoticons), POS_,:1 (punctuation)
automatic lexicon features	HS_unigrams_positive_count:4, HS_unigrams_negative_total_score:1.51, HS_unigrams_POS_N_combined_total_score:0.19, HS_bigrams_positive_total_score:3.55, HS_bigrams_negative_max_score:1.98
manual lexicon features	MPQA_positive_affirmative_score:2, MPQA_negative_negated_score:1, BINGLIU_POS_V_negative_negated_score:1
punctuation	punctuation_!:1
emoticons	emoticon_positive:1, *emoticon_positive_last*
elongated words	elongation:1

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. **Representation learning: A review and new perspectives**. *IEEE Trans. Pattern Analysis and Machine Intelligence*

tweets. In *SemEval 2013.*

Outline

- Definition of sentiment analysis
- Sentiment-specific word embedding (Duyu)
- Sentence composition (Meishan)
- Document composition (Duyu)
- Fine-grained sentiment classification/extraction (Meishan)

Word Embedding

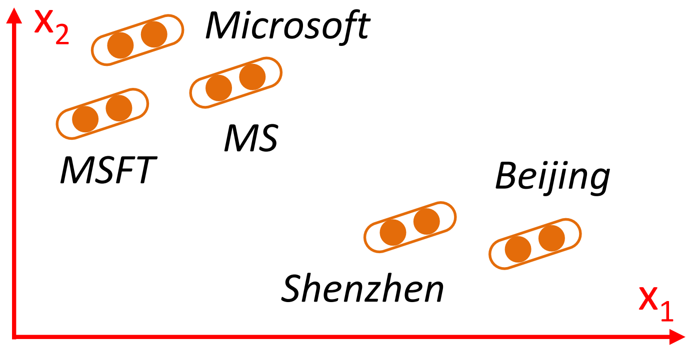
- Traditional: one-hot representation
  - Words are treated atomic, one-hot representation

Microsoft = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0]
MSFT = [0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0]

→ ***Microsoft* & *MSFT* = 0**

- Embedding
  - Continuous representation of meaning

$$\textit{Microsoft} = \begin{pmatrix} 1.045 \\ 0.912 \\ -0.894 \\ -1.053 \\ 0.459 \end{pmatrix}$$



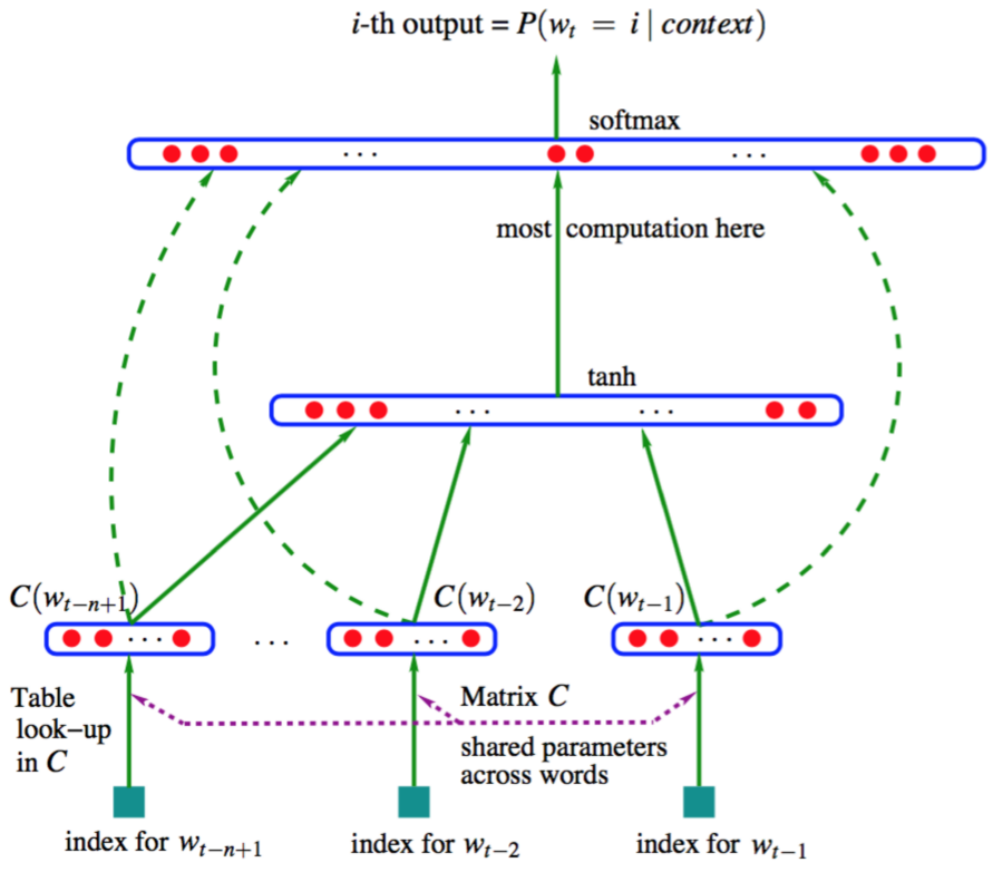
Context-based Models

- Neural language model
  - Predict based approach

- Objective function

Loss(target word | context words; Vectors)

$$L = \frac{1}{T}\sum_t \log \hat{P}(w_t | w_1^{t-1})$$



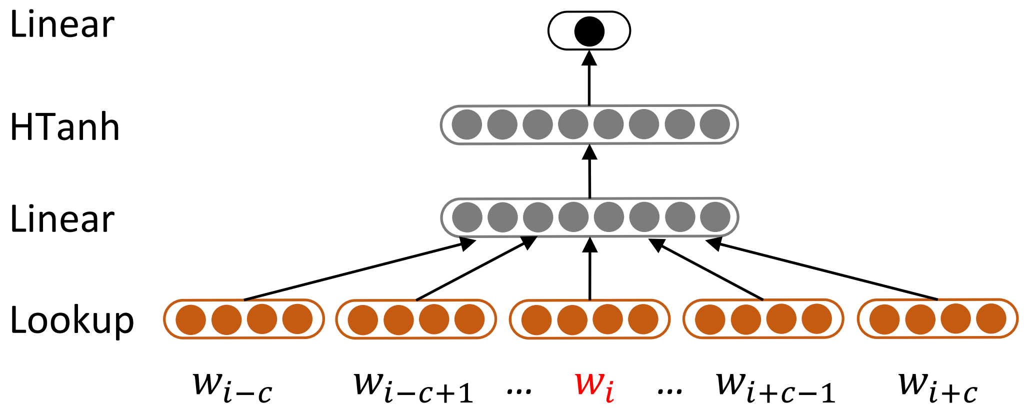
Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. **A neural probabilistic language model**. *J. Mach. Learn. Res.* 3, 1137-1155.

Context-based Models

- Ranking based approach
  - Distinguish between real and corrupted word sequence

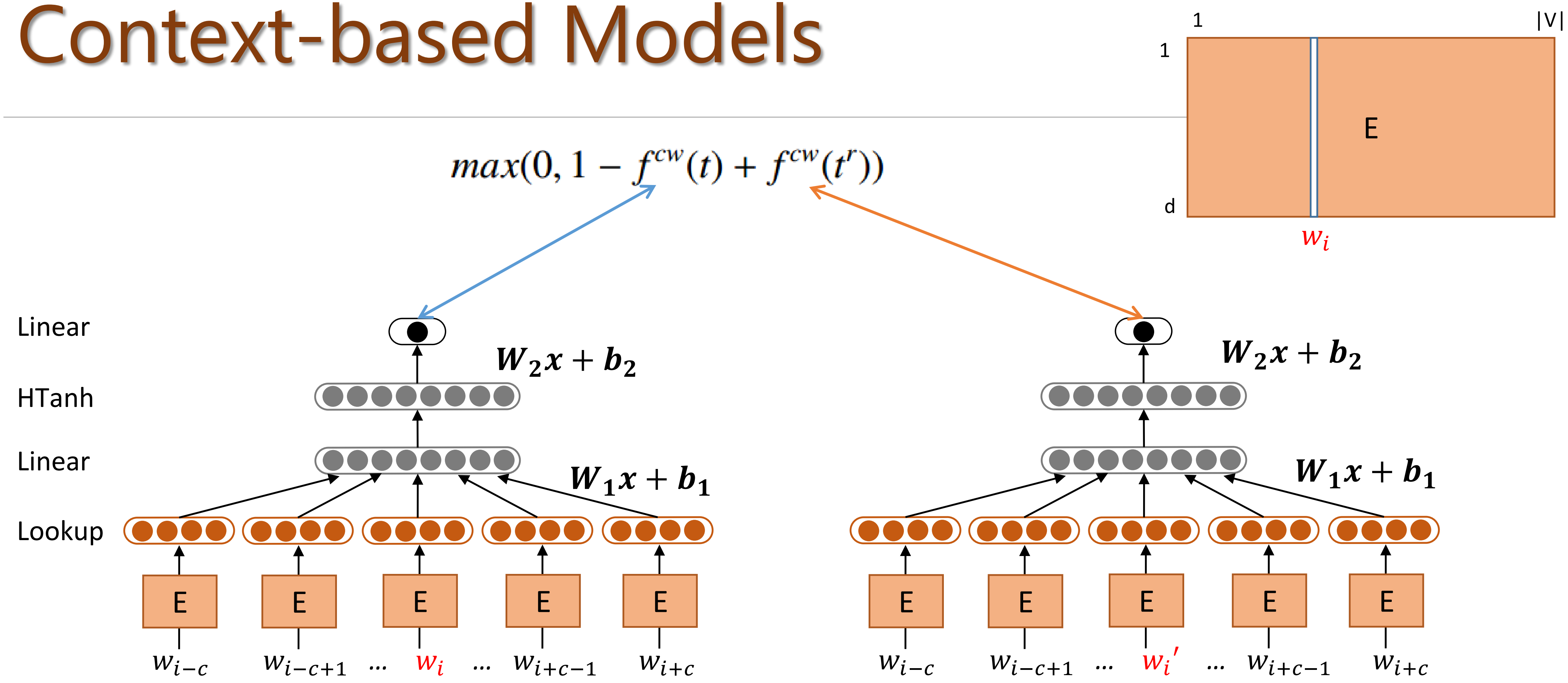
- Objective function

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



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.

Context-based Models



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.

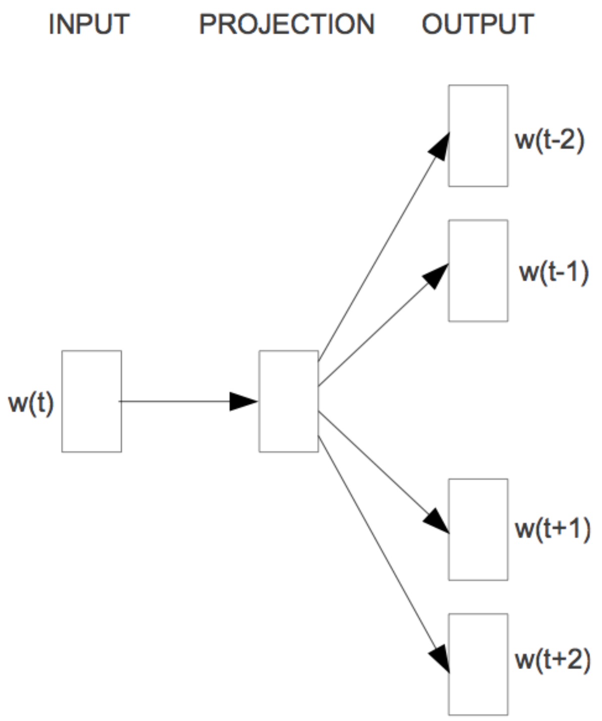
Context-based Models

- Predict based approach
  - word2vec

- Objective function

Loss(context words | target word; Vectors)

$$\frac{1}{T}\sum_{i=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ p(w_{i+j}|e_i)$$

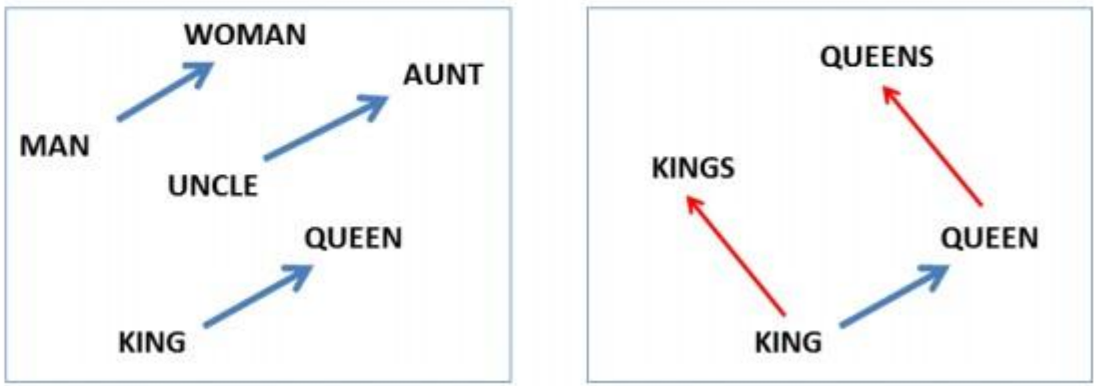


Skip-gram

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.

Linguistic Regularities in Continuous Space Word Representations (Mikolov, et al. 2013)

- Measuring Linguistic Regularity
  - Syntactic/Sementic Test



Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig. 2013. **Linguistic Regularities in Continuous Space Word Representations.** In Proceedings of NAACL 2013

Word Embedding Results (Web 130G)

```
ipad
iphone  0.052583
psp     0.261444
ios     0.285772
xbox    0.286089
zune    0.321024
android 0.330966
imac    0.331299
ipod    0.345187
tablet  0.357396
wii     0.363553
desktop 0.379299
playstation     0.387091
blackberry      0.388451
app     0.406429
pc      0.41485
os      0.43970
smartphone
firefox 0.44461
laptop  0.44913
```

electronic products

```
zemin
qichen  0.104208
jintao  0.11451
rongji  0.128581
hongzhi 0.196312
xiaoping        0.214295
qinghong        0.223624
xiaochuan       0.249642
guangsheng      0.254094
jiechi  0.26547
jiaxuan 0.266882
guangcheng      0.28297
jiabao  0.285498
bangguo 0.289105
wannian 0.292798
enlai   0.294228
```

Bosses of China

```
ceo
cfo     0.193586
founder 0.247986
chairman        0.254936
president       0.305918
owner   0.323968
publisher       0.328711
developer       0.330035
director        0.342627
analyst 0.34414
manager 0.365071
producer        0.376013
cto     0.381128
co-founder      0
coo     0.400879
chief   0.40133
vp      0.429992
```

Bosses of Company

```
cheaper
smaller 0.338234
stronger        0.338306
expensive       0.368454
smarter 0.374804
faster  0.380407
larger  0.401047
profitable      0.404831
safer   0.412357
pricey  0.418853
costlier        0.424498
attractive      0.429141
inexpensive     0.431263
pricier 0.432898
affordable      0
usable  0.438529
cheap   0.448377
```

comparative adjective

```
funny
silly   0.178232
scary   0.179388
weird   0.189012
boring  0.234093
sexy    0.237722
creepy  0.263487
sad     0.264687
crazy   0.271426
awesome 0.28078
cute    0.281509
stupid  0.285263
hilarious       0.291912
curious 0.315346
awkward 0.319954
bizarre 0.334614
ugly    0.340006
```

feeling adjective

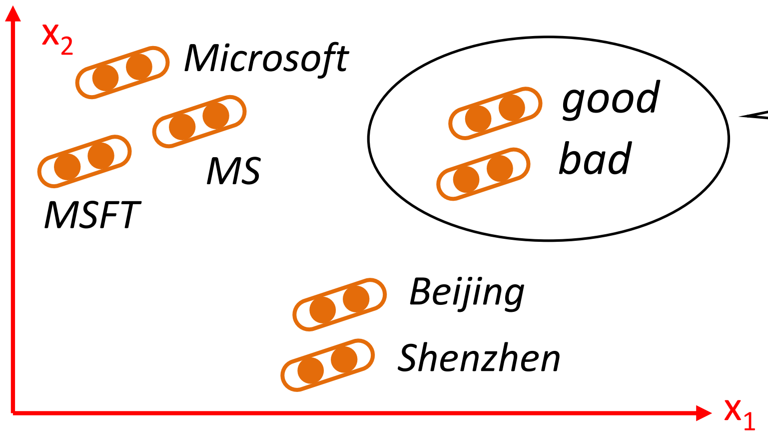
Sentiment-Specific Word Embedding

- Existing embedding learning models are context-based
  - A word is represented by the company it keeps [Firth, J.R. 1959]

… formed the ***good*** *habit of …*

… formed the ***bad*** *habit of …*

Same contexts

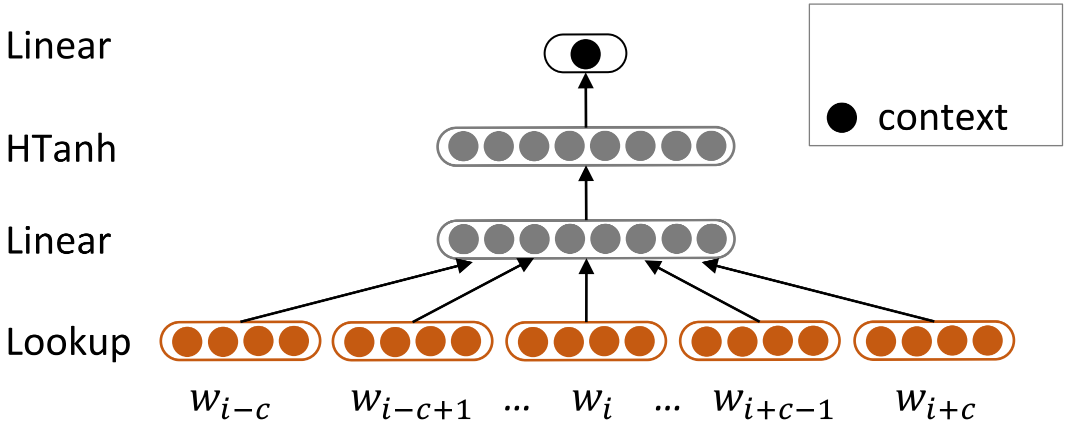


The words with **similar** contexts but **opposite** sentiment polarity are mapped into **close** vectors.

Sentiment-Specific Word Embedding

- The intuition
  - Use contexts of words and sentiment of texts (e.g. sentences)
  - Solution: Incorporate sentiment information into standard context-based approach

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



Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, Ming Zhou. 2016. **Sentiment Embeddings with Applications to Sentiment Analysis**. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Sentiment-Specific Word Embedding

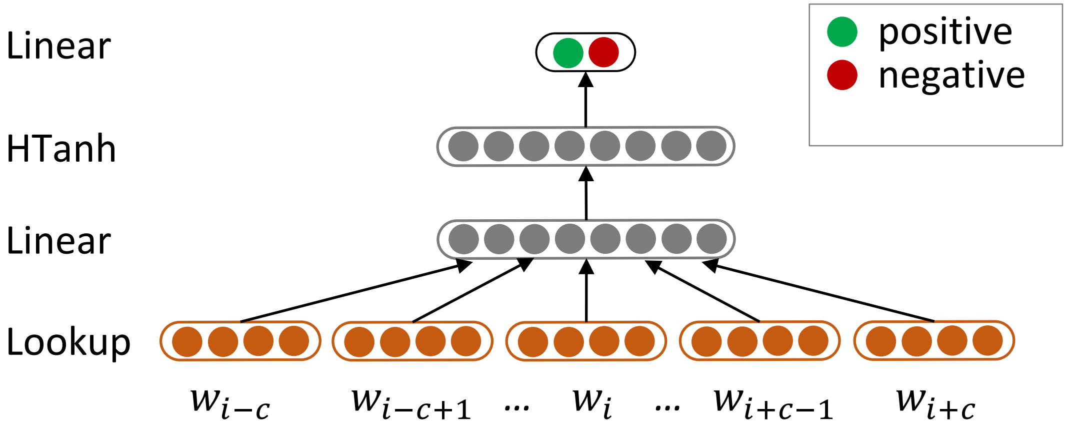
- The intuition
  - Use contexts of words and sentiment of texts (e.g. sentences)
  - Solution: Incorporate sentiment information into standard context-based approach

~~**contexts**~~ $\cancel{loss_{cw}(t, t^r) = max(0, 1 - f^{cw}(t) + f^{cw}(t^r))}$

sentiment

$$loss_s(t) = max(0, 1 - \delta_s(t) f_0^r(t) + \delta_s(t) f_1^r(t))$$

$$\delta_s(t) = \begin{cases} 1 & \text{if } \boldsymbol{f}^g(t) = [1, 0] \\ -1 & \text{if } \boldsymbol{f}^g(t) = [0, 1] \end{cases}$$



Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, Ming Zhou. 2016. **Sentiment Embeddings with Applications to Sentiment Analysis**. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Sentiment-Specific Word Embedding

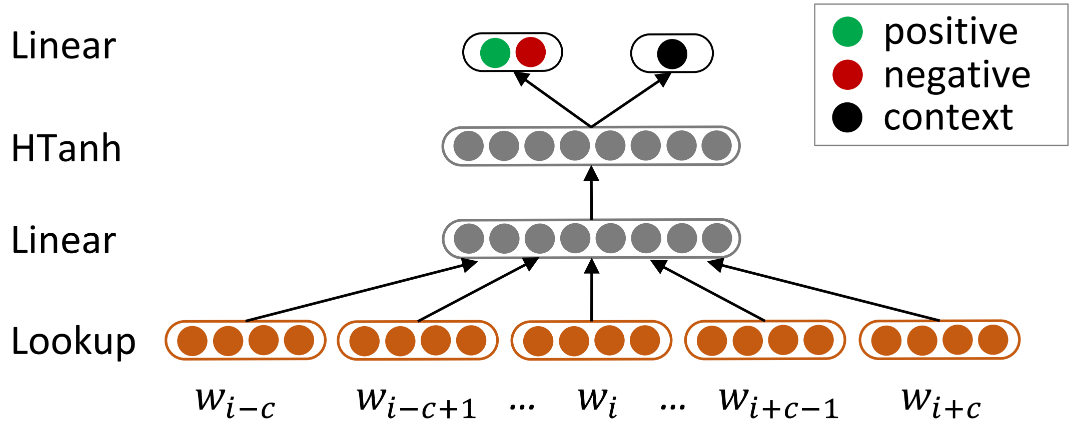
- The intuition
  - Use contexts of words and sentiment of texts (e.g. sentences)
  - Solution: Incorporate sentiment information into standard context-based approach

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

sentiment

$$loss_s(t) = max(0, 1 - \delta_s(t) \boldsymbol{f}_0^r(t) + \delta_s(t) \boldsymbol{f}_1^r(t))$$

$$\delta_s(t) = \begin{cases} 1 & \text{if } \boldsymbol{f}^g(t) = [1,0] \\ -1 & \text{if } \boldsymbol{f}^g(t) = [0,1] \end{cases}$$



contexts + sentiment $$loss(t,t^r) = \alpha\, loss_{cw}(t,t^r) + (1-\alpha)\, loss_s(t)$$

Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, Ming Zhou. 2016. **Sentiment Embeddings with Applications to Sentiment Analysis**. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Model Training

- Use emotions/smileys as sentiment signals to collect massive tweets as training data
  - We use 5 positive emoticons, 3 negative emoticon [Hu et al. 2013]
  - 5 million positive and 5 million negative tweets from April, 2013

Positive Emoticons	:)	:)	:-)	:D	=)
Negative Emoticons	:(	: (	:-(		

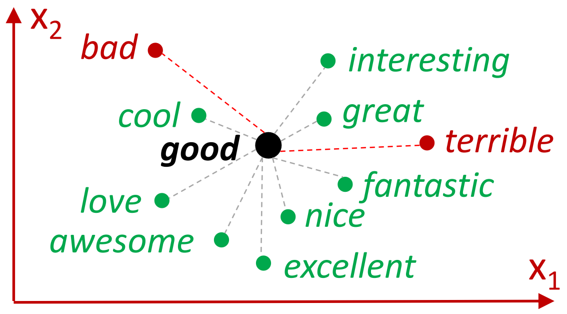
- Parameter Learning
  - Back-propagation, SGD

Querying Similar Words

- good
  - sweet
  - favorite
  - cool
  - movie
  - excited
  - amazing
  - awesome
  - well
  - love
  - great
  - favourite
  - happy
- bad
  - cry
  - wrong
  - hard
  - alone
  - annoying
  - hate
  - tired
  - lost
  - happened
  - pain
  - sorry
  - jealous
  - mad

Querying Similar Words

- Find top K nearest neighbors in the embedding space, and calculate the accuracy of sentiment consistency



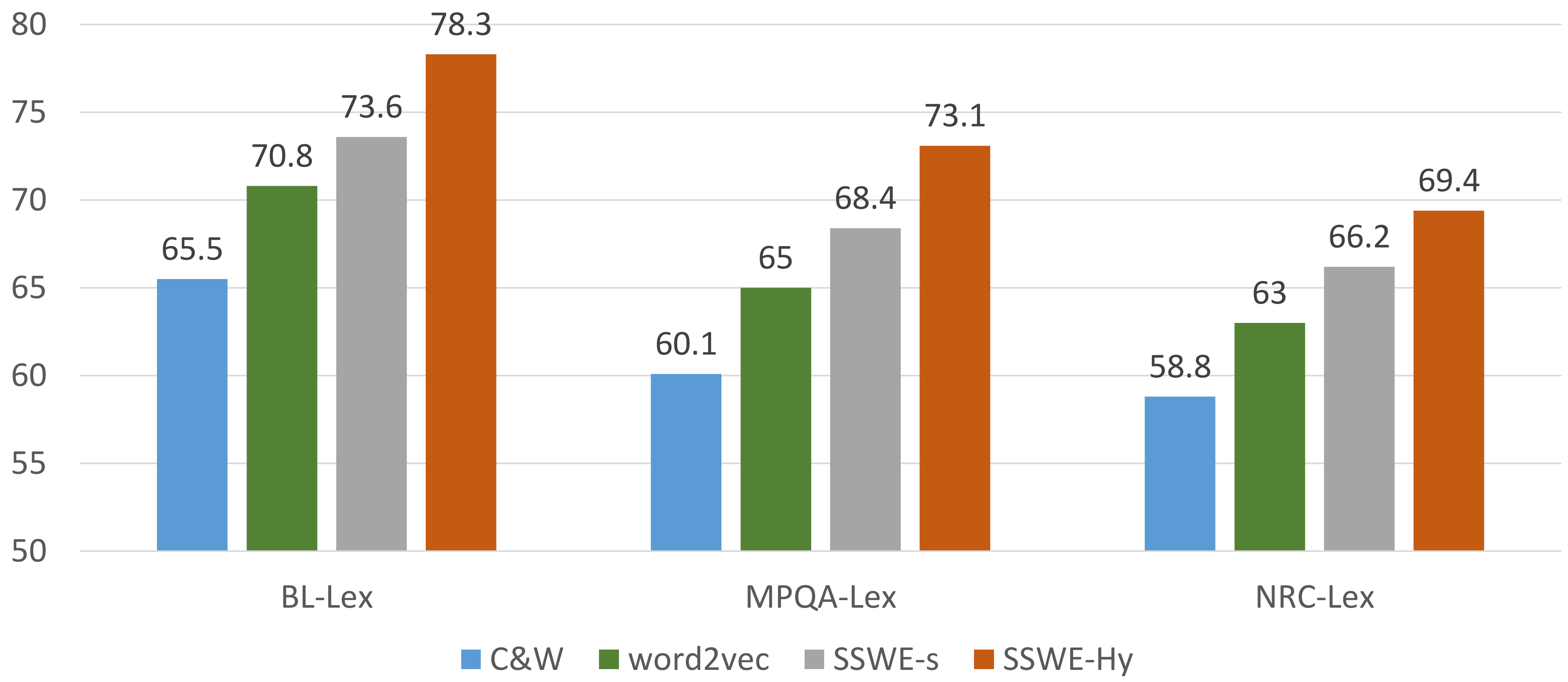
$$Accuracy = \frac{\sum_{i=1}^{\#Lex} \sum_{j=1}^{N_w} \delta_w(w_i, c_{ij})}{\#Lex \times N_w}$$

- We conduct experiments on existing sentiment lexicons

Lexicon	#Positive	#Negative	#Total
BL-Lex	2,006	4,780	6,786
MPQA-Lex	2,301	4,150	6,451
NRC-Lex	2,231	3,324	5,555

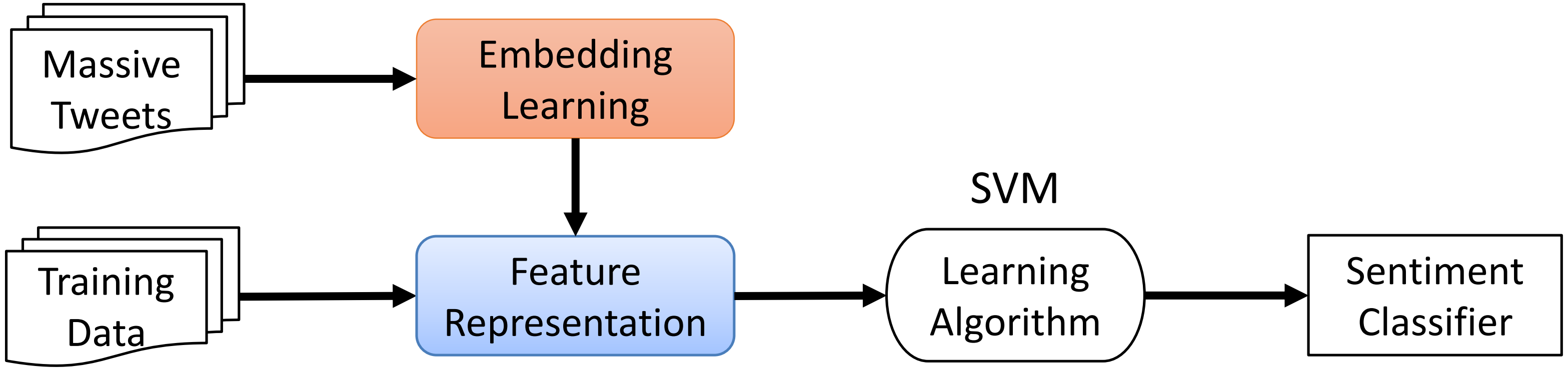
Querying Similar Words

Experimental results



Twitter Sentiment Classification

- Determine the sentiment polarity of a tweet



- Run experiment on benchmark dataset in SemEval 2013

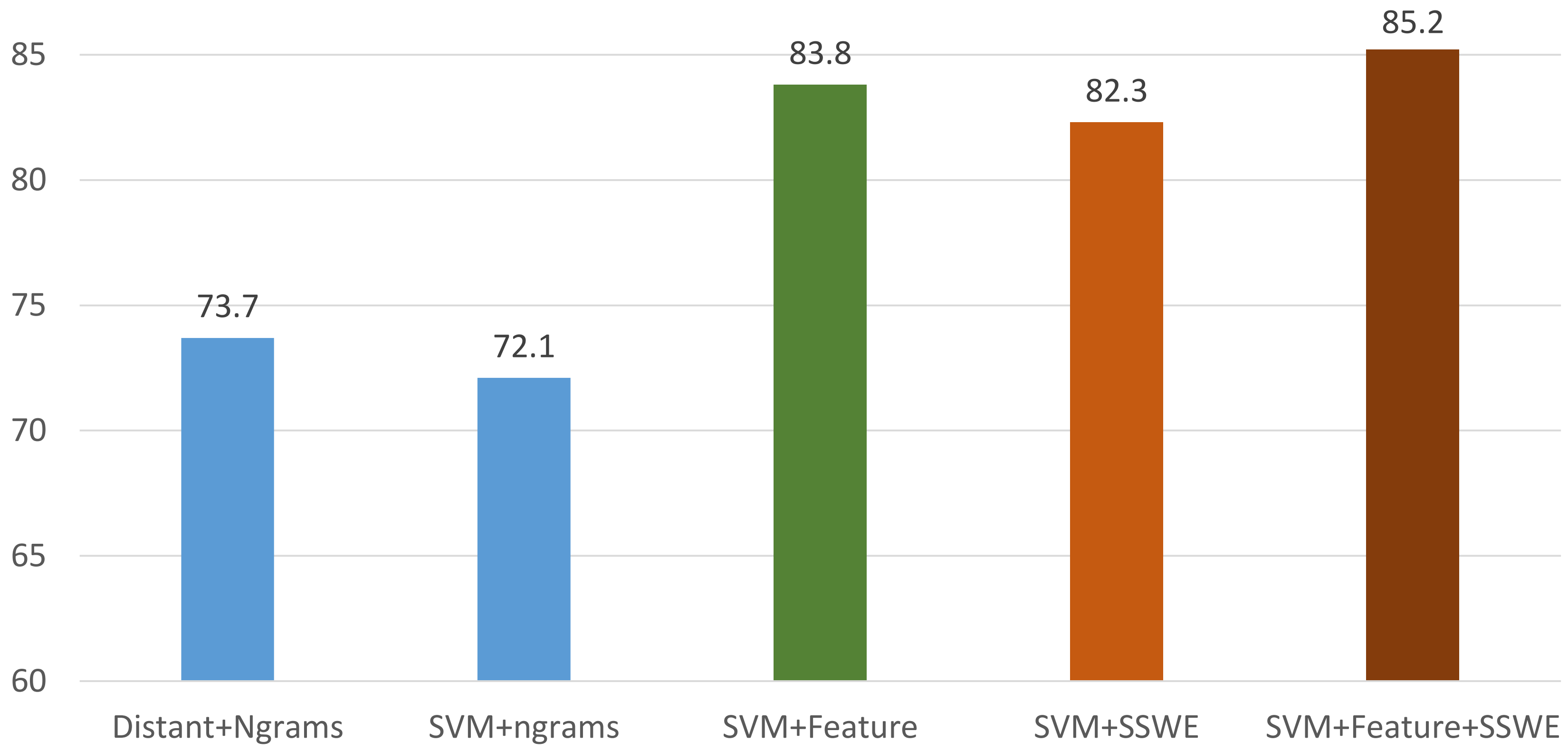
Dataset	#Positive	#Negative	#Total
Training	2,642	994	3,636
Development	408	219	627
Test	1,570	601	2,171

From word vector to tweet vector

- Each word is represented by a 50-dimension vector
- Each sentence/tweet is represented by a 150-dimension vector (50 dimensions for mean, 50 dimensions for max, 50 dimensions for min)
- Optimal: from words to phrases
  - Learn embeddings for ngrams similar to unigrams

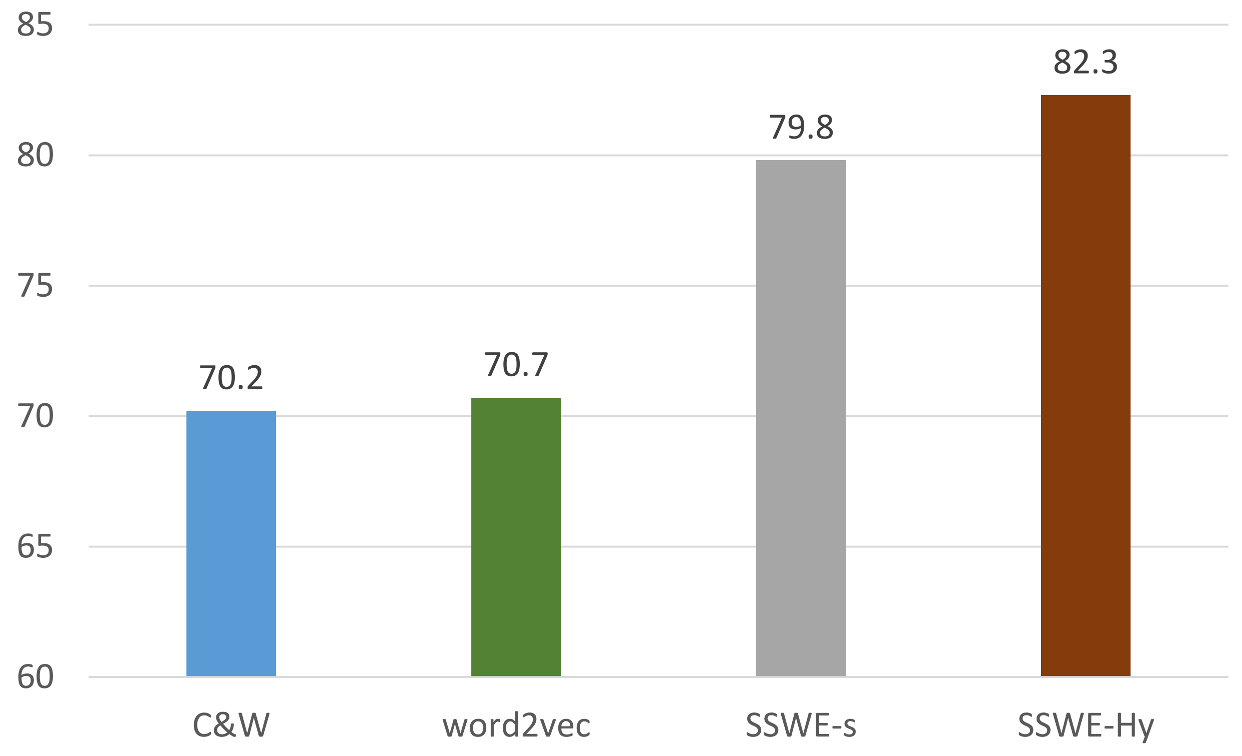
Twitter Sentiment Classification

- Compare with different classification algorithms



Twitter Sentiment Classification

- Compare with different embedding learning algorithms

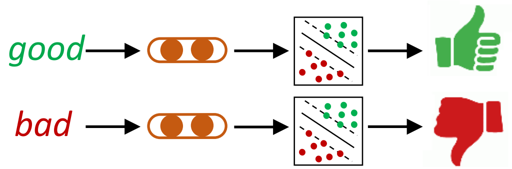


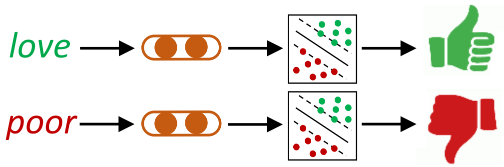
Building Sentiment Lexicon

- A sentiment lexicon is a list of words, each of which is assigned with a positive/negative score

Positive words	Negative words
excellent (0.99); awesome (0.98); good (0.97)	*bad (-0.98); poor (-0.97); awful (-0.76)*

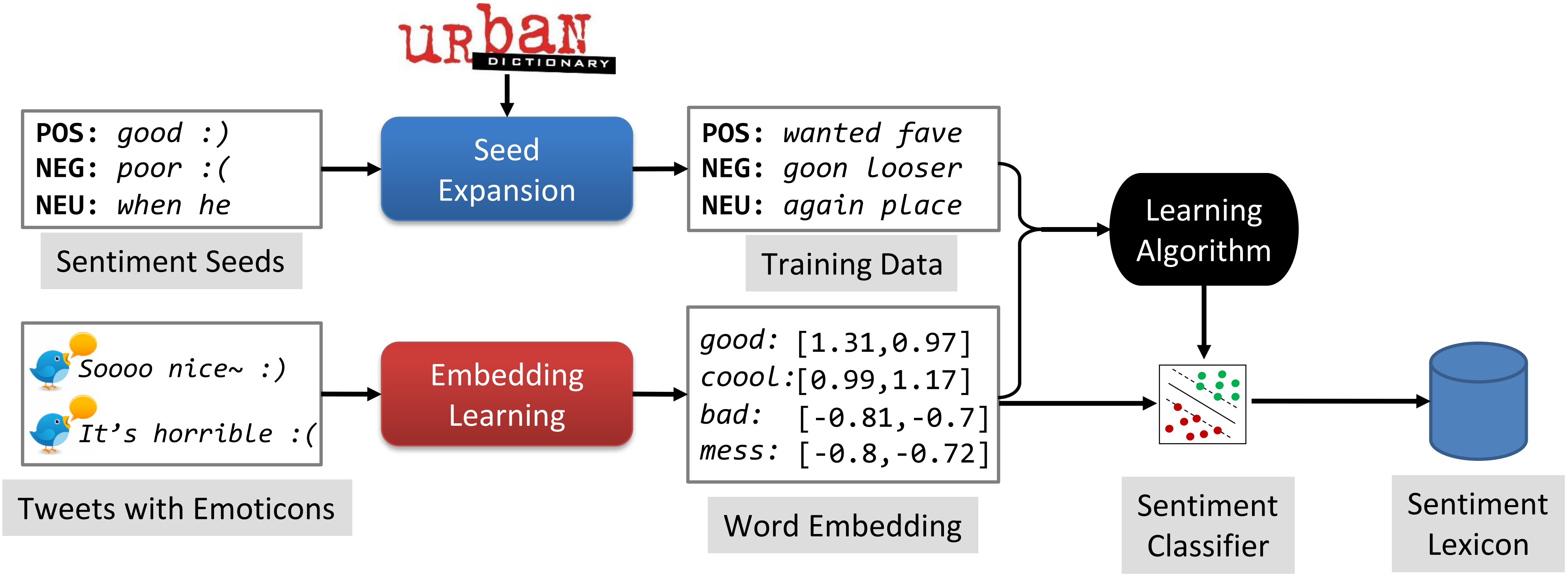
- We treat lexicon construction as a classification problem
  - Train a word level sentiment classifier by regarding word embedding as features





Building Sentiment Lexicon

- The framework



Building Sentiment Lexicon

- Lexicon scale

Lexicon	#Positive	#Negative	#Total
BL-Lex	2,006	4,780	6,786
MPQA-Lex	2,301	4,150	6,451
NRC-Lex	2,231	3,324	5,555
HashtagLex	32,048	22,081	54,129
Sentiment140Lex	38,312	24,156	62,468
Our Lexicon	**31,591**	**33,012**	**64,603**

Manually labeled

Automatically generated

Building Sentiment Lexicon

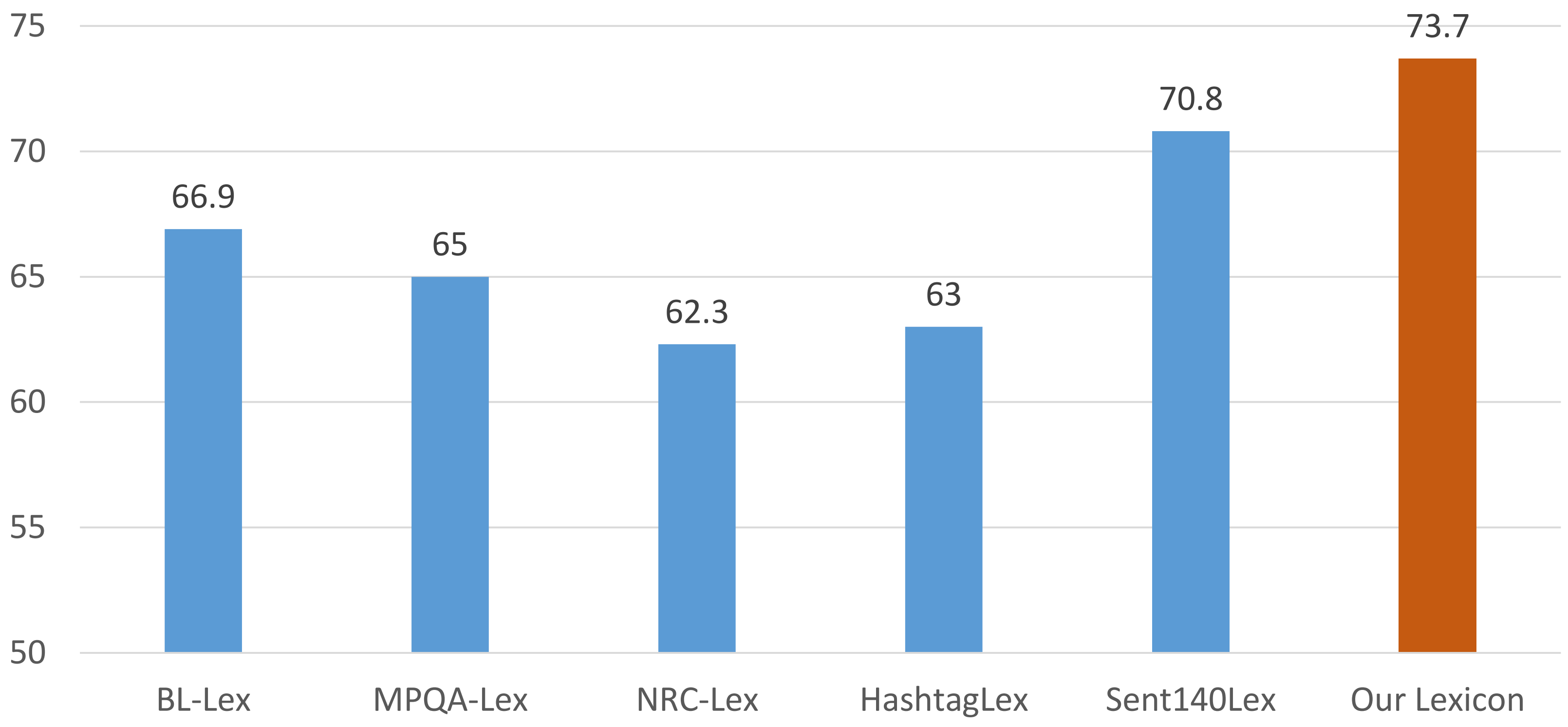
- Applying sentiment lexicon as features to Twitter sentiment classification

- Feature templates
  - total count of tokens in the tweet with score greater than 0;
  - the sum of the scores for all tokens in the tweet;
  - the maximal score;
  - the non-zero score of the last token in the tweet;

Saif Mohammad, Svetlana Kiritchenko, Xiaodan Zhu. 2013. **NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets**. In *SemEval 2013.*

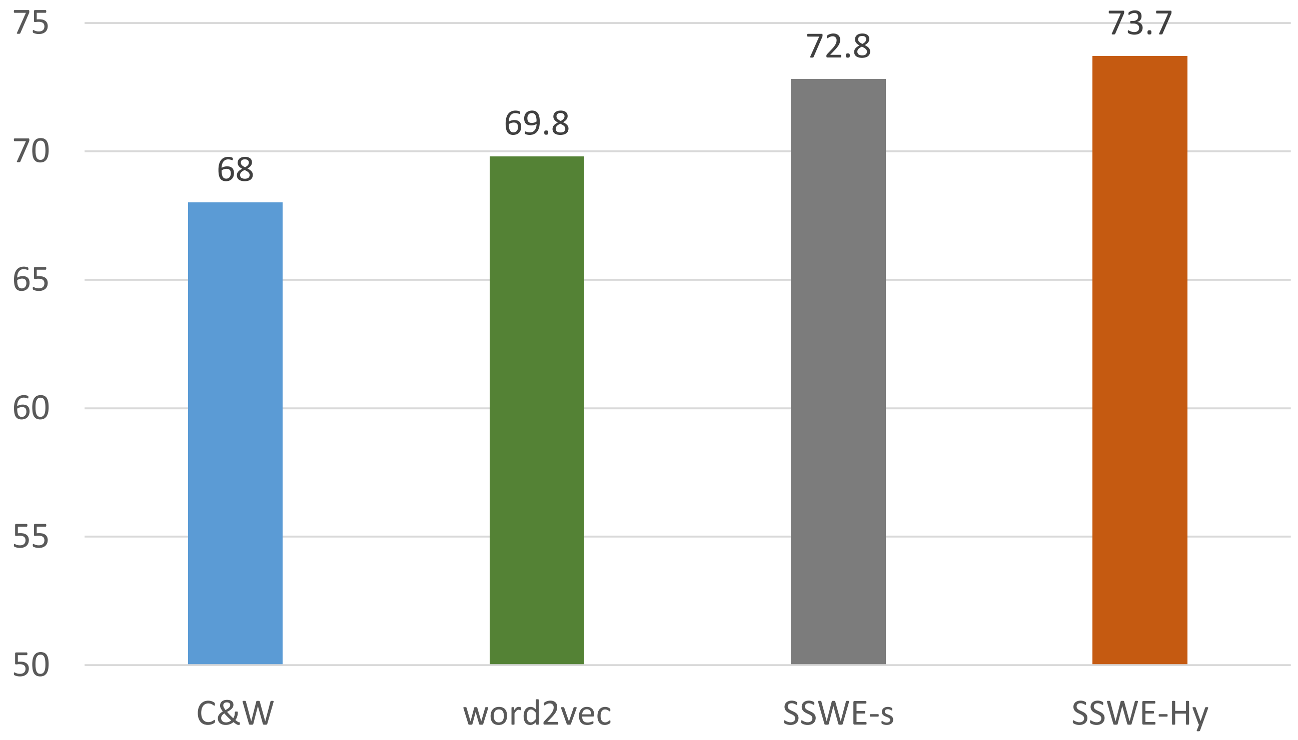
Building Sentiment Lexicon

- Compare with different sentiment lexicons



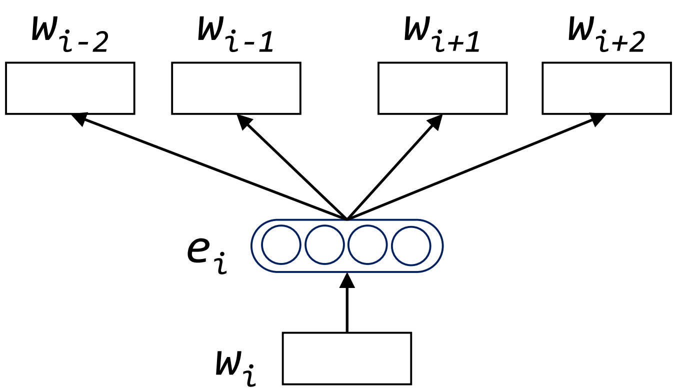
Building Sentiment Lexicon

- Compare with different embedding learning algorithms



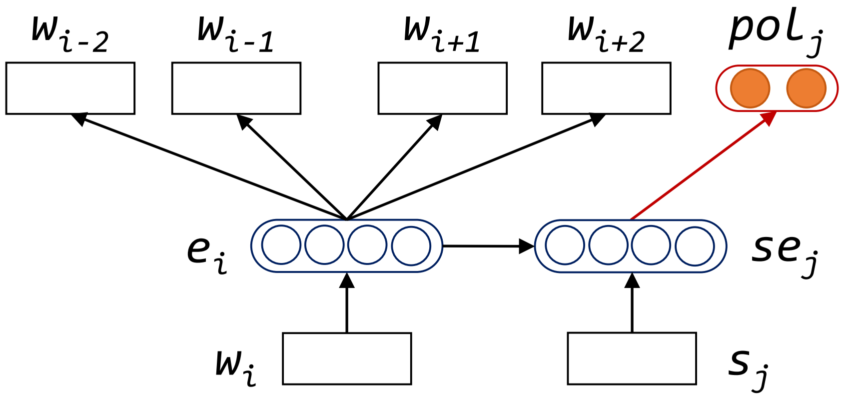
Extend SkipGram

- Extend SkipGram model to encode sentiment information



SkipGram

$$f_{syntactic} = \frac{1}{T} \sum_{i=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} log\ p(w_{i+j} | e_i)$$



+ Sentiment

$$\alpha \cdot f_{syntactic} + (1 - \alpha) \cdot \frac{1}{S} \sum_{j=1}^{S} log\ p(pol_j | se_j)$$

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, Ming Zhou. 2014. **Building Large-Scale Twitter-Specific Sentiment Lexicon : A Representation Learning Approach**. *International Conference on Computational Linguistics(COLING)*.

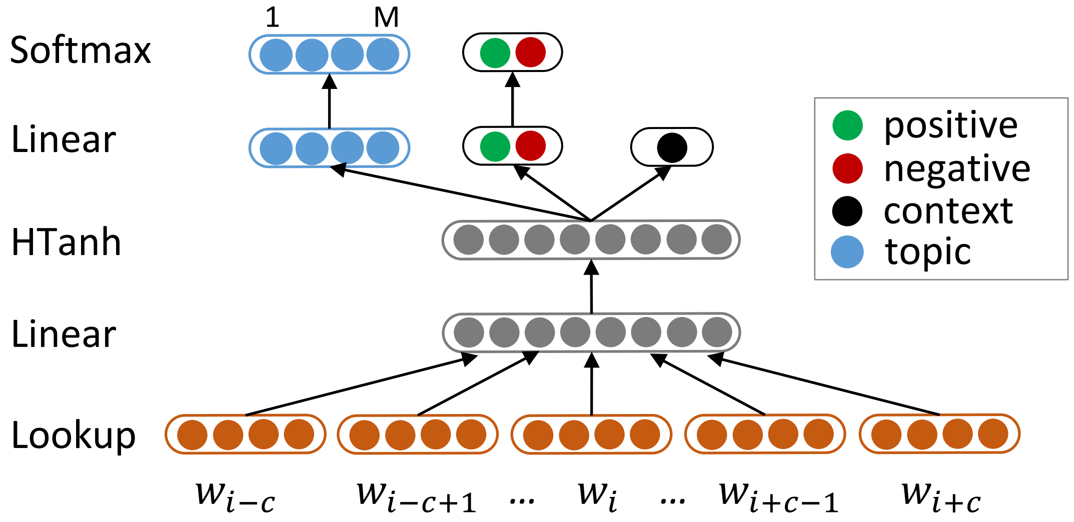
Extension on SSWE

- Incorporate topic information
  - predicting the topic distribution of text based on input n-grams
  - the topic distribution is generated using LDA (Blei et al., 2003)

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

sentiment $$loss_{sen} = -\sum_{j=1}^{K} f_j^{sen}(t) * log(g_j^{sen}(t))$$

topic $$\begin{cases} f^{top}(t) = softmax(W_t * a + b_t) \\ loss_{top} = -\sum_{i=1}^{M} f_i^{top}(t) * log(g_i^{top}(t)) \end{cases}$$



contexts + sentiment + topic $$loss(t,t^r) = \alpha\, loss_{cw}(t,t^r) + \beta\, loss_s(t) + (1 - \alpha - \beta)\, loss_{top}(t)$$

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. **Improving Twitter Sentiment Classification Using Topic-Enriched Multi-Prototype Word Embeddings**. In Proceedings of *AAAI*.

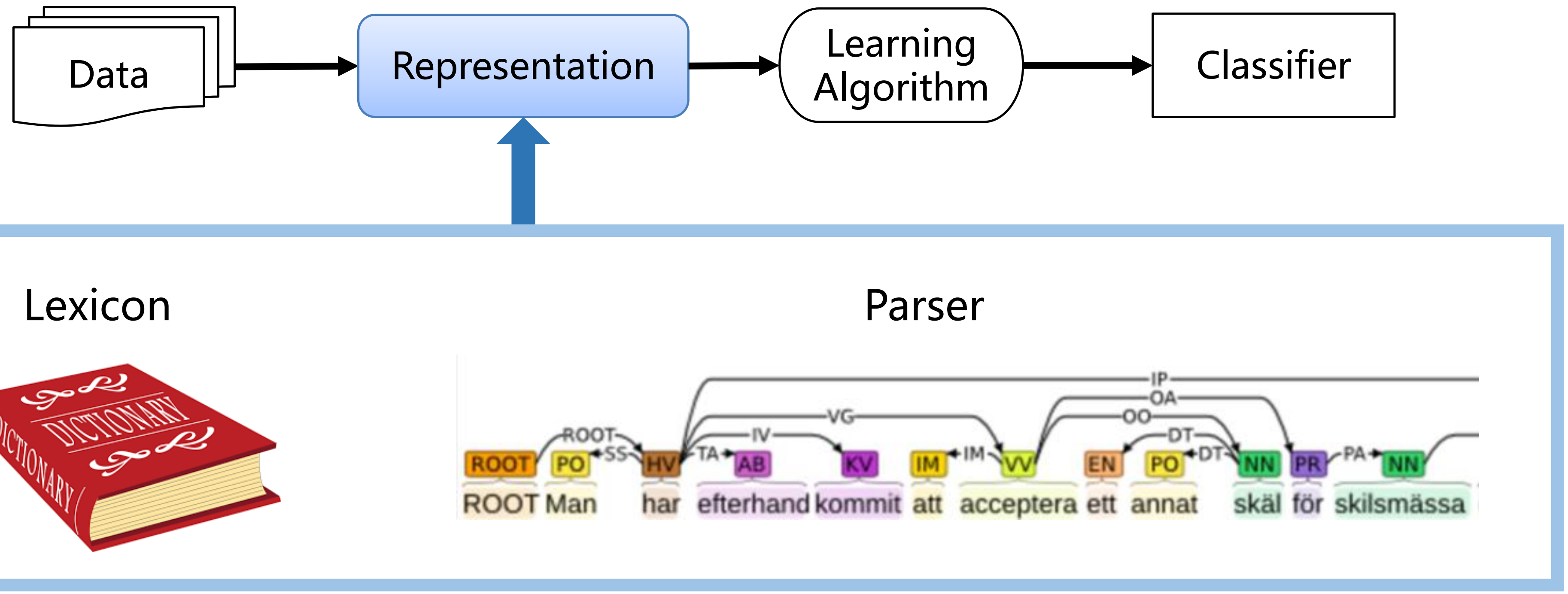
Aspect Level Sentiment Classification

- Task definition
  - Input：Sentence + Aspect
  - Output: The sentiment of the sentence towards the aspect

Sentence	Aspect	Polarity
great food but the service was dreadful	food	positive
great food but the service was dreadful	service	negative

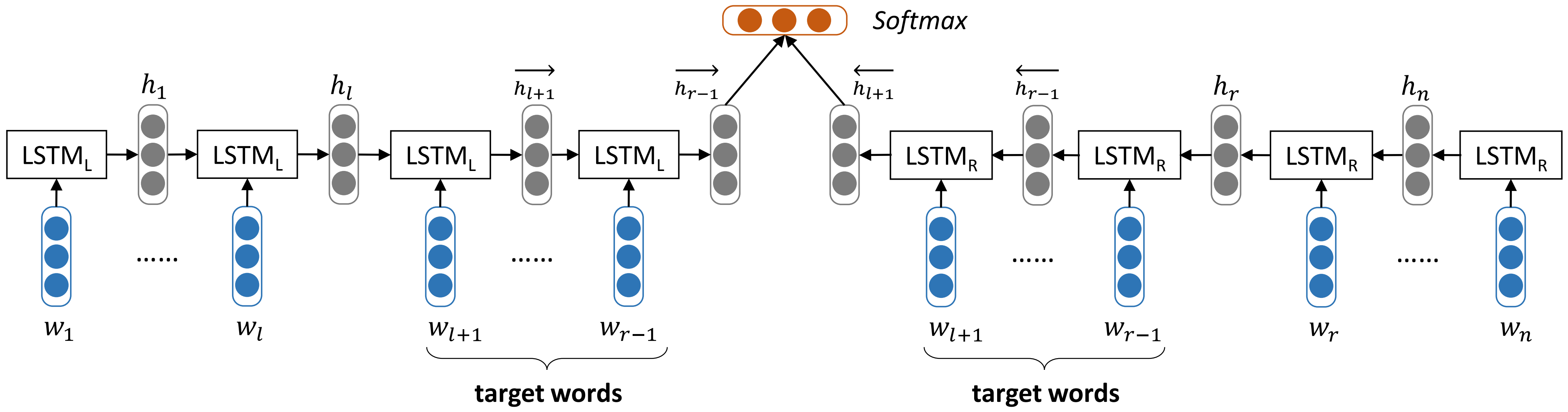
Existing Solutions

- Feature based SVM
  - Cons: Rely on feature engineering, ….



Existing Solutions

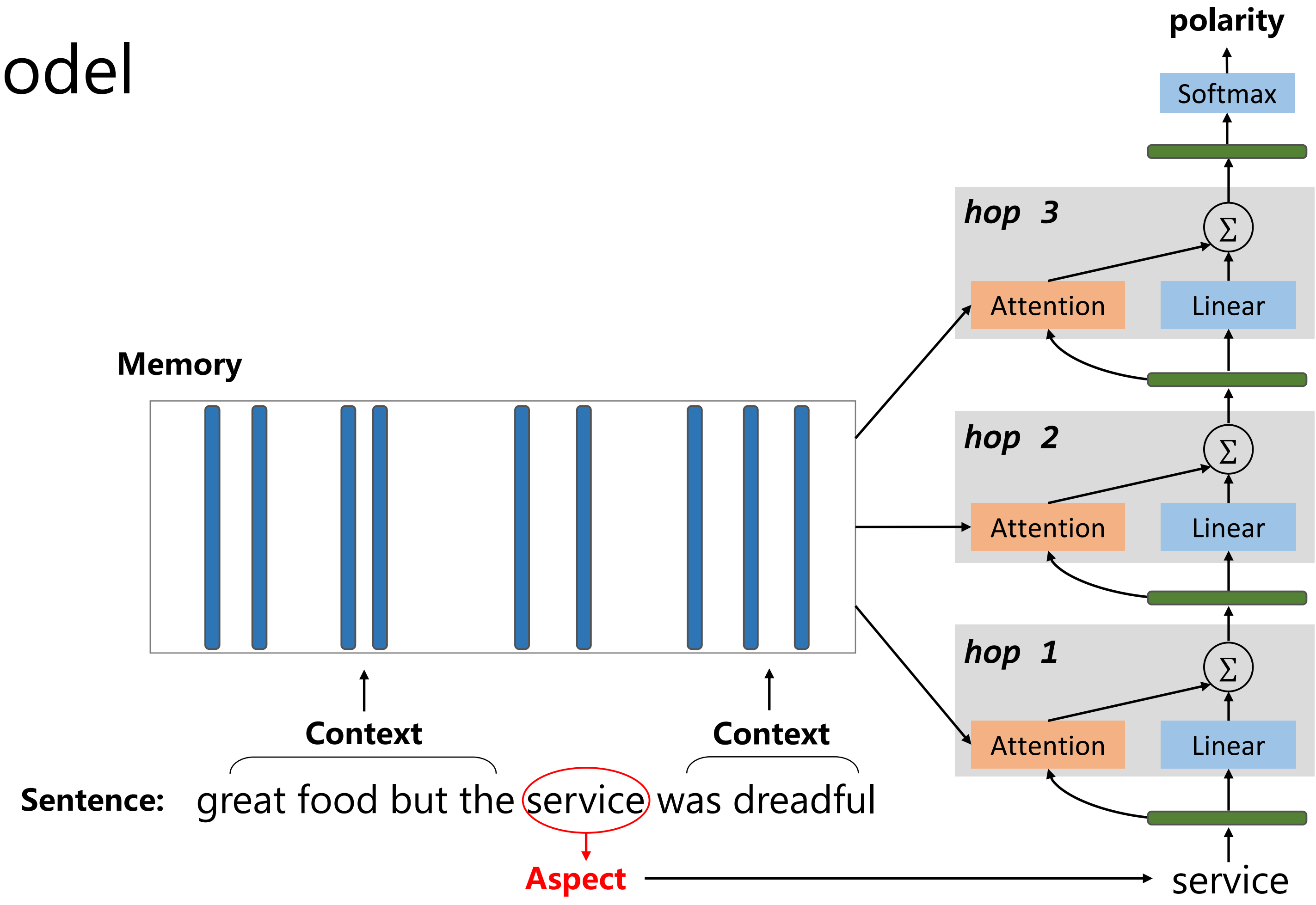
- LSTM RNN
  - Pros: Learning from data
  - Cons: Could not explicitly reveal the importance/contribution of context words with regard to the aspect



Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu. 2016. **Target-Dependent Sentiment Classification with Long Short Term Memory** . *http://arxiv.org/abs/1512.01100*.

Deep Memory Network

- The model



Duyu Tang, Bing Qin, Ting Liu. 2016. **Aspect Level Sentiment Classification with Deep Memory Network** . *Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

Content based Attention

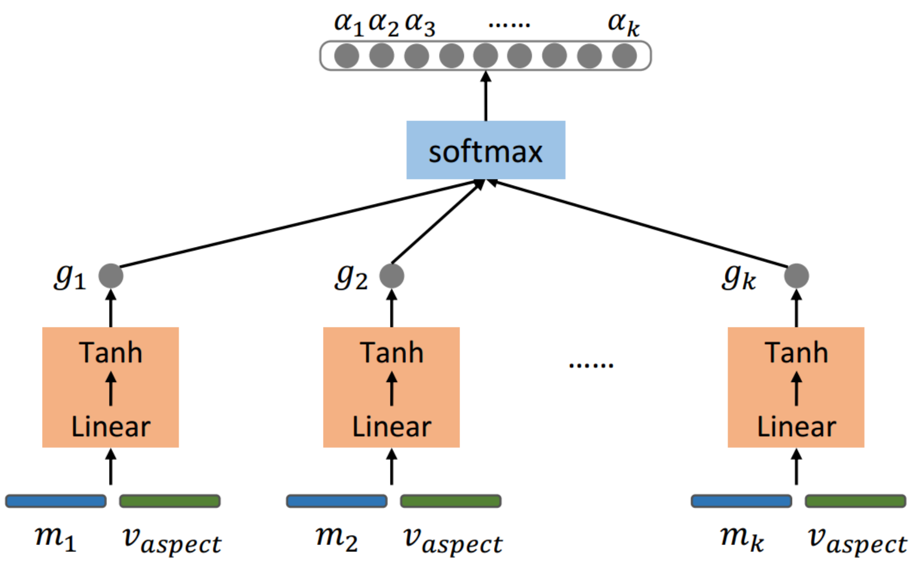
- Calculate vec based on the representation of each piece of memory m_i

$$vec = \sum_{i=1}^{k} \alpha_i m_i$$

- Calculate the attention weights α

$$g_i = tanh(W_{att}[m_i; v_{aspect}] + b_{att})$$

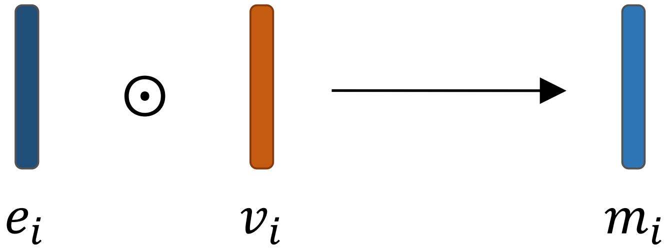
$$\alpha_i = \frac{exp(g_i)}{\sum_{j=1}^{k} exp(g_j)}$$



Location Enhanced Attention

- Each memory cell m_i is calculated by elementwise multiplication between word vec e_i and location vec v_i

$$v_i = 1 - l_i/n$$



Model Training

- Supervised Learning, minimize cross-entropy error

$$loss = -\sum_{(s,a)\in T} \sum_{c\in C} p_c(s,a) \cdot log(p_c^*(s,a))$$

- Parameter Learning
  - Use Glove vector, clamp the values
  - Back-propagation, SGD

Experimental Setting

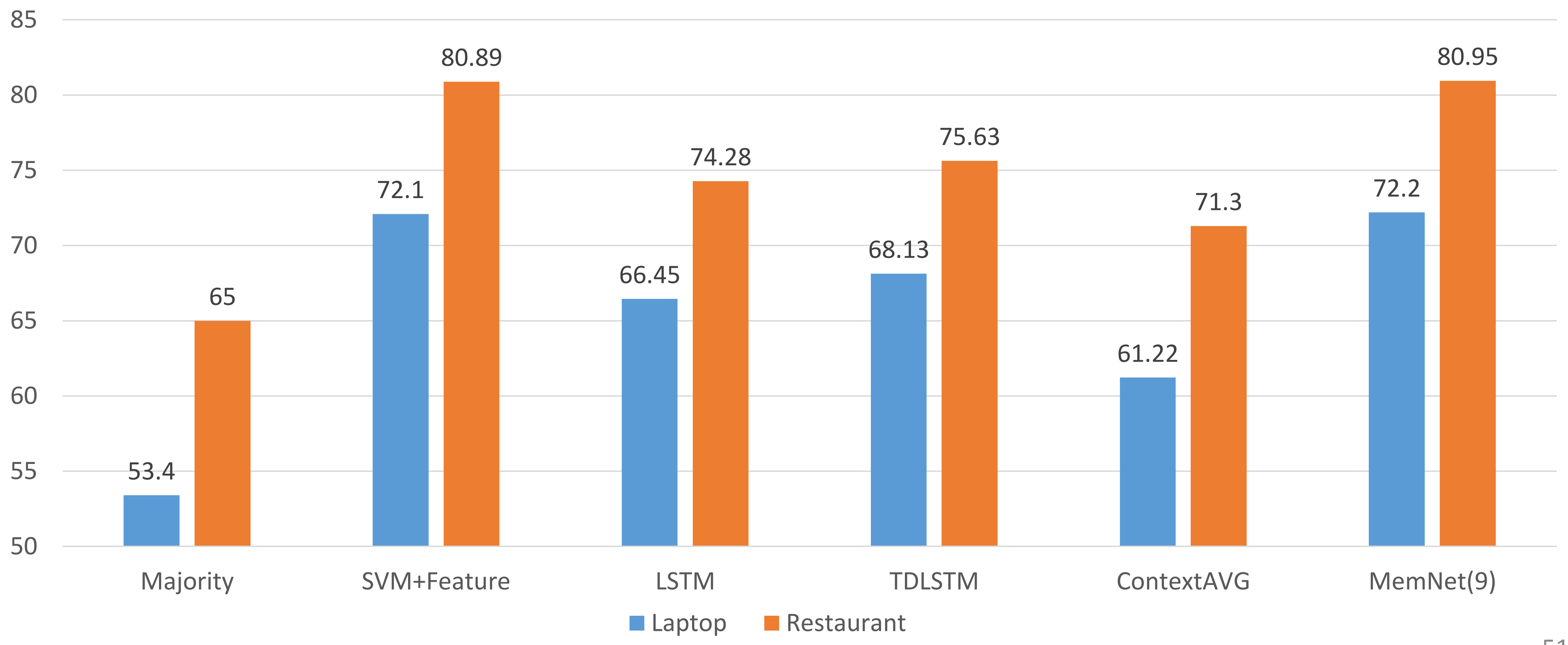
- We use two datasets from SemEval 2014

Dataset	Pos.	Neg.	Neu.
Laptop-Train	994	870	464
Laptop-Test	341	128	169
Restaurant-Train	2164	807	637
Restaurant-Test	728	196	196

- Evaluation metric: classification accuracy

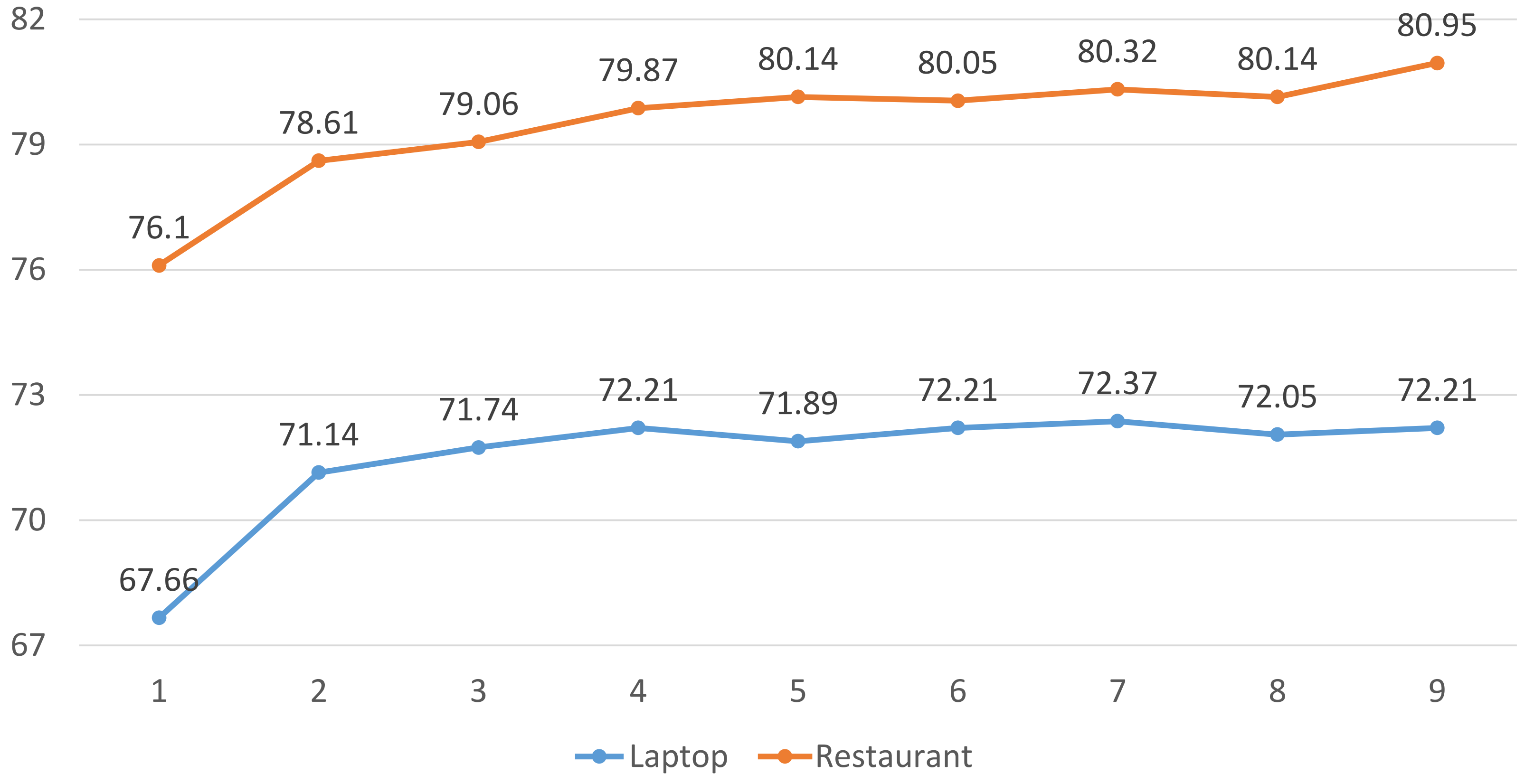
Results

- Compare with different classification algorithms



Results

The influence of the number of hops



Visualize the Attention Weights

- great food but the service was dreadful

Aspect: food, Answer: +1, Prediction: -1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.22	0.12	0.14	0.12	0.20
but	0.21	0.11	0.10	0.11	0.12
the	0.03	0.11	0.08	0.11	0.06
service	0.11	0.11	0.08	0.11	0.06
was	0.04	0.11	0.08	0.11	0.06
dreadful	0.22	0.32	0.45	0.32	0.43
!	0.16	0.11	0.08	0.11	0.07

Content-based Attention

Aspect: food, Answer: +1, Prediction: +1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.31	0.26	0.32	0.28	0.32
but	0.14	0.18	0.15	0.18	0.15
the	0.08	0.05	0.08	0.05	0.07
service	0.09	0.09	0.09	0.08	0.09
was	0.09	0.08	0.09	0.08	0.08
dreadful	0.18	0.21	0.18	0.22	0.19
!	0.11	0.12	0.10	0.11	0.10

Location-enhanced Attention

Document Level Sentiment Classification

- Task Definition
  - Input: A piece of document
  - Output: The overall sentiment/polarity expressed in the doc

  - Sentiment
    - positive/negative 
    - 1-5 stars 

I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. Despite it is a little expensive, I love it.

Lexicon based Approach

- Basic idea
  - Use the dominant polarity of the opinion words in the document to determine its polarity
  - If positive/negative opinion prevails, the opinion document is regarded as positive/negative
- **Lexicon + Counting**
- **Lexicon + Grammar Rule + Inference Method**

Minqing Hu and Bing Liu. **Mining and summarizing customer reviews**. *KDD*: 168-177, 2004.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. **Lexicon-Based Methods for Sentiment Analysis**. *Computational Linguistics*: 37(2), 267-307. 2011.

Feature based SVM

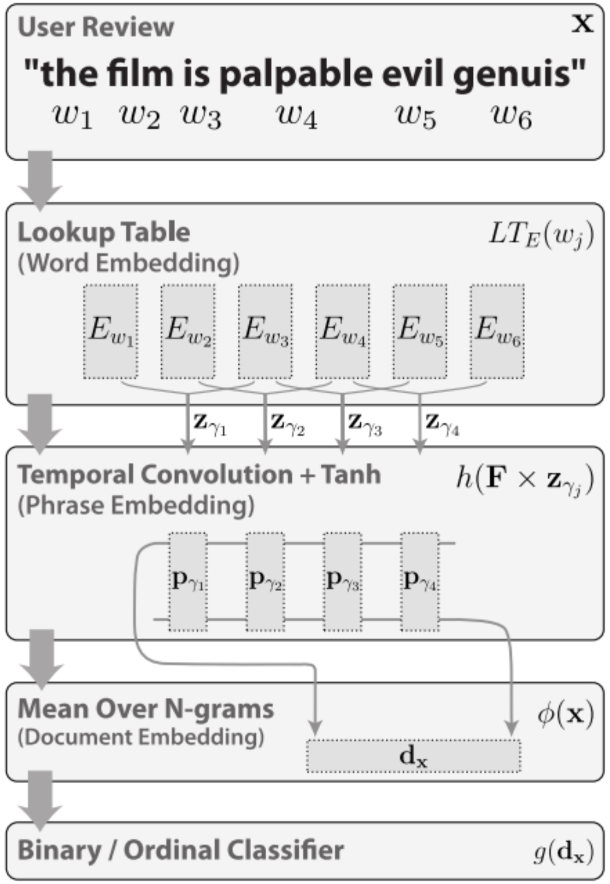
- Basic idea
  - Treat sentiment classification simply as a special case of topic-based categorization
  - With the two "topics" being positive sentiment and negative sentiment
  - Use machine learning approach (e.g. SVM/NB) + features

- Pang et al. (2002) show that **SVM + bag-of-word** feature performs well.
  - A very strong baseline for doc-level sentiment classification.

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan. **Thumbs up? Sentiment Classification using Machine Learning Techniques**. *EMNLP, 2002*.

Latent N-Gram Analysis

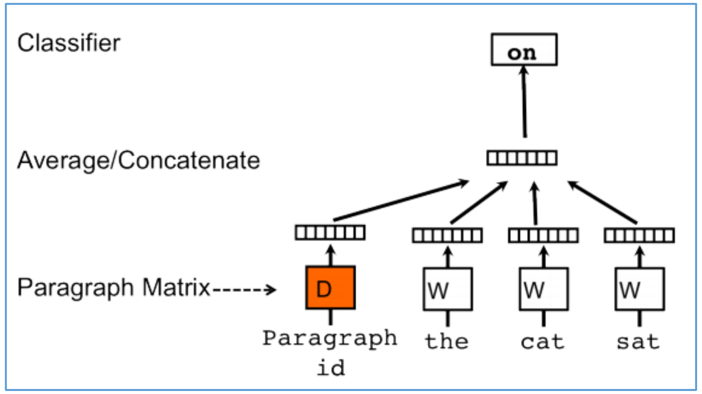
- Basic idea
  - Project n-gram to low-dimensional latent semantic space
  - Word -> Phrase -> Document
  - End-to-End training with SGD

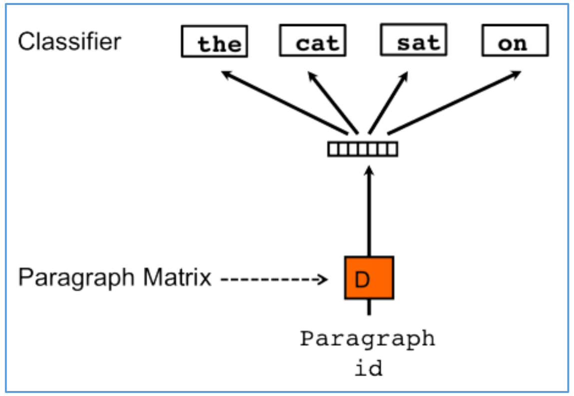


Dmitriy Bespalov, Bing Bai, Yanjun Qi, Ali Shokoufandeh. **Sentiment Classification Based on Supervised Latent n-gram Analysis**. *Proceedings of the Conference on Information and Knowledge Management, 2011*.

Paragraph Vector

- Basic idea
  - represents each document by a dense vector which is trained to predict words in the document
  - Motivation
    - bag-of-words features have two major weaknesses: **they lose the ordering of the words** and **they also ignore semantics of the words**

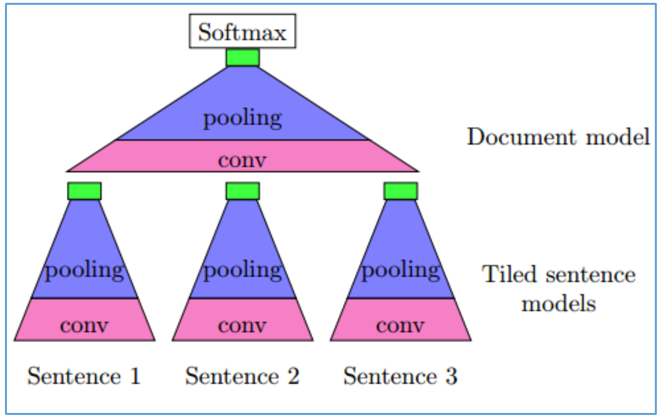


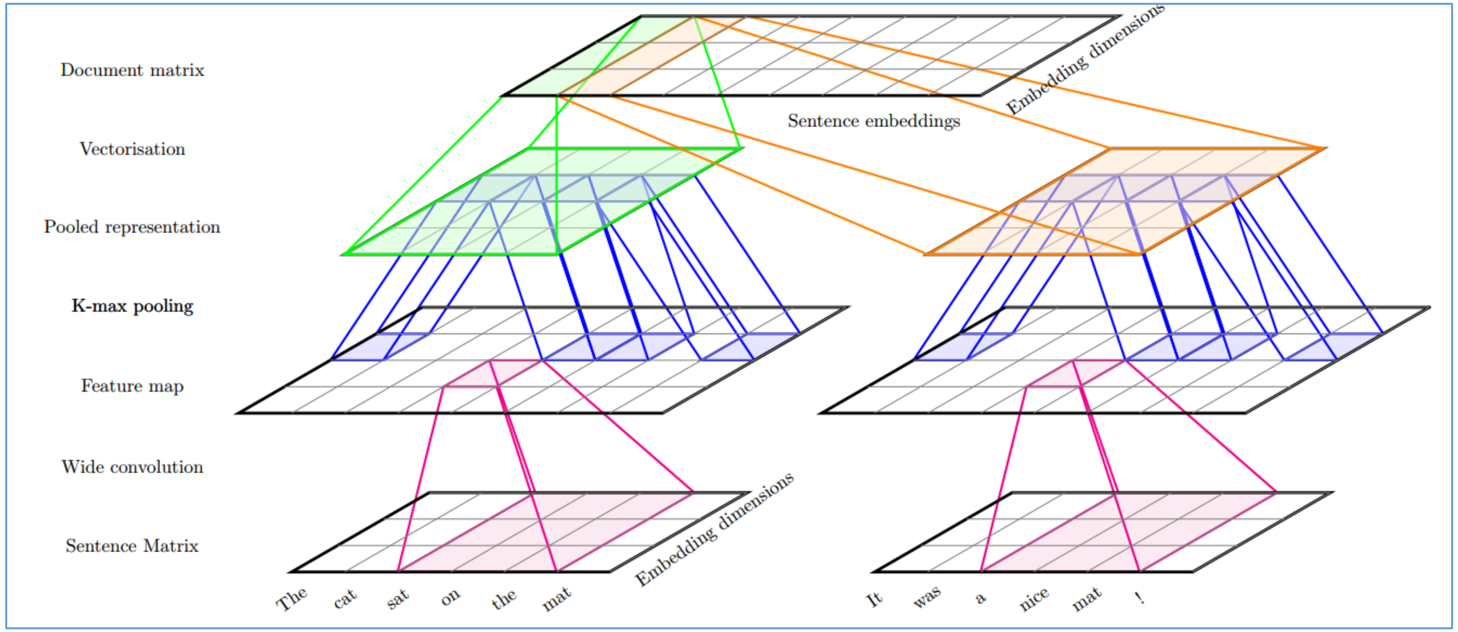


Quoc Le, Tomas Mikolov. **Distributed Representations of Sentences and Documents.** *In ICML 2014*

Convolution NN

- Basic idea
  - Word -> Sentence -> Document





Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, Nando de Freitas. **Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network.** *arxiv.org. 1406.3830*

Hierarchical NN

- A human writes and reads an article in a hierarchical way.

I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. Despite it is a little expensive, I love it.

Human writing

Thought
↓
Document
↓
Sentence
↓
Word

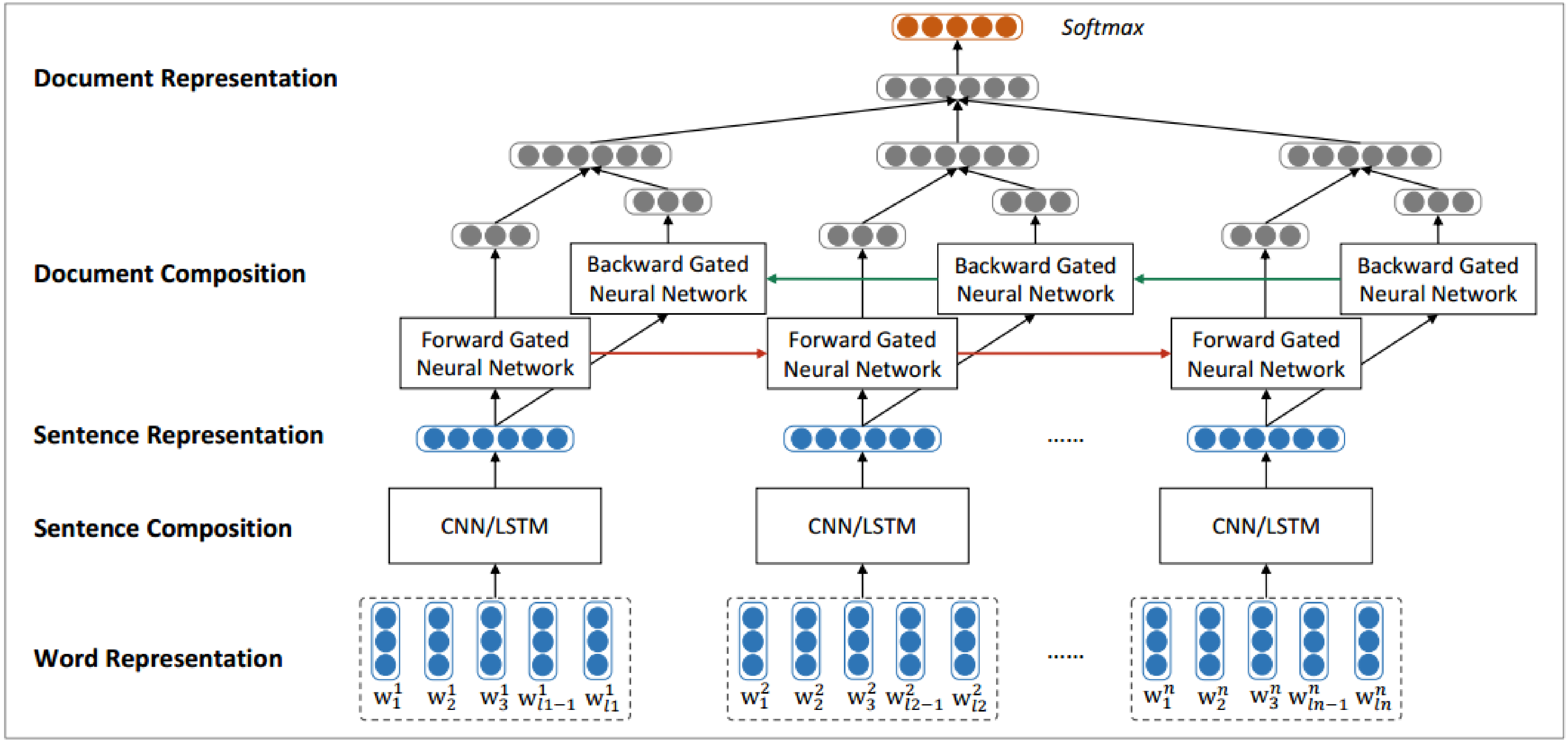
Human reading

Thought
↑
Document
↑
Sentence
↑
Word

Machine reading

Thought
↑
Document
↑
Sentence
↑
Word

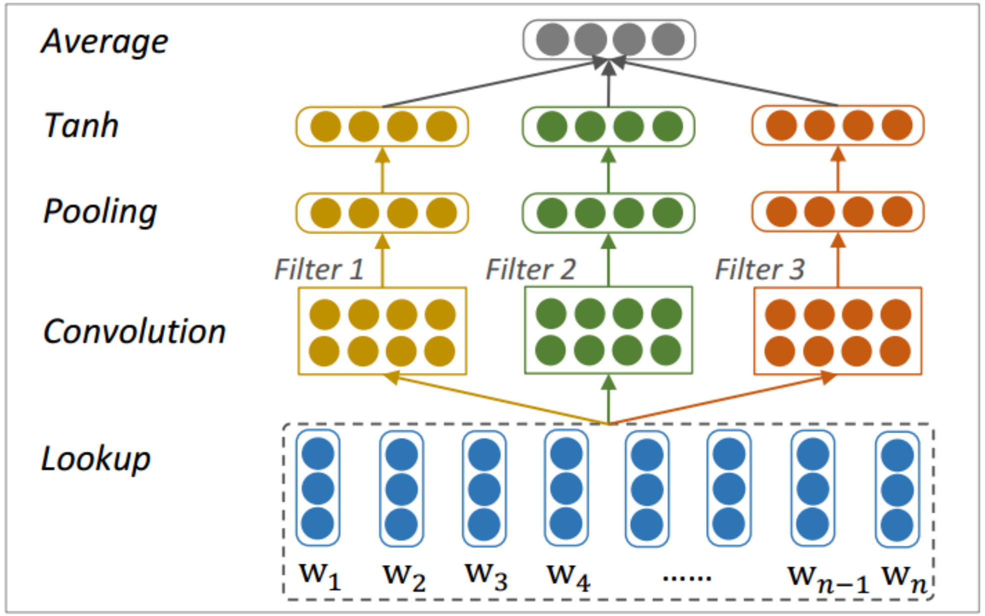
Hierarchical NN

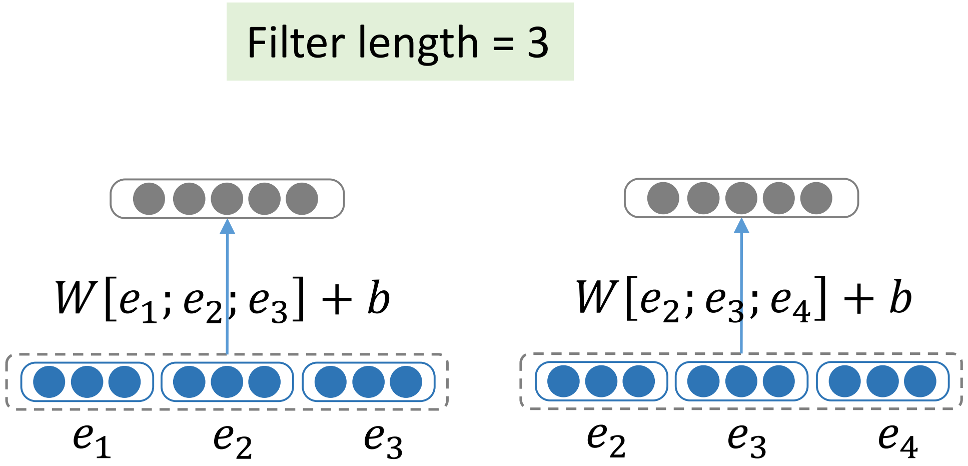


Duyu Tang, Bing Qin, Ting Liu. 2015. **Document modeling with gated recurrent neural network for sentiment classification**. *In EMNLP 2015*.

Sentence Modeling

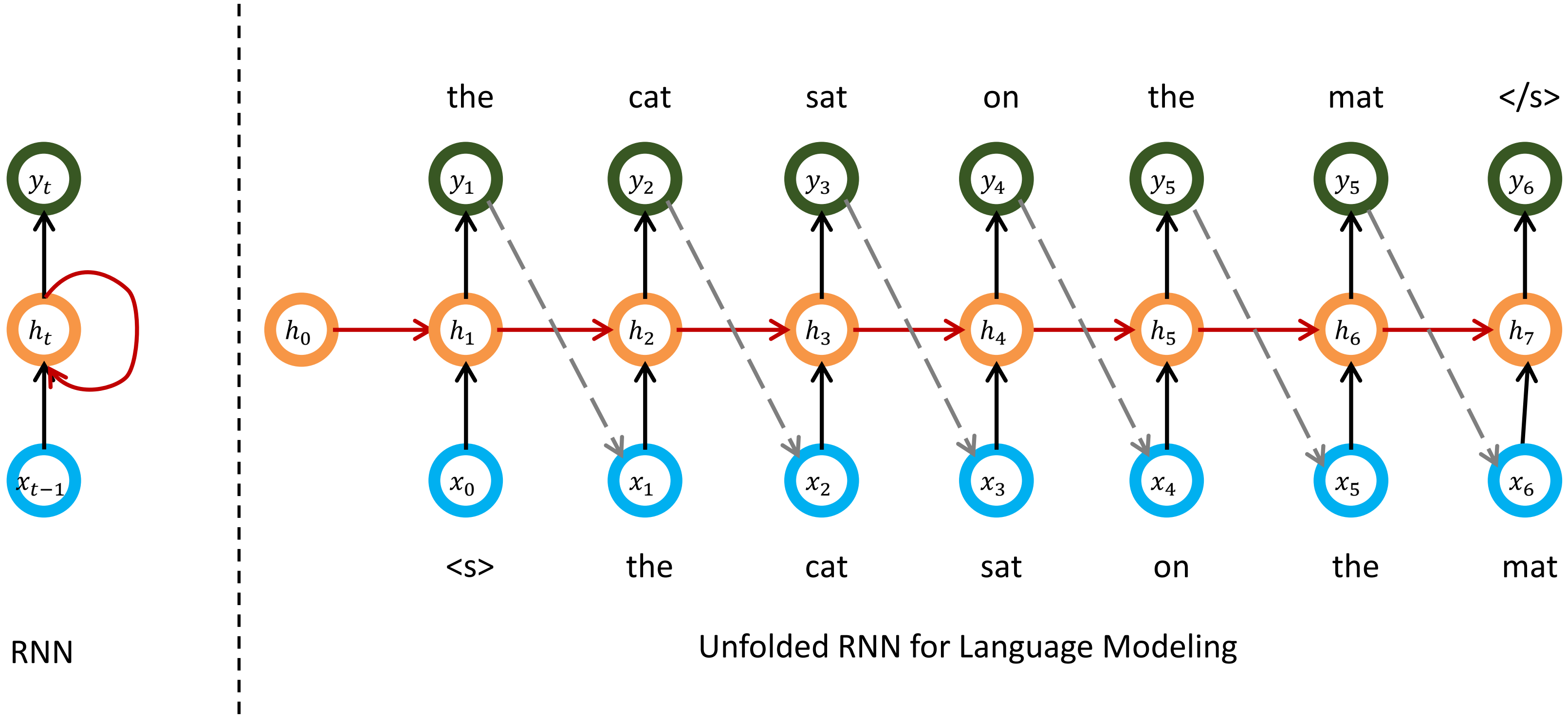
- CNN with multiple filters
  - Use unigram, bigram, trigram information





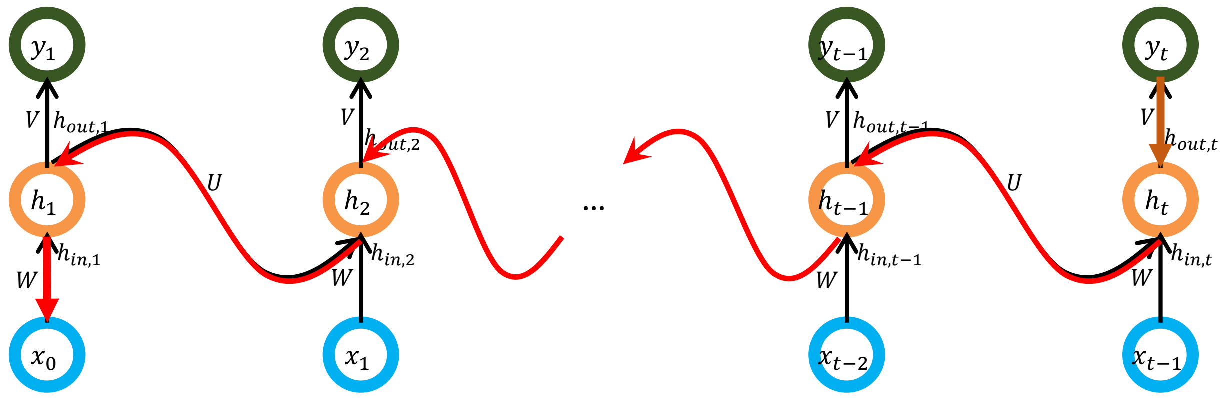
Recurrent Neural Network

Unfolded RNN for Language Modeling



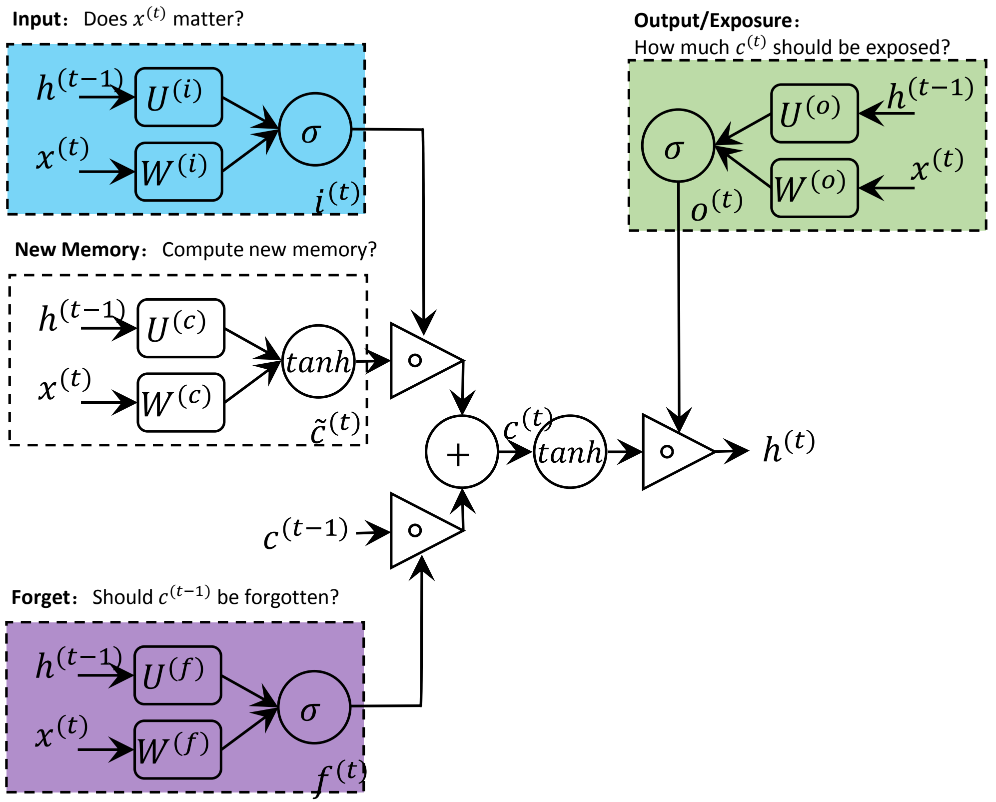
Unfolded RNN for Language Modeling

Vanishing Gradient Problem



$$\delta_{in,1} = \delta_{out,t} \times \frac{\partial h_{out,t}}{\partial h_{in,t}} \times \frac{\partial h_{in,t}}{\partial h_{out,t-1}} \times \cdots \times \frac{\partial h_{in,2}}{\partial h_{out,1}} \times \frac{\partial h_{out,1}}{\partial h_{in,1}}$$

LSTM: Long Short Term Memory



$$i^{(t)} = \sigma(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)})$$

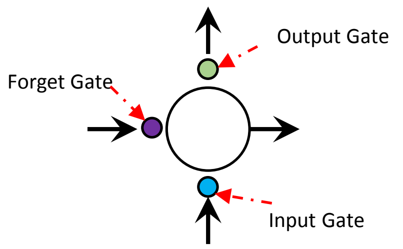
$$f^{(t)} = \sigma(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)})$$

$$o^{(t)} = \sigma(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)})$$

$$\tilde{c}^{(t)} = tanh(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)})$$

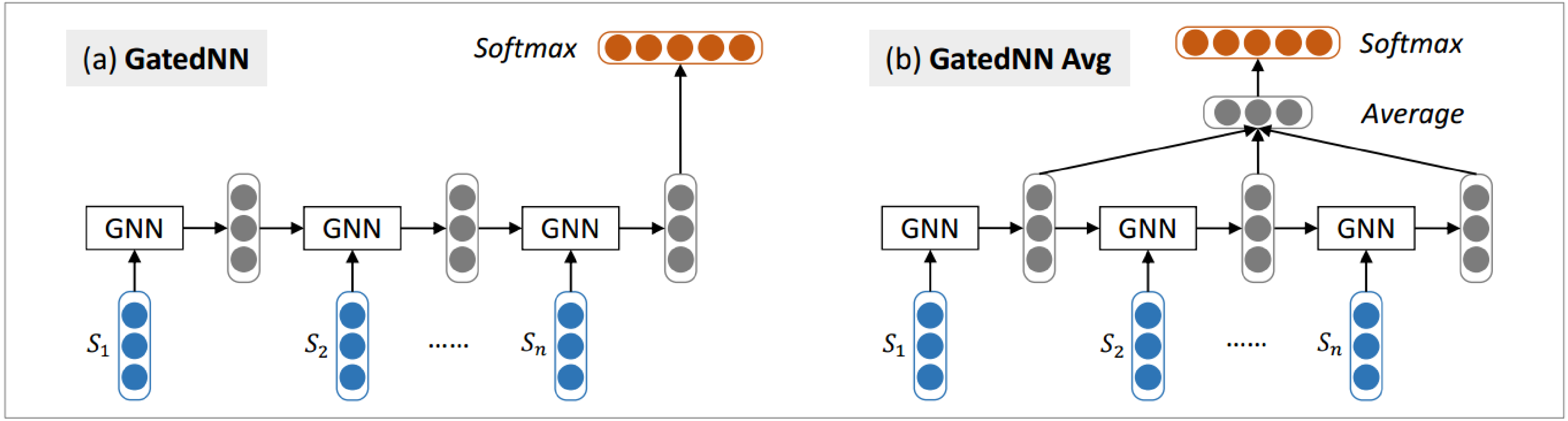
$$c^{(t)} = f^{(t)}\circ\tilde{c}^{(t-1)} + i^{(t)}\circ\tilde{c}^{(t)}$$

$$h^{(t)} = o^{(t)}\circ tanh(c^{(t)})$$



Document Modeling with RNNLSTM

- Two options
  - Use the last hidden vector as the document representation
  - Use all the hidden vectors (average them to get the doc vec)



Model Training

- Objective function
  - Minimize the cross-entropy error

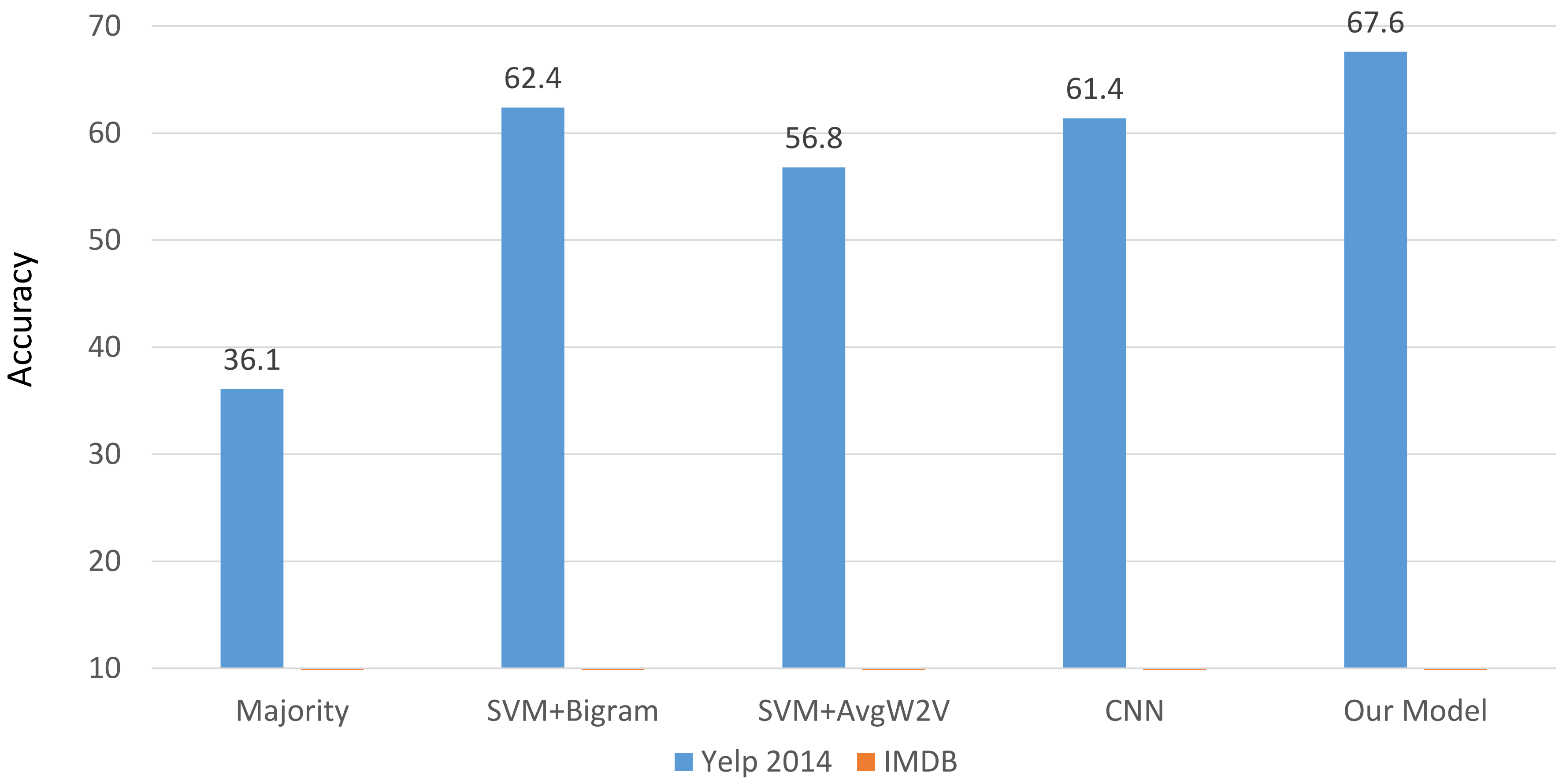
$$loss = -\sum_{d \in T} \sum_{i=1}^{C} P_i^g(d) \cdot log(P_i(d))$$

- Dataset
  - Get massive reviews from Yelp and IMBD, regarding user generated rating star as the sentiment label.
    - Train/Dev/Test = 8:1:1
    - Multi-class classification

Dataset	#documents	#sentences/ document	#words/ document	#vocabulary	#Class	Class Distribution
Yelp	1,569,264	8.97	151.9	612,636	5	.10/.09/.14/.30/.37
IMDB	348,415	14.02	325.6	115,831	10	.07/.04/.05/.05/.08/.11/ .15/.17/.12/.18

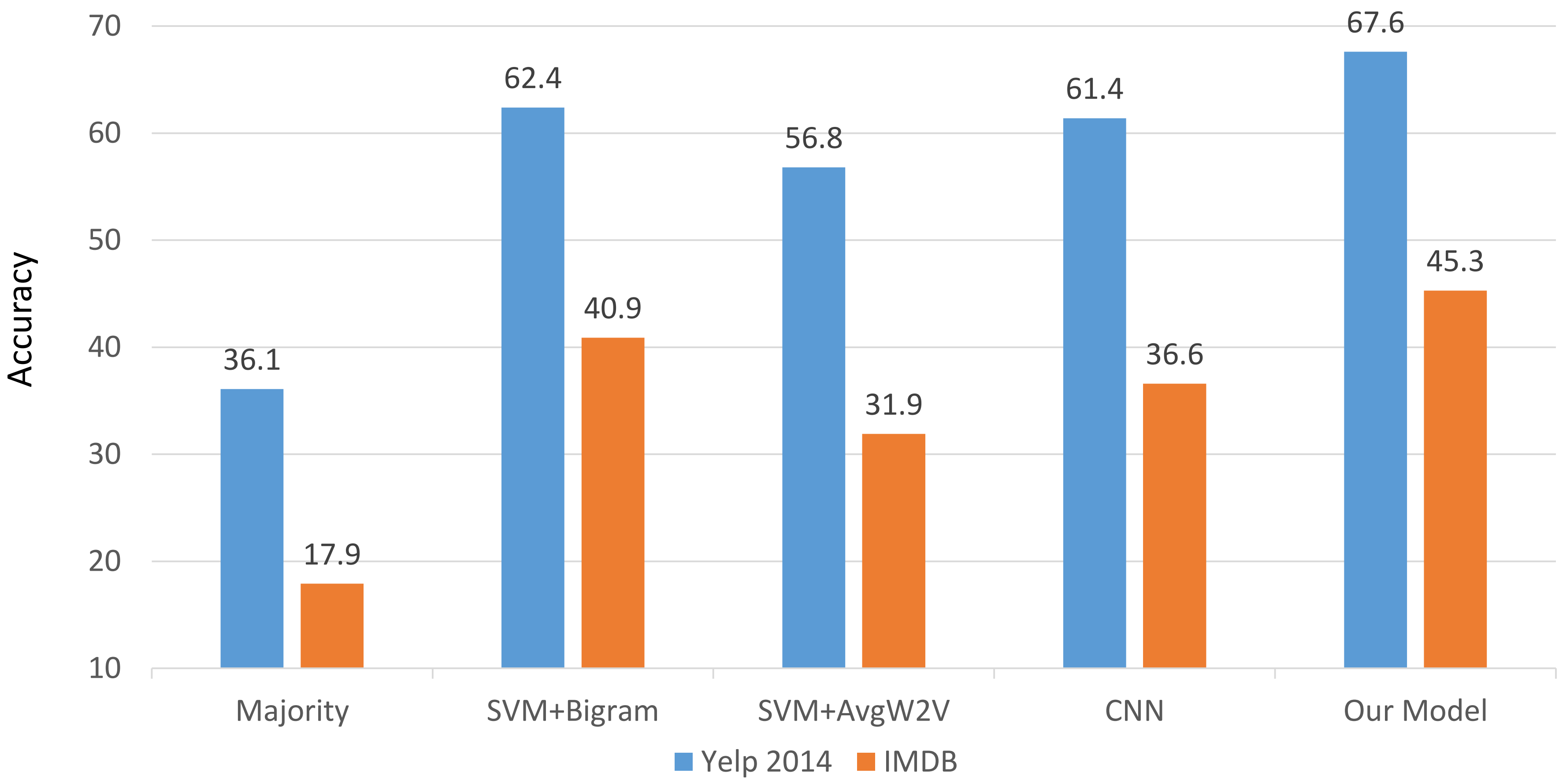
Experimental Results

- Compare with different classification algorithms



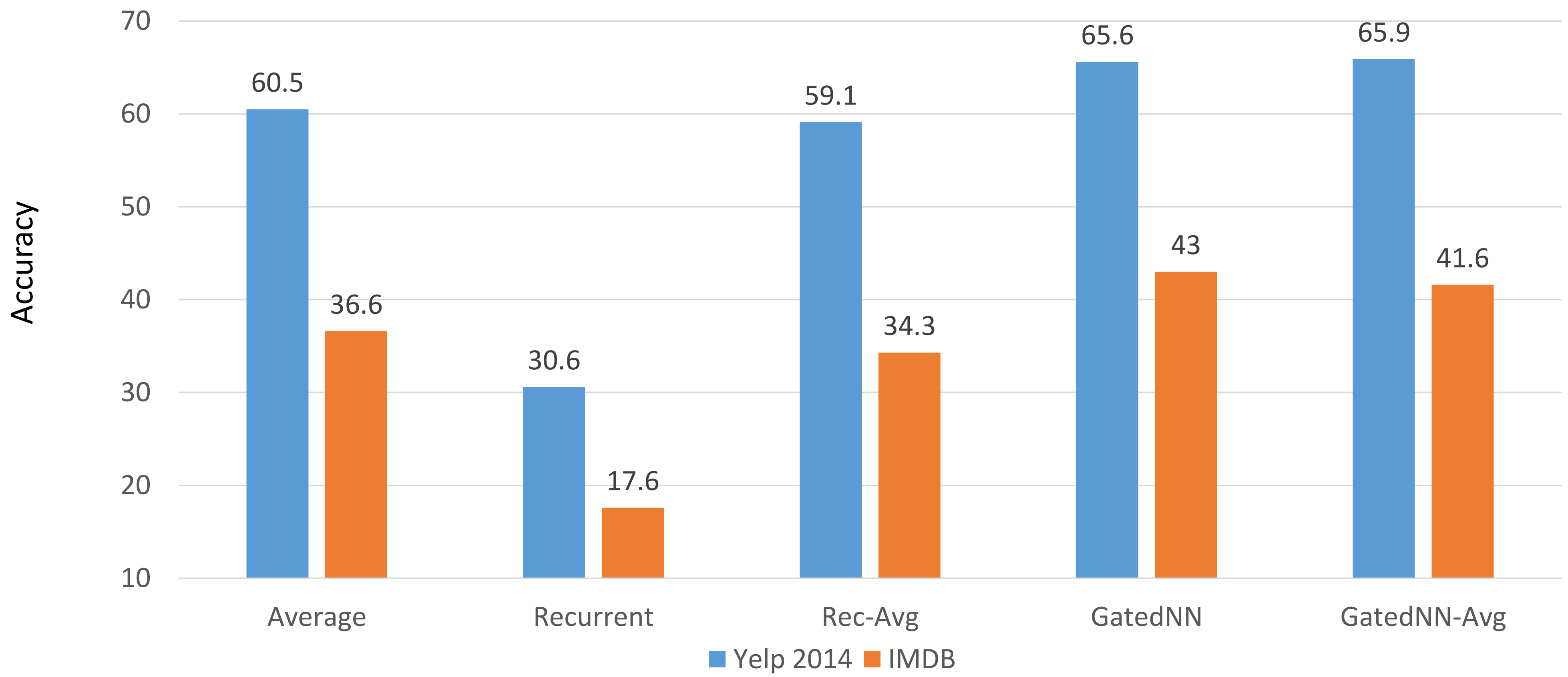
Experimental Results

- Compare with different classification algorithms



Experimental Results

- Compare with different compositional models

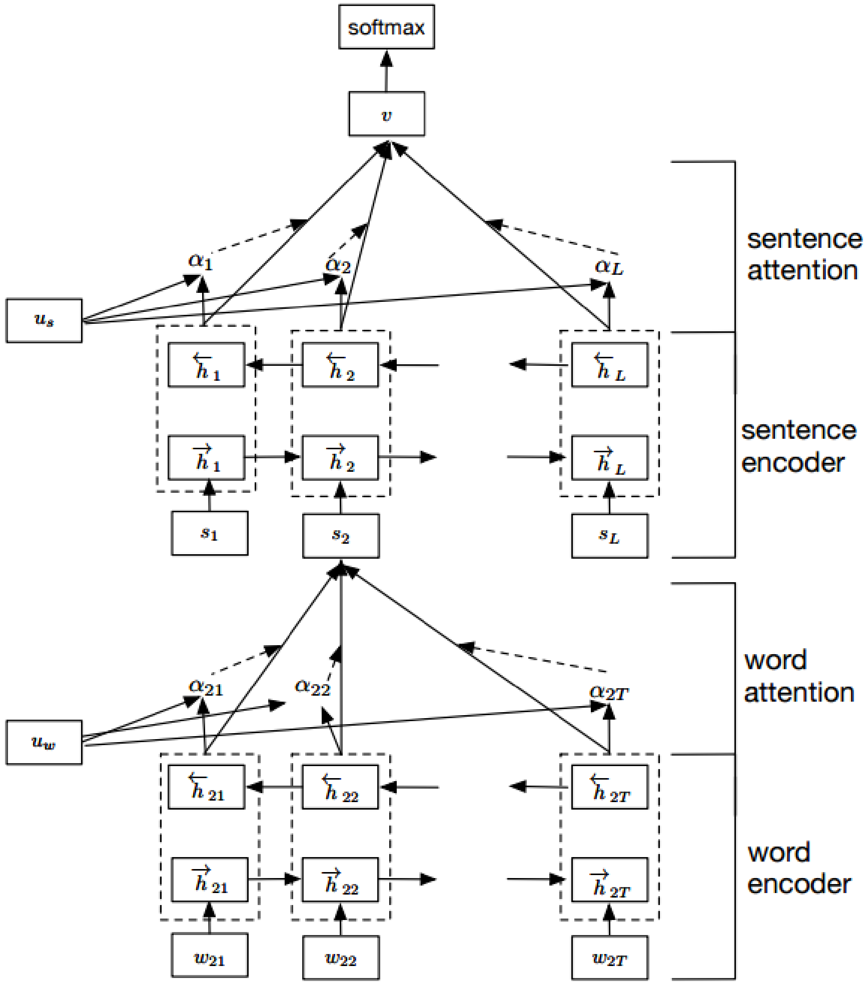


Hierarchical Attention Networks

- Four components
  - A word sequence encoder
  - A word-level attention layer
  - A sentence encoder
  - A sentence-level attention layer

Tang et al., 2015	Paragraph Vector	57.7	59.2	60.5	34.1
	CNN-word	59.7	61.0	61.5	37.6
	Conv-GRNN	63.7	65.5	66.0	42.5
	LSTM-GRNN	65.1	67.1	67.6	45.3
This paper	HN-AVE	67.0	69.3	69.9	47.8
	HN-MAX	66.9	69.3	70.1	48.2
	HN-ATT	**68.2**	**70.5**	**71.0**	**49.4**

HN stands for Hierarchical Network, **AVE** indicates averaging, **MAX** indicates max-pooling, and **ATT** indicates hierarchical attention model.



Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy. 2016. **Hierarchical Attention Networks for Document Classification**. *In NAACL 2016*.

FastText

- FastText
  - The word representations are averaged into a text representation, which is in turn fed to a linear classifier.
  - Does not use pre-trained word embeddings

Model	Yelp'13	Yelp'14	Yelp'15	IMDB
SVM+TF	59.8	61.8	62.4	40.5
CNN	59.7	61.0	61.5	37.5
Conv-GRNN	63.7	65.5	66.0	42.5
LSTM-GRNN	65.1	67.1	67.6	45.3
`fastText`	64.2	66.2	66.6	45.2

Table 3: Comparision with Tang et al. (2015). The hyper-parameters are chosen on the validation set. We report the test accuracy.

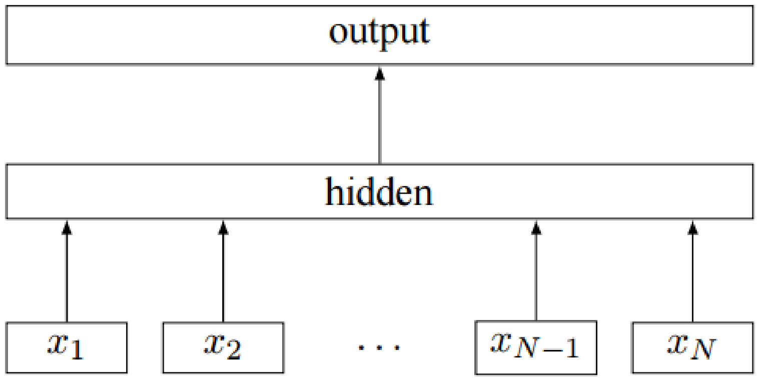


Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

fastText takes less than a minute to train on these datasets. The GRNNs method of Tang et al. (2015) takes around 12 hours per epoch on CPU with a single thread.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov. 2016. **Bag of Tricks for Efficient Text Classification**. *In arxiv.org 1607.01759*.

Directly learning embedding of text regions

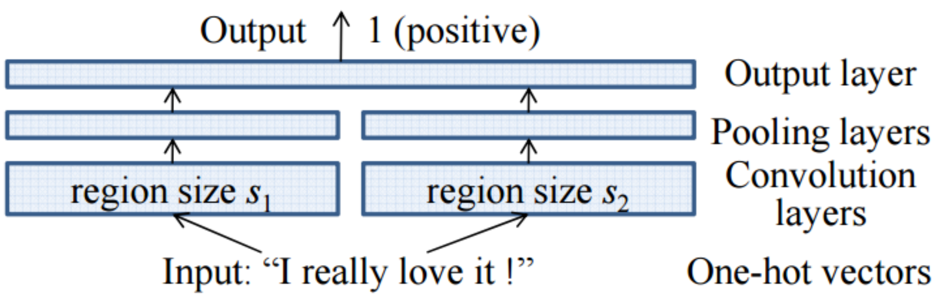
- Apply CNN to high-dimensional (**one-hot**) text data

seq-CNN

$$\mathbf{r}_0(\mathbf{x}) = \left[\begin{array}{c} 0 \\ 0 \\ \mathbf{1} \\ 0 \\ 0 \\ \text{—} \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \end{array}\right] \begin{array}{l} \text{don't} \\ \text{hate} \\ \textbf{I} \\ \text{it} \\ \text{love} \\ \\ \text{don't} \\ \text{hate} \\ \text{I} \\ \text{it} \\ \textbf{love} \end{array} \quad \mathbf{r}_1(\mathbf{x}) = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \text{—} \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \\ 0 \end{array}\right] \begin{array}{l} \text{don't} \\ \text{hate} \\ \text{I} \\ \text{it} \\ \textbf{love} \\ \\ \text{don't} \\ \text{hate} \\ \text{I} \\ \textbf{it} \\ \text{love} \end{array}$$

bow-CNN

$$\mathbf{r}_0(\mathbf{x}) = \left[\begin{array}{c} 0 \\ 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \end{array}\right] \begin{array}{l} \text{don't} \\ \text{hate} \\ \textbf{I} \\ \text{it} \\ \textbf{love} \end{array} \quad \mathbf{r}_1(\mathbf{x}) = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{array}\right] \begin{array}{l} \text{don't} \\ \text{hate} \\ \text{I} \\ \textbf{it} \\ \textbf{love} \end{array}$$



CNN with two conv layers in parallel

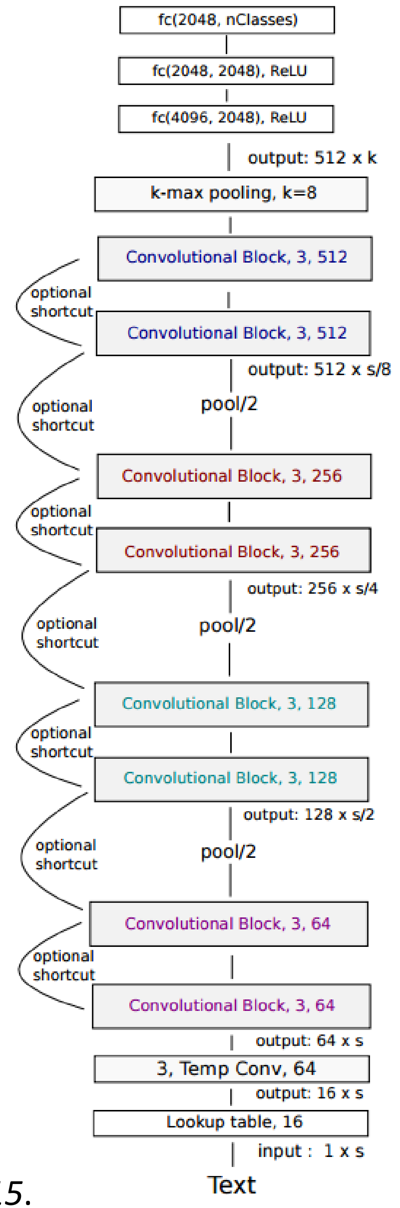
Rie Johnson and Tong Zhang. **Effective use of word order for text categorization with convolutional neural networks**. In NAACL 2015
Rie Johnson, and Tong Zhang. **Semi-supervised convolutional neural networks for text categorization via region embedding**. In *NIPS 2015*.
Rie Johnson, and Tong Zhang. **Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings**. In *ICML 2016*

Character Level CNN

- Represent text from character
  - with 6 layer → 29 layers convolutional NNs

- The alphabet consists of 70 characters,
  - including 26 english letters,
  - 10 digits,
  - 33 other characters and the new line character.

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'''/\|_@#$%^&*~`+-=<>()[]{}
```

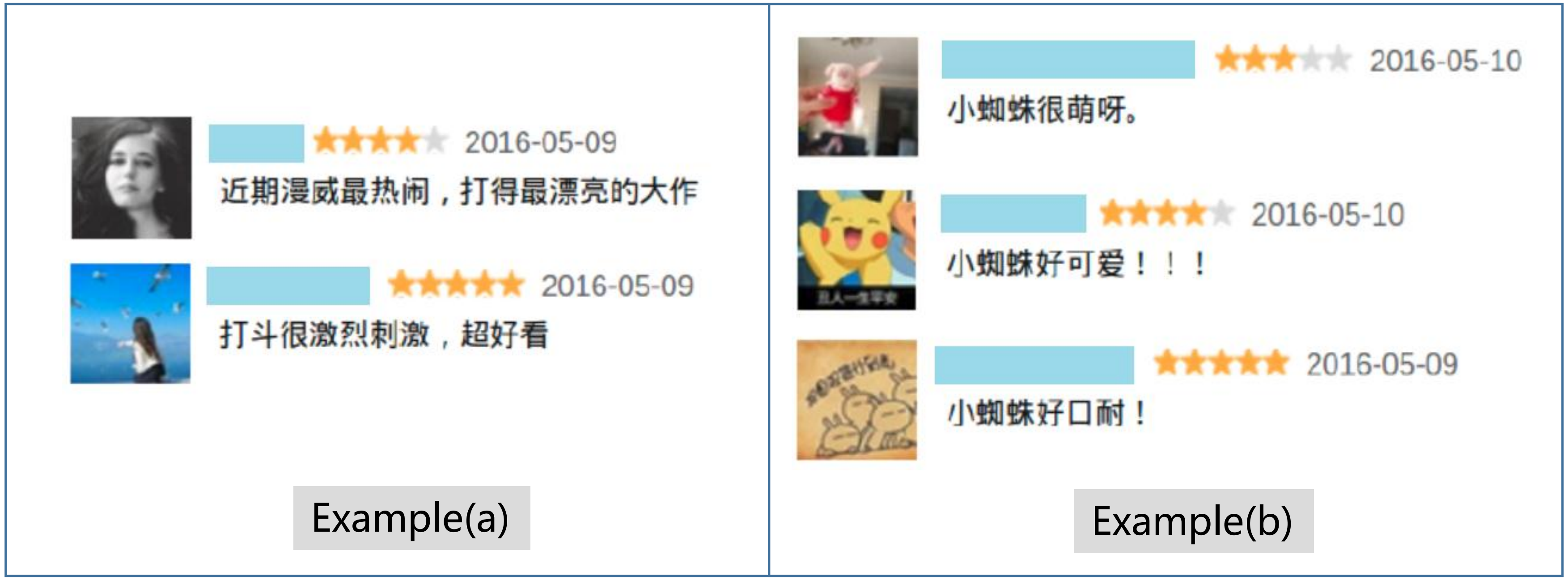


Xiang Zhang, Junbo Zhao, and Yann LeCun. **Character-level convolutional networks for text classification**. In *NIPS 2015*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. **Very Deep Convolutional Networks for Natural Language Processing**. *arXiv.org 1606.01781*.

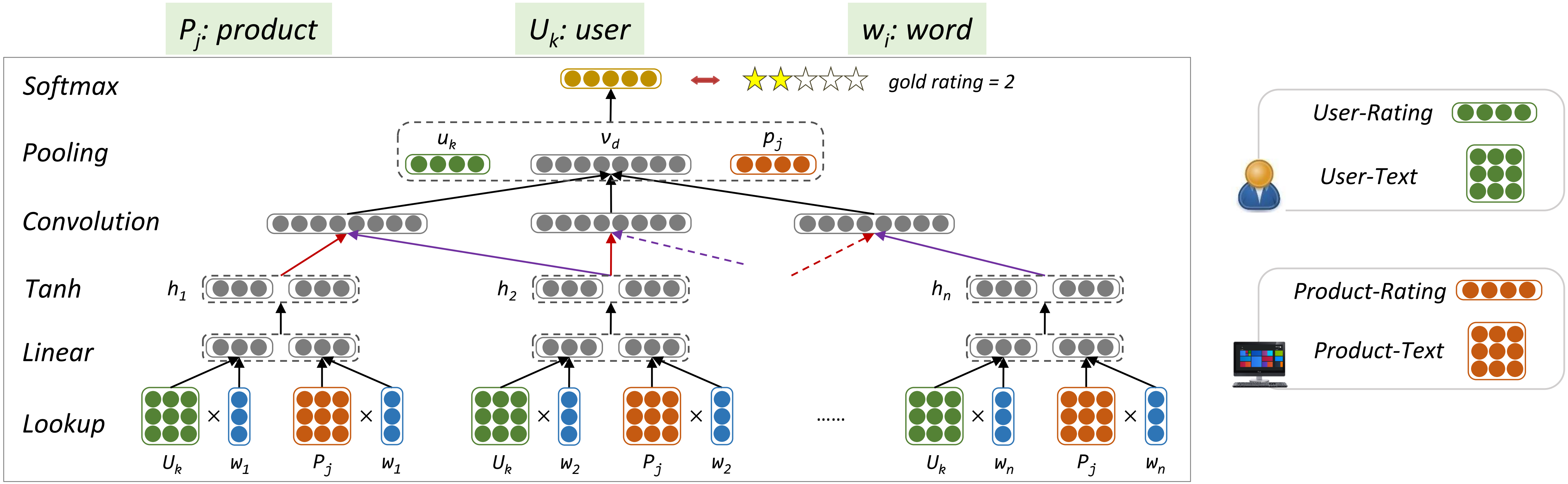
Take User Bias into Consideration

- From a sentiment analysis perspective , users have different habits to
  - Assign sentiment ratings on IMDB, Yelp ...
  - Use different sentiment words to express one's feeling



User and Product Enhanced Neural Model for Sentiment Analysis

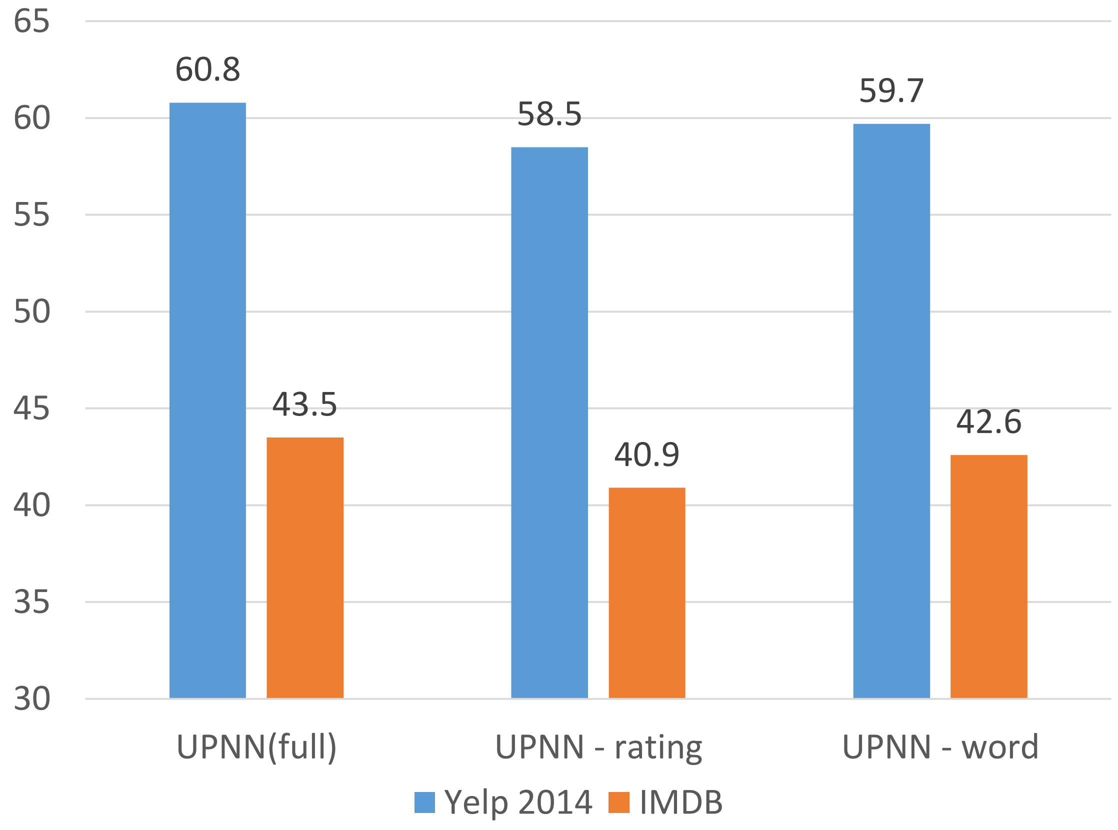
- Take into account of the evidences from text, user and product to infer the sentiment label (numeric rating).

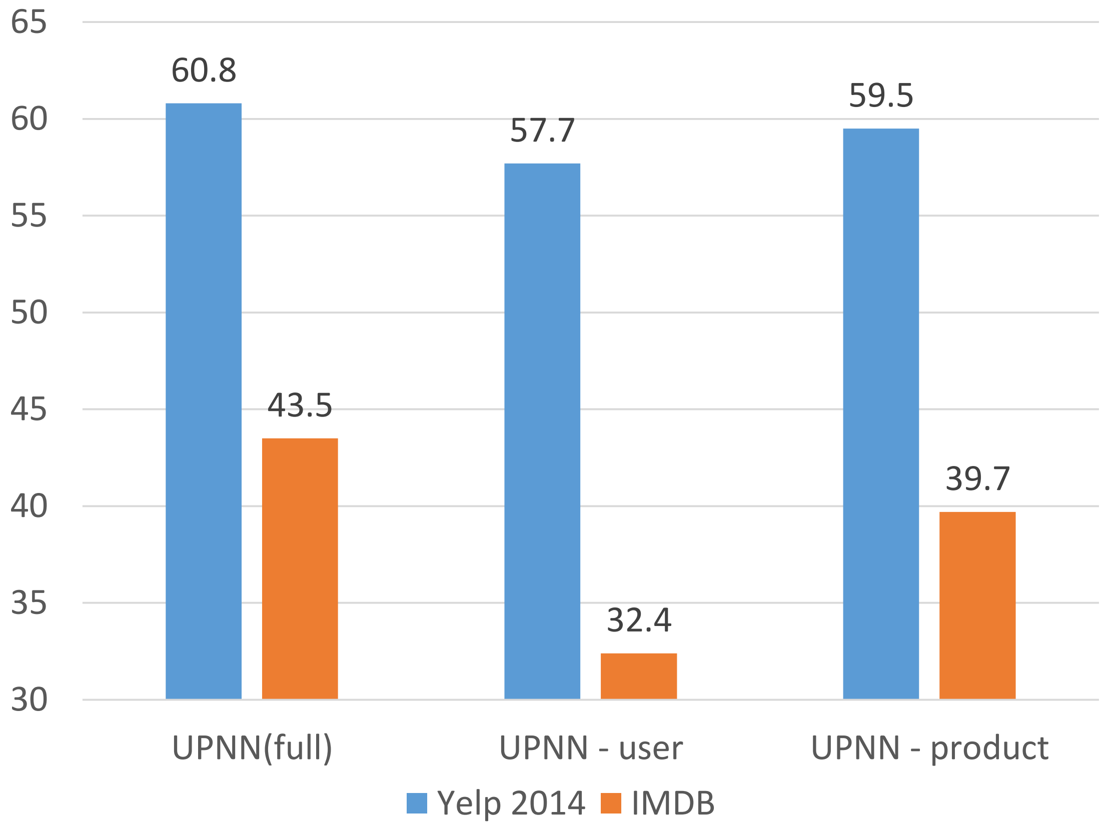


Duyu Tang, Bing Qin, Ting Liu. **Learning semantic representations of users and products for document level sentiment classification**. *In ACL 2015*.

Experimental Results

- The effects of different preferences



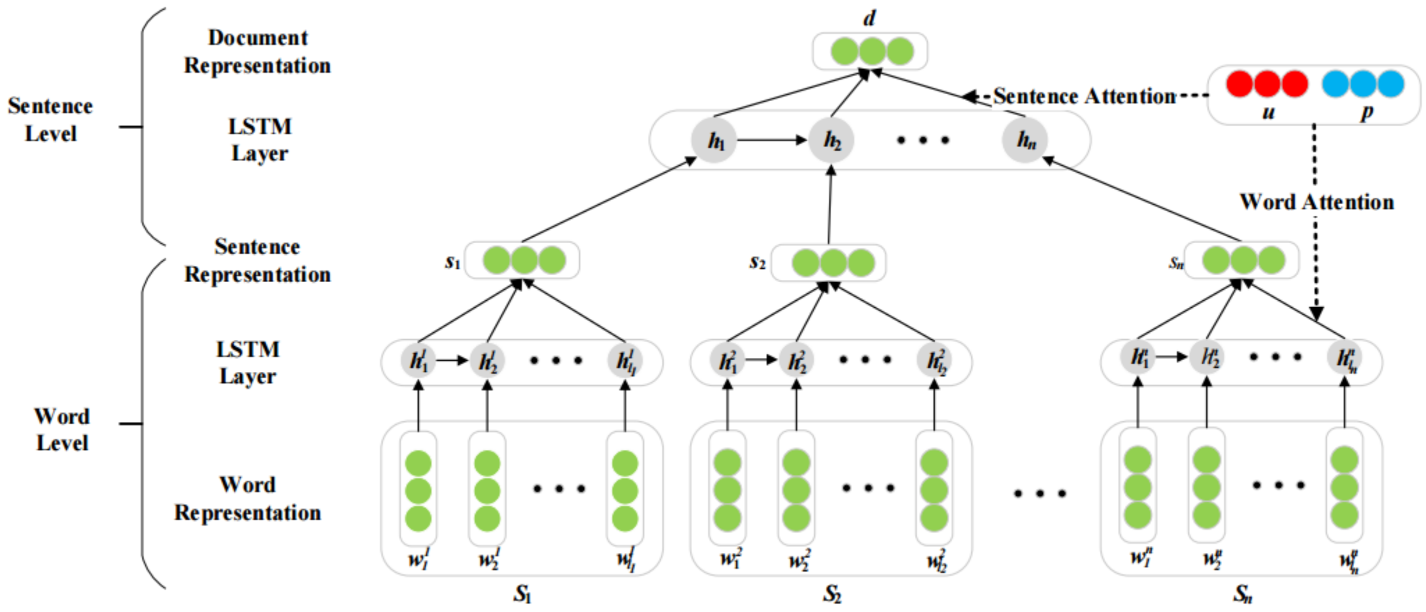


User Product Attention

- Calculate sentence/doc vec with UP attention

$$\mathbf{s}_i = \sum_{j=1}^{l_i} \alpha_j^i \mathbf{h}_j^i$$

$$\alpha_j^i = \frac{\exp(e(\mathbf{h}_j^i, \mathbf{u}, \mathbf{p}))}{\sum_{k=1}^{l_i} \exp(e(\mathbf{h}_k^i, \mathbf{u}, \mathbf{p}))}$$



$$e(\mathbf{h}_j^i, \mathbf{u}, \mathbf{p}) = \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W}_H \mathbf{h}_{ij} + \mathbf{W}_U \mathbf{u} + \mathbf{W}_P \mathbf{p} + \mathbf{b})$$

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, Zhiyuan Liu. **Neural Sentiment Classification with User and Product Attention**. *In EMNLP 2016*.

Emotion Cause Extraction

- It is a new task for sentiment analysis
- Objective: Given an emotional document
  - Identify the cause of emotion.
- Tasks:
  - Clause level classification
  - Phrase level extraction
- Data:
  - http://hlt.hitsz.edu.cn/?page_id=694

- Example:

在劝说过程中

消防官兵了解到

clause

该女子是由于对方拖欠工程款

phrase

cause

家中又急需用钱

emotion

无奈才选择跳楼轻生

Lin Gui, Dongyin Wu, Ruifeng Xu*, Qin Lu, Yu Zhou. Event-Driven Emotion Cause Extraction with Corpus Construction. In EMNLP 2016.

Emotion Cause Extraction

- Gui et al. proposed an event-driven method:
  - Use linguistic rules to extract events
  - Use multi-kernel SVMs to identify the cause

- Discussion
  - No deep learning approach on this task.
  - Performance of existing method is limited (0.67 F-measure)

Lin Gui, Dongyin Wu, Ruifeng Xu*, Qin Lu, Yu Zhou. Event-Driven Emotion Cause Extraction with Corpus Construction. In EMNLP 2016.