Sentiment Classification with Supervised Sequence Embedding

Dmitriy Bespalov[*+],
Yanjun Qi[*],
Bing Bai[*], Ali
Shokoufandeh[+]

Kindly Presented by :
Evangelos Papalexakis
from CMU

+ Drexel UNIVERSITY

* NEC

OVERVIEW

- Introduction
- Method
- Experimental results

Overview

- Introduction
- Method
- Experimental results

TASK: SENTIMENT CLASSIFICATION (SC)

- We focus on document-level sentiment classification (D-SC)
- Tackle SC as a supervised text classification task
- Two variants of D-SC:
  - **Binary sentiment classification**
    - Estimates overall sentiment of text as positive or negative
  - **Multi-class sentiment classification**
    - Determines overall sentiment of text using Likert scale
    - e.g., 5-star system for online reviews

REVIEW TEXT➔

"i believe that this book is not at all helpful since it does not explain thoroughly the material ."

→ SC →

PRIOR WORK

- Surveys [1,2] on latest developments in sentiment analysis

- Discriminative supervised methods are (close to) state-of-art
  - Linear SVM trained on Bag-of-Word (BoW) with TF-IDF representation
  - We consider BoW and BoN (Bag-of-Ngram) with TF-IDF as baselines

[1] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval 2008.*

[2] Bing Liu. Sentiment Analysis and Opinion Mining. *Lectures on HLT 2012.* Morgan & Claypool Publishers.

Baseline: Bag-of-Words Representation for Text

“Think and wonder, wonder and think.”

and	2
think	2
wonder	2

- **Bag-of-Words (BoW)** model treats text as order-invariant collection of features
  - Enumerate all unique words in text corpus and place into dictionary $\mathcal{D}$
  - Let $\mathbf{x} = (w_1, \cdots, w_N)$ denote a document from corpus
  - Define canonical basis vector with single non-zero entry at position w_i:

$$\mathbf{e}_{w_i} = (0, \ldots, 1, \ldots, 0)^\top$$

  - Thus, BoW representation of document $\mathbf{x}$:

$$\tilde{\mathbf{e}}_{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{e}_{w_i} \qquad \dim(\tilde{\mathbf{e}}_{\mathbf{x}}) = \dim(\mathbf{e}_{w_i}) = |\mathcal{D}| \times 1$$

  - Optionally, assign weights (e.g., TF-IDF, BM25) to every word

WORD PHRASES (N-GRAMS) IMPORTANT FOR SC TASK

- Short phrases / n-grams better capture sentiment than single words
  - E.g. words "recommend" and "book"

"I absolutely recommend this book"

"I highly recommend this book"

"I recommend this book"

"I somewhat recommend this book"

"I don't recommend this book"

How to model n-grams / phrases in BoW Model 1

"the film is palpable evil genius"

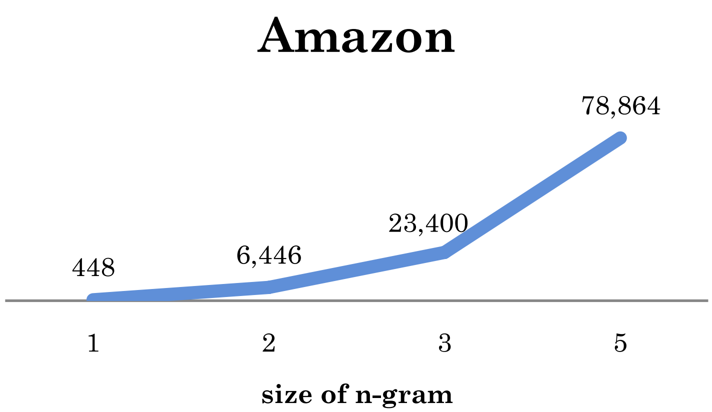
"the film"	**1**
"film is"	1
"is palpable"	1
"palpable evil"	1
"evil genius"	1

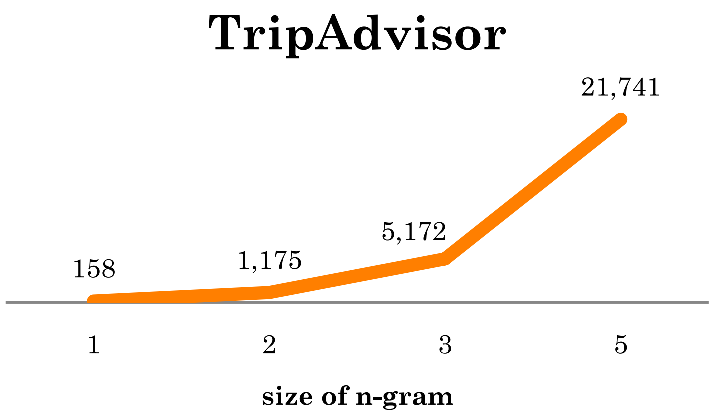
- Extend BoW to encode distributions of n-grams
  - n continuous words (i.e., n-grams) from corpus
  - Add n-grams to set Γ and use their distribution as features in BoW model:

$$\dim(\mathbf{e}_{w_i}) = |\Gamma| \times 1, \quad |\Gamma| = O(|\mathcal{D}|^n)$$

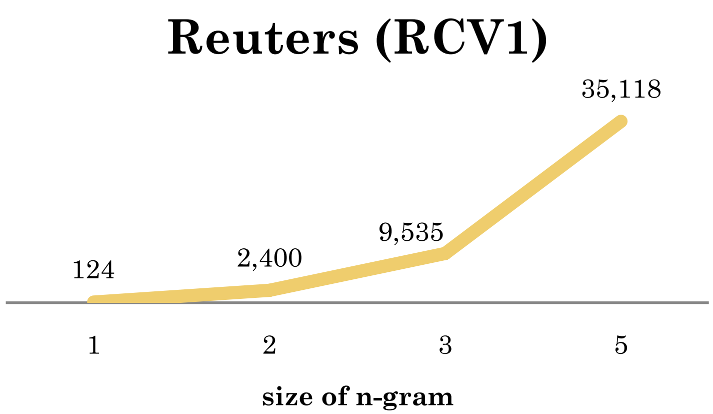
- BoW with n-grams will be referred to as bag of n-grams (BoN)

BoN: Curse of Dimensionality (Following numbers are in thousands)





Dimensionality of BoN grows exponentially with n, thus feature selection pre-processing is required



Overview

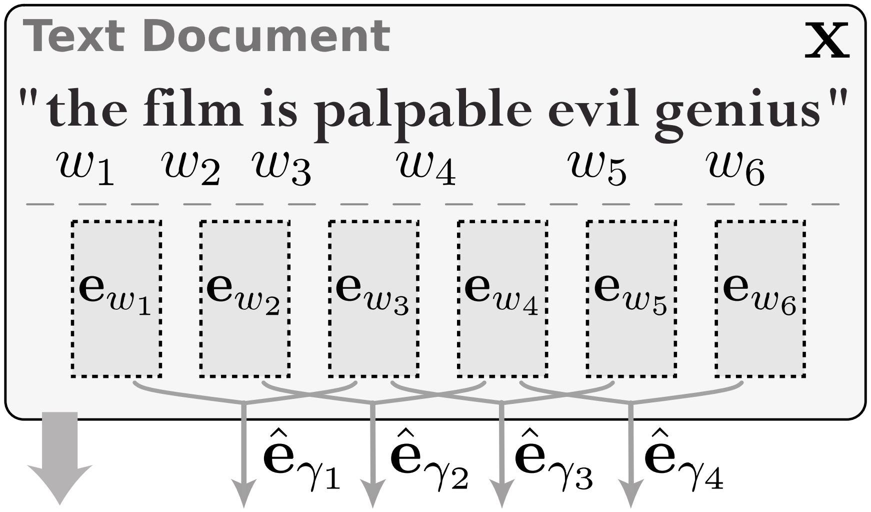
- Introduction
- Method
- Experimental results

The Proposed Method: Supervised Sequence Encoder (SSE)

- A model efficiently encodes text phrases and document
- KEY: embed all sliding n-gram windows from text into a learned latent space based on supervised signals
- Implemented as deep Neural Network (NN) architecture
- Latent projection and supervised classifier are jointly trained with back-propagation using stochastic gradient descent

$$\mathbf{e}_{w_i} = (0, \ldots, 0, \underset{\text{at index } w_i}{1}, \ldots, 0)^\top$$

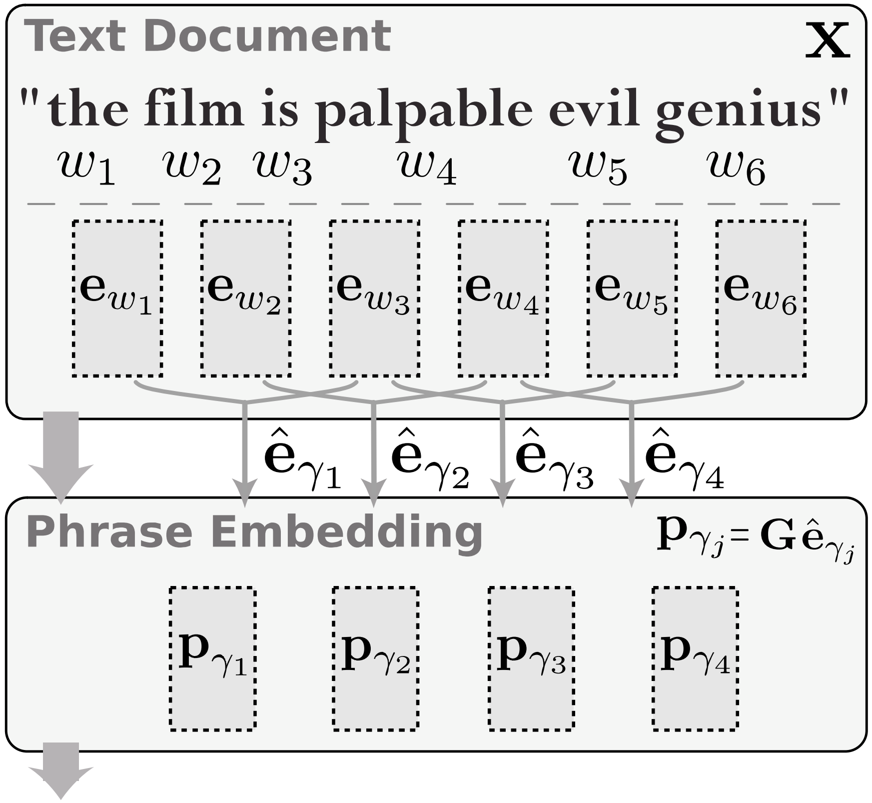
$$\hat{\mathbf{e}}_{\gamma_j} = [\mathbf{e}_{w_j}^\top, \mathbf{e}_{w_{j+1}}^\top, \ldots, \mathbf{e}_{w_{j+n-1}}^\top]^\top$$



$$\mathbf{e}_{w_i} = (0, \ldots, 0, \underset{\text{at index } w_i}{1}, \ldots, 0)^\top$$

$$\hat{\mathbf{e}}_{\gamma_j} = [\mathbf{e}_{w_j}^\top, \mathbf{e}_{w_{j+1}}^\top, \ldots, \mathbf{e}_{w_{j+n-1}}^\top]^\top$$

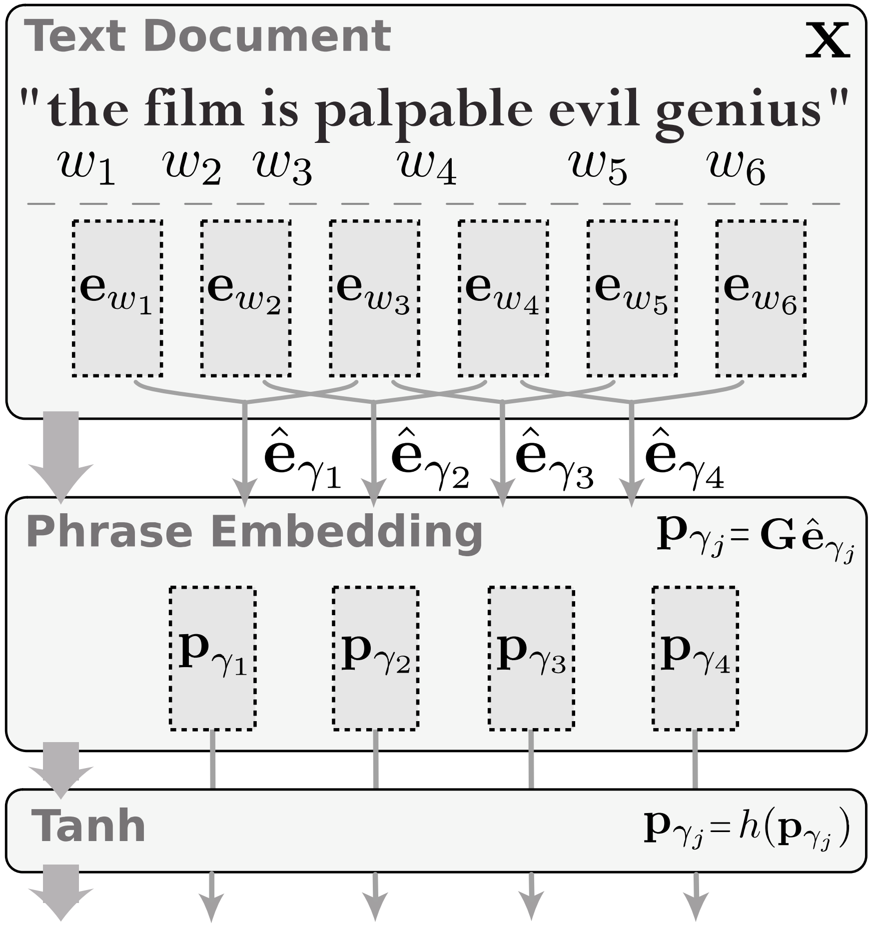
$$\mathbf{p}_{\gamma_j} = \mathbf{G} \times \hat{\mathbf{e}}_{\gamma_j}$$



$$\mathbf{e}_{w_i} = (0, \ldots, 0, \underset{\text{at index } w_i}{1}, \ldots, 0)^\top$$

$$\hat{\mathbf{e}}_{\gamma_j} = [\mathbf{e}_{w_j}^\top, \mathbf{e}_{w_{j+1}}^\top, \ldots, \mathbf{e}_{w_{j+n-1}}^\top]^\top$$

$$\mathbf{p}_{\gamma_j} = \mathbf{G} \times \hat{\mathbf{e}}_{\gamma_j}$$

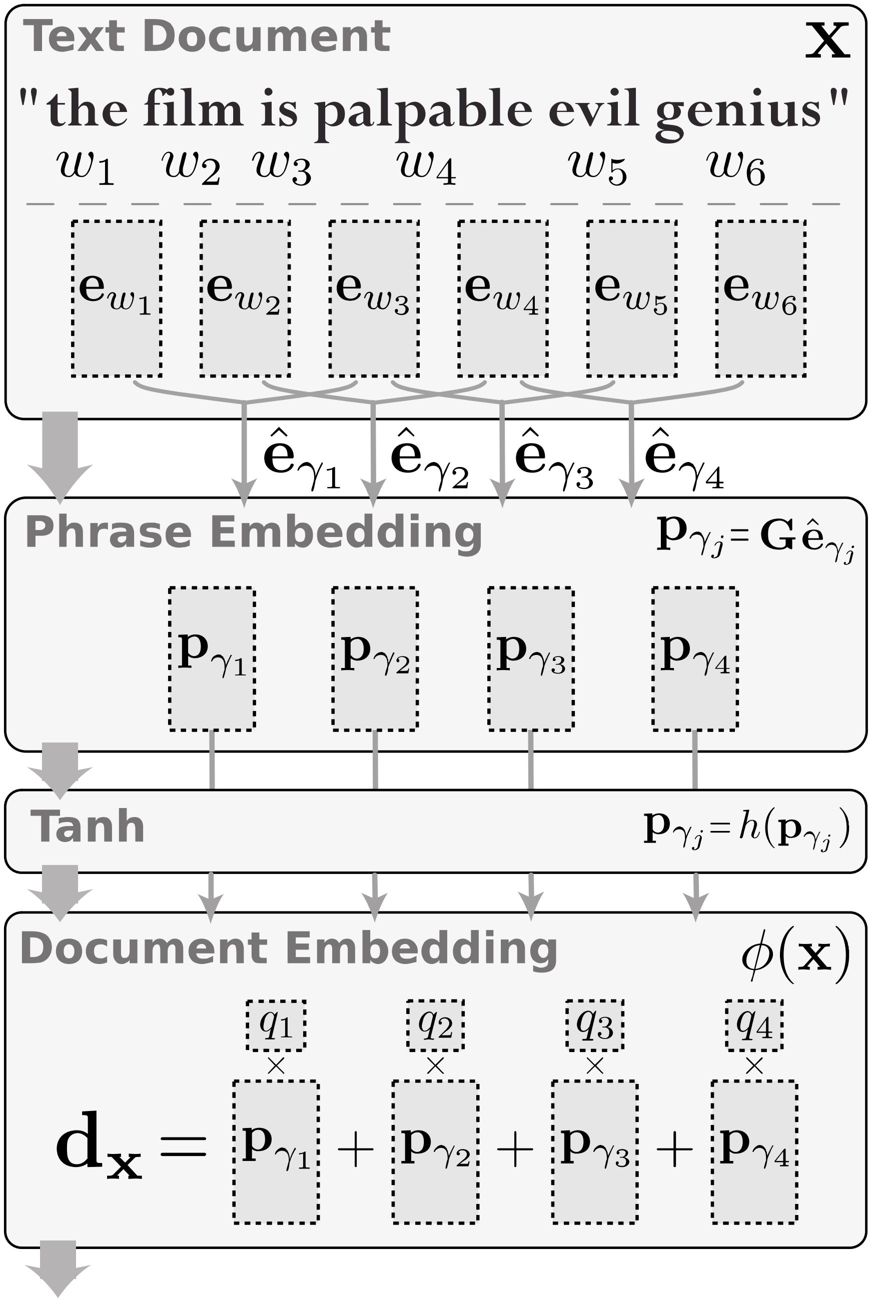


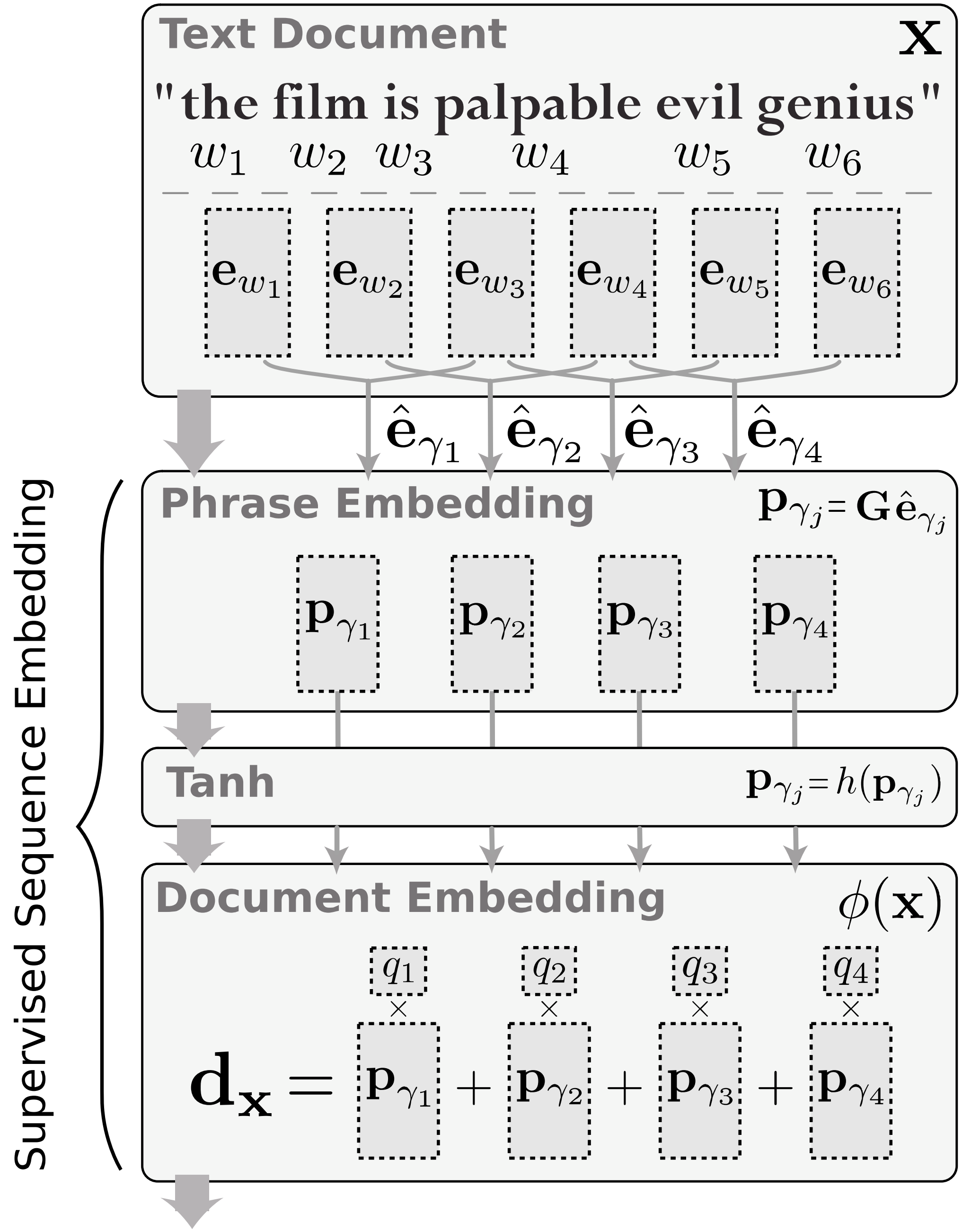
$$\mathbf{e}_{w_i} = (0, \ldots, 0, \underset{\text{at index } w_i}{1}, \ldots, 0)^\top$$

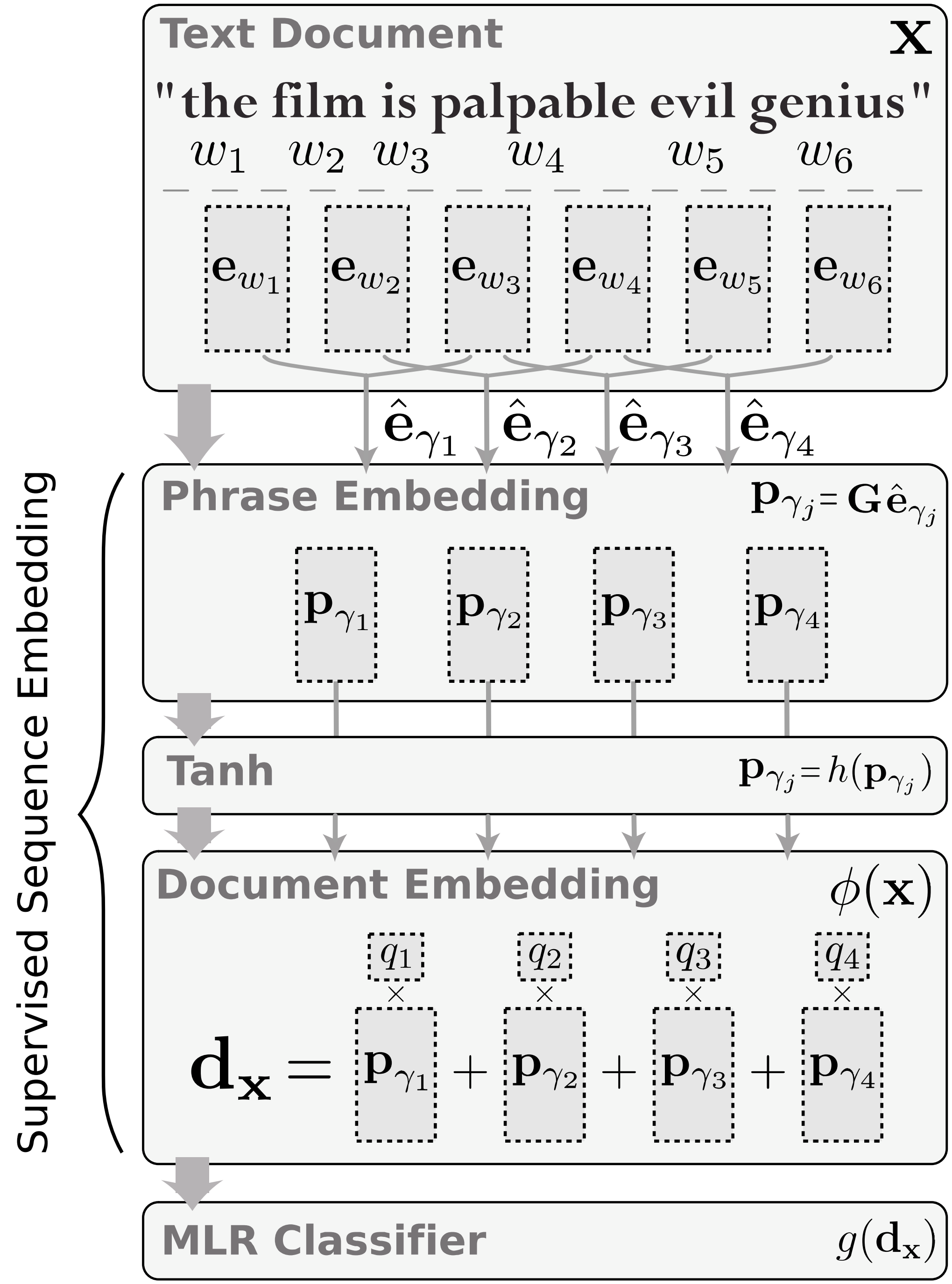
$$\hat{\mathbf{e}}_{\gamma_j} = [\mathbf{e}_{w_j}^\top, \mathbf{e}_{w_{j+1}}^\top, \ldots, \mathbf{e}_{w_{j+n-1}}^\top]^\top$$

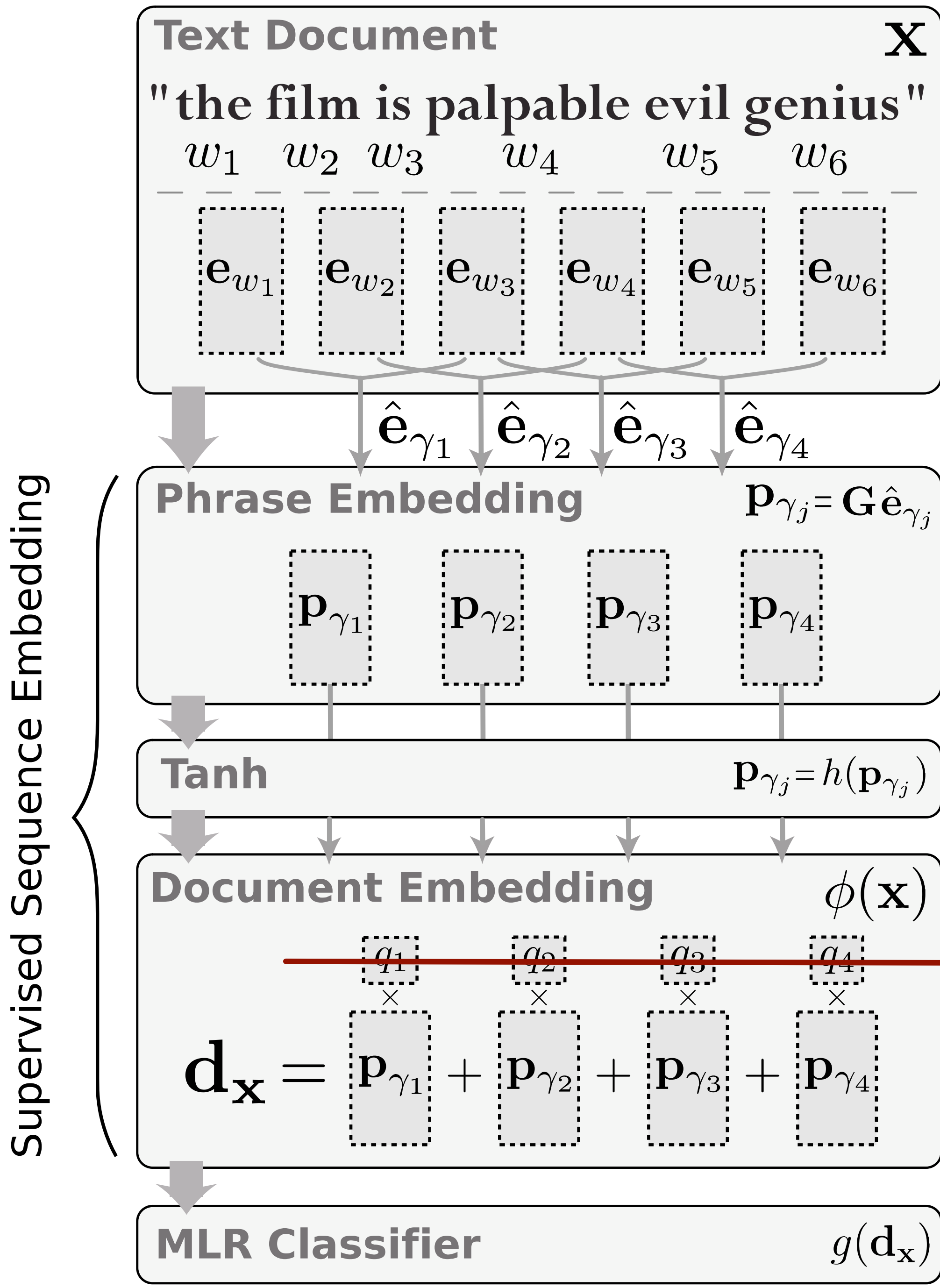
$$\mathbf{p}_{\gamma_j} = \mathbf{G} \times \hat{\mathbf{e}}_{\gamma_j}$$

$$\phi(\mathbf{x}) \equiv \mathbf{d}_\mathbf{x} = \sum_{j=1}^{N} q_j \times h(\mathbf{p}_{\gamma_j})$$




Text Document
x
"the film is palpable evil genius"
w1 w2 w3 w4 w5 w6
e_w1 e_w2 e_w3 e_w4 e_w5 e_w6
ê_γ1 ê_γ2 ê_γ3 ê_γ4
Phrase Embedding
p_γj = G ê_γj
p_γ1 p_γ2 p_γ3 p_γ4
Tanh
p_γj = h(p_γj)
Document Embedding
φ(x)
q1 q2 q3 q4
d_x = p_γ1 + p_γ2 + p_γ3 + p_γ4
Supervised Sequence Embedding


Text Document
x
"the film is palpable evil genius"
w_1 w_2 w_3 w_4 w_5 w_6
$\mathbf{e}_{w_1}$ $\mathbf{e}_{w_2}$ $\mathbf{e}_{w_3}$ $\mathbf{e}_{w_4}$ $\mathbf{e}_{w_5}$ $\mathbf{e}_{w_6}$
$\hat{\mathbf{e}}_{\gamma_1}$ $\hat{\mathbf{e}}_{\gamma_2}$ $\hat{\mathbf{e}}_{\gamma_3}$ $\hat{\mathbf{e}}_{\gamma_4}$
Phrase Embedding
$\mathbf{p}_{\gamma_j} = \mathbf{G}\hat{\mathbf{e}}_{\gamma_j}$
$\mathbf{p}_{\gamma_1}$ $\mathbf{p}_{\gamma_2}$ $\mathbf{p}_{\gamma_3}$ $\mathbf{p}_{\gamma_4}$
Tanh
$\mathbf{p}_{\gamma_j} = h(\mathbf{p}_{\gamma_j})$
Document Embedding
$\phi(\mathbf{x})$
q_1 q_2 q_3 q_4
$\mathbf{d}_{\mathbf{x}} = q_1 \times \mathbf{p}_{\gamma_1} + q_2 \times \mathbf{p}_{\gamma_2} + q_3 \times \mathbf{p}_{\gamma_3} + q_4 \times \mathbf{p}_{\gamma_4}$
MLR Classifier
$g(\mathbf{d}_{\mathbf{x}})$
Supervised Sequence Embedding



Two variants of SSE for SC task:
-I: SSE
-II: SSE-W

$$\phi(\mathbf{x}) \equiv \mathbf{d_x} = \sum_{j=1}^{N} q_j \times h(\mathbf{p}_{\gamma_j})$$

SSE: uniform weights
$q_j = \frac{1}{N} = \frac{1}{4}, \forall j \in [1, N]$

SSE-W: learn weights from n-gram locations $(\frac{j}{N})$ using mixture model

Classification with Multinomial Logistic Regression (MLR)

- Popular loss model for classification [12]
- Known to rival hinge loss (SVM-like)
- Predicts conditional probability distribution over labels given input vector **d**
- Learns coefficient weights $\boldsymbol{\beta}_i$ for every label $i \in [1, C]$
- Performs label inference:

$$g(\mathbf{d}) = \arg\max_{i \in [1,C]} \frac{\exp\left(\boldsymbol{\beta}_i^\top \mathbf{d}\right)}{1 + \sum_k \exp\left(\boldsymbol{\beta}_k^\top \mathbf{d}\right)}$$

- Called **negative log-likelihood loss** in literature due to the form of objective (loss function)

[12] Y. LeCun et al. A Tutorial on Energy-Based Learning. *Predicting Structured Data.* MIT Press 2006.

Backpropagation & Stochastic Gradient Descent

- **Backpropagation** [10] is supervised learning method for **neural network (NN)**
  - Using backward recurrence it jointly optimizes all NN parameters
  - Requires all activation functions to be differentiable
  - Enables flexible design in deep NN architecture
  - Gradient descent is used to (locally) minimize objective:

$$\mathbf{A}^{\mathbf{k+1}} = \mathbf{A}^{\mathbf{k}} - \eta \frac{\partial \mathrm{L}}{\partial \mathbf{A}^{\mathbf{k}}}$$

- **Stochastic Gradient Descent (SGD)** [11] is first-order iterative optimization
  - SGD is an **online learning** method
  - Approximates “true” gradient with a gradient at one data point
  - Attractive because of low computation requirement
  - Rivals **batch learning** (e.g., SVM) methods on large datasets

[10] Y. LeCun et al. 1998. Efficient BackProp.

[11] Olivier Bousquet and Ulrike von Luxburg. 2004. Stochastic Learning.

Advantages of SSE

- SSE utilizes only unigram features:
  - latent n-grams are defined as cumulative of unigram vectors
- Phrase structure is encoded by learning n embedding vectors for a unigram, one per every position in the n-gram
- SSE-W extension encodes positional information of each ngram in the global document structure
- Parameter space of SSE grows linear with n (i.e., size of n-gram)
- Computation of latent n-grams in SSE is extremely fast
  - requires only vector additions and multiplications with scalars
  - i.e., equivalent to n (sparse) projections of BoW representation

Overview

- Introduction
- Method
- Experimental results

Baseline I: Linear SVM [13] with BoN Representation

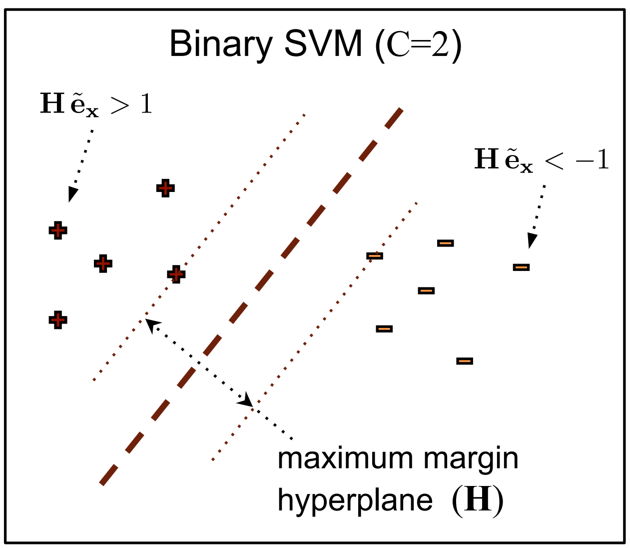
Review Text $\mathbf{X}$

↓

BoN with TF-IDF $\tilde{\mathbf{e}}_{\mathbf{x}}$

↓

Linear SVM $\mathbf{H}\,\tilde{\mathbf{e}}_{\mathbf{x}}$



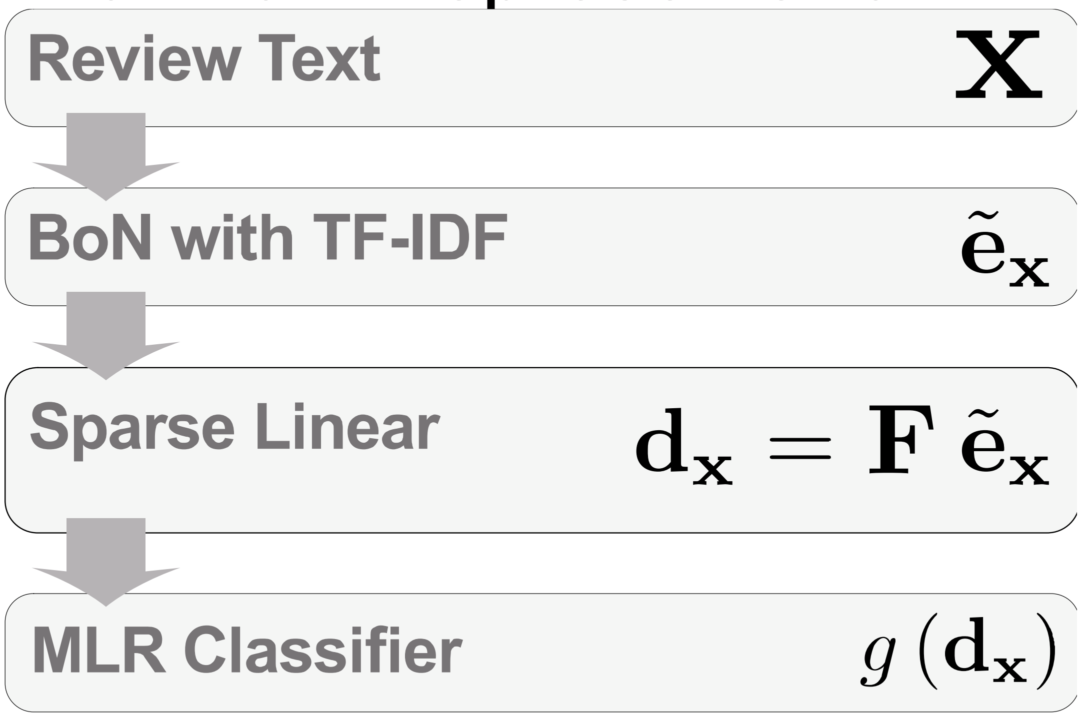
$$\dim(\tilde{\mathbf{e}}_{\mathbf{x}}) = |\Gamma_n| \times 1$$

$$\dim(\mathbf{H}) = C \times |\Gamma_n|$$

Multi-class (C>2) is reduced to
C binary (one-vs-all) SVM classifiers

[13] C. Cortes and V. Vapnik, Support-Vector Networks, *Machine Learning, 20, 1995.*

Baseline II: Perceptron (PRC) for BoN Representation



$$\dim\left(\tilde{\mathbf{e}}_{\mathbf{x}}\right) = |\Gamma_n| \times 1$$

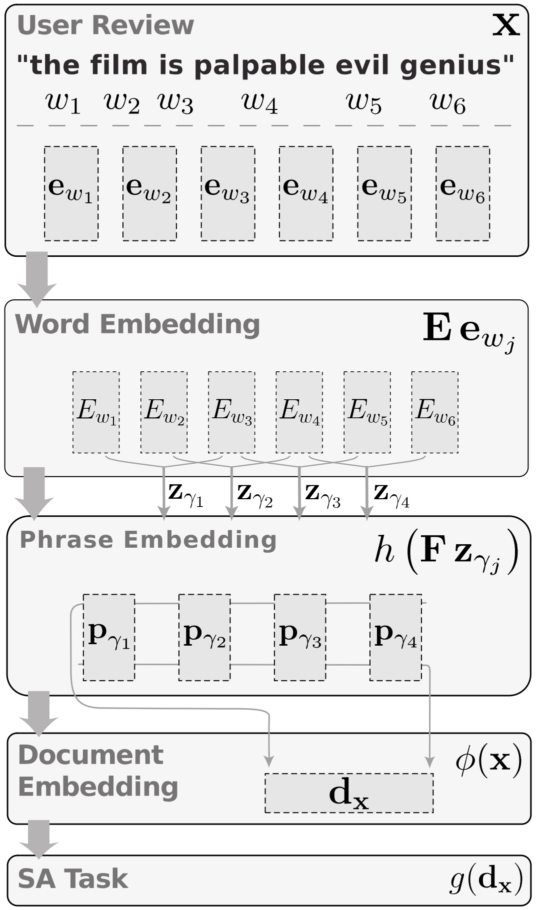
$$\dim\left(\mathbf{F}\right) = M \times |\Gamma_n$$

$$|\Gamma_n| = O\left(|\mathcal{D}|^n\right)$$

$$\dim\left(\mathbf{d}_{\mathbf{x}}\right) = M \times 1$$

Baseline III: LTC based SC

- SSE was motivated by Lookup Temporal Convolution (LTC)
  - originally proposed by Collobert and Weston [8]
  - adopted to sentiment classification in our prior work [9]
  - LTC is based on supervised word embedding



[8] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. *ICML 2008.*

[9] Dmitriy Bespalov and Bing Bai and Yanjun Qi and Ali Shokoufandeh. Sentiment Classification Based on Supervised Latent n-gram Analysis. *CIKM 2011.*

Sentiment Datasets

- Use two large-scale sentiment datasets
  - Amazon & TripAdvisor
- Amazon contains product reviews from 25 categories
  - samples 257,900 training / 110,562 testing / 10,000 validation
  - e.g., apparel, automotive, baby, DVDs, electronics, magazines
- TripAdvisor contains hotel reviews from across the globe
  - Samples 55,306 training / 10,078 samples testing / 5,000 validation
  - Consider only overall ratings for reviews
- Create balanced **70/30%** train-test splits
  - Validating set was sampled from the respective **test** sets
- For baseline BoN approaches, filtering n-grams with mutual information (MI) [14]
  - Retained top **500,000** phrases from respective training sets

[14] J. Blitzer et al. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. *ACL 2007*.

Experimental Results: Sentiment Classification

- **Micro-average error** rate is reported
- Numbers marked with † (or ‡) are statistically significantly better than **SVM BoN-3g** with $p < 0.0001$ (or $p < 0.01$)
- $2 \cdot \star$ denotes binary SC; $5 \cdot \star$ and $4 \cdot \star$ denote multi-class settings $4 \cdot \star$
  - i.e., ignores neutral reviews

Method	Amazon		TripAdvisor		
	$2 \cdot \star$	$4 \cdot \star$	$2 \cdot \star$	$4 \cdot \star$	$5 \cdot \star$
SVM BoW-1g	10.68	29.66	8.97	33.76	44.02
SVM BoW-2g	6.60	23.69	7.60	32.05	42.17
SVM BoW-3g	6.39	23.45	7.46	32.00	43.07
SVM BoW-5g	6.48	23.53	7.53	31.93	44.02
Prc BoW-3g	6.55	23.00	7.54	33.94	43.05
LTC	7.05	-	8.49	-	-
SSE	**5.69**	**22.40**	**6.90**	**33.90**	**42.21**
SSE-W	**5.63**†	**22.05**†	7.01	**31.41**	**40.76**‡

Experimental Results: Sentiment Classification (cont'd)

- **Macro-average error** rate is reported
- $5 \cdot \star$ and $4 \cdot \star$ denote multi-class settings
  - i.e., $4 \cdot \star$ ignores neutral reviews

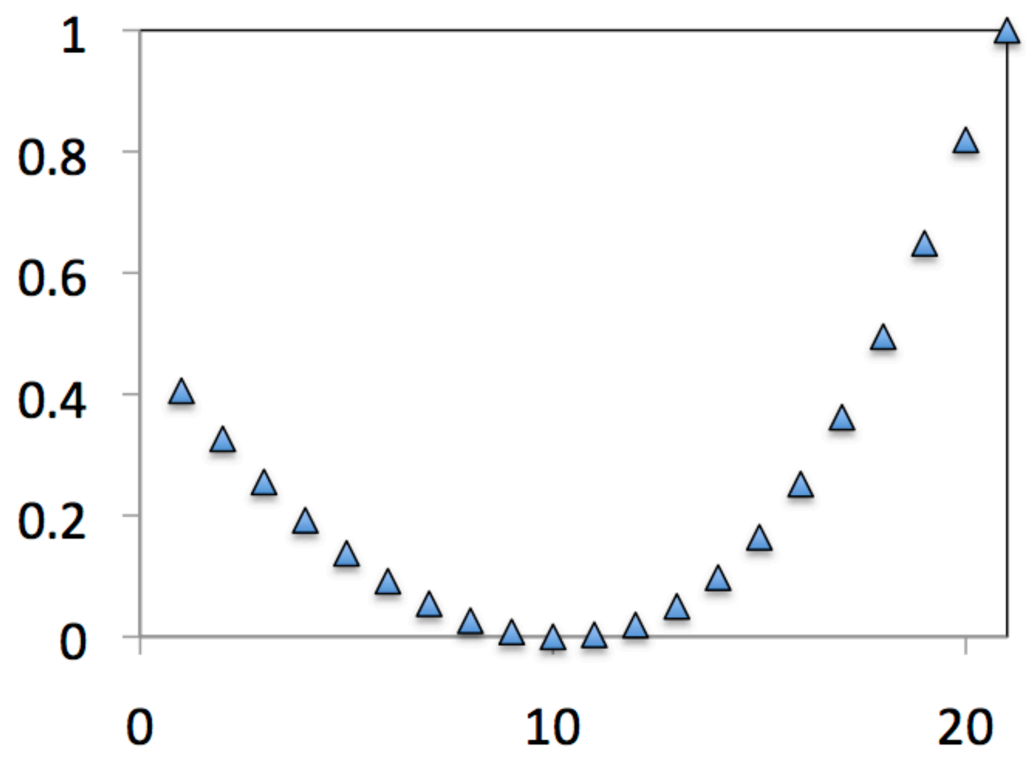
Method	Amazon	TripAdvisor	
	$4 \cdot \star$	$4 \cdot \star$	$5 \cdot \star$
SVM BoW-1g	35.78	35.41	46.41
SVM BoW-2g	28.26	33.68	<u>44.68</u>
SVM BoW-3g	<u>27.98</u>	33.50	45.12
SVM BoW-5g	28.02	<u>33.45</u>	46.41
Prc BoW-3g	26.45	34.73	43.58
SSE	25.30	34.22	42.88
SSE-W	**24.61**	**32.25**	**40.54**

Q&A

Thanks a million to "Evangelos Papalexakis" !

N-GRAM WEIGHTS IN SSE-W

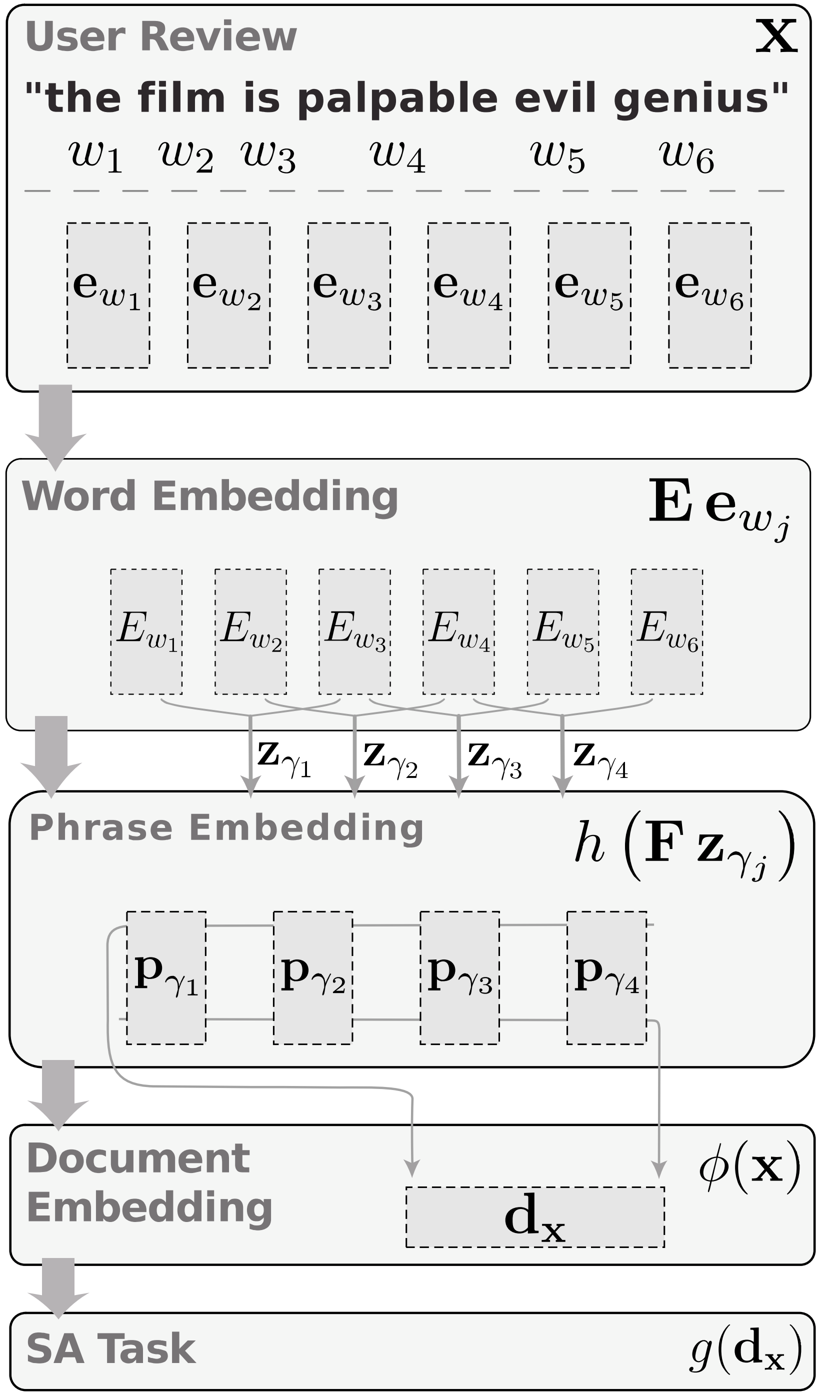
- SSE-W model was trained on Amazon with multi-class setting



Sentiment Analysis / Opinion Mining

- Sentiment Analysis (SA) deals with "computational treatment of opinion, sentiment, and subjectivity in text" [1]
- Prominent directions of opinion mining research include:
  - **Sentiment and subjectivity classification**
    - *Sentence-level* identifies subjective statements, and labels their sentiment
    - *Document-level* predicts overall sentiment expressed in whole text
  - **Feature-based** and **comparative SA** are structured data extraction problems
    - Feature-based detects entities:
      - object of the review, opinion holder, sentiment of opinion, related aspects
    - Comparative SA deals with opinions expressed with comparative sentences:
      - e.g., product-X is better than product-Y, but not as good as product-Z
  - **Opinion search and retrieval**
    - deals with indexing, retrieval and querying of opinionated documents
  - **Opinion spam**
    - detects fake reviews with undeserving positive or malicious negative opinions

[1] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval 2*



Previous method: sentiment classification based on supervised word embedding

[9] Dmitriy Bespalov and Bing Bai and Yanjun Qi and Ali Shokoufand eh.

Sentiment Classificatio n Based on Supervised Latent n-gram Analysis. *CIKM 2011*.

Experimental Results: Sentiment Classification (cont'd)

- In our previous work [9], different test-train split was used
  - Validating set was sampled from respective **training** sets
  - BoN was limited to only **127,000** features
- **Micro-average error** rate is reported
- Numbers marked with † (or ‡) are statistically significantly better than **SVM BoW-3g** with $p < 0.0001$ (or $p < 0.01$)

Method	Amazon 2 · ⋆	Amazon 4 · ⋆	TripAdvisor 2 · ⋆	TripAdvisor 4 · ⋆	TripAdvisor 5 · ⋆
SVM BoW-1g	11.10	30.31	8.89	33.54	43.93
SVM BoW-2g	7.45	25.28	7.47	32.27	42.34
SVM BoW-3g	<u>7.13</u>	<u>25.02</u>	<u>7.25</u>	<u>32.22</u>	<u>42.20</u>
SVM BoW-5g	7.34	25.67	7.43	32.55	42.31
Prc BoW-3g	7.41	27.49	7.31	31.99	41.29
LTC	7.12	27.10	8.33	33.40	42.69
SSE	**7.04**	**23.59**	**6.59**	**27.60**	**37.56**
SSE-W	**7.00**	**23.11**†	**6.43**‡	27.68†	38.09†

[9] Dmitriy Bespalov and Bing Bai and Yanjun Qi and Ali Shokoufandeh.
Sentiment Classification Based on Supervised Latent n-gram Analysis. *CIKM 2011*.

Experimental Results: Binary Topic Categorization

- Used **four** most frequent topics in training set of RCV1
- **500,000** most frequent phrases were retained in BoN
- **Macro-average error** rate is reported

Method	RCV1			
	CCAT	GCAT	MCAT	C15
SVM BoW-1g	6.45	5.66	5.70	7.95
SVM BoW-2g	5.82	<u>5.42</u>	5.60	7.62
SVM BoW-3g	<u>5.79</u>	5.53	<u>5.59</u>	<u>7.46</u>
SVM BoW-5g	5.89	5.72	5.75	7.55
SSE	5.74	4.79	**4.41**	6.21
SSE-W	**5.71**	**4.70**	4.45	**5.50**

Feed-forward Deep Architectures

Input Vector $\mathbf{x}$

Linear Projection $\mathbf{p}_1 = \mathbf{A}\mathbf{x} + \mathbf{b}$

Non-linear Transfer Function $\mathbf{p}_2 = \mathrm{h}(\mathbf{p}_1)$

$$\tanh(t) = \frac{e^{2t} - 1}{e^{2t} + 1} \qquad \mathrm{sigmoid}(t) = \frac{1}{1 + e^{-t}}$$

• • •

Optimization objective (i.e., NN Criterion) $\mathrm{L}(\mathbf{p}_k)$