Distributed Representations of Sentences and Documents

Quoc Le QVL@GOOGLE.COM
Tomas Mikolov TMIKOLOV@GOOGLE.COM
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Bag-of-Word as a Document Vector

BoW and Bo-N-grams have little sense about the semantics of the words

Distances between “powerful,” “strong” and “Paris” are equally distant
“powerful” should be closer to “strong” than “Paris.”

Bag-of-Word as a Document Vector

Documents that have "powerful," should have similar vectors as those that have "strong"*!*

Word Vectors as a Document Vector

A function on words' vectors of a document Doc = $[w_1, w_2, \ldots, w_n]$

$f(Doc) = g(h(w_1), h(w_2), \ldots, h(w_n))$

$h: Vocab \rightarrow R^d$

$g: R^{d \times n} \rightarrow R^{d'}$

h: could be 1-hot, Term-Doc, TF-iDF, …, Word2Vec

g: Concatenation, SUM, AVG, …

Word Vectors as a Document Vector

A function on words' vectors of a document Doc = $[w_1, w_2, \ldots, w_n]$

$f(Doc) = g(h(w_1), h(w_2), \ldots, h(w_n))$

h: Vocab $\rightarrow R^d$

g: $R^{d \times n} \rightarrow R^{d'}$

h: could be 1-hot, Term-Doc, TF-iDF, …, Word2Vec

g: unknown! Can we learn it?

Word2Vec

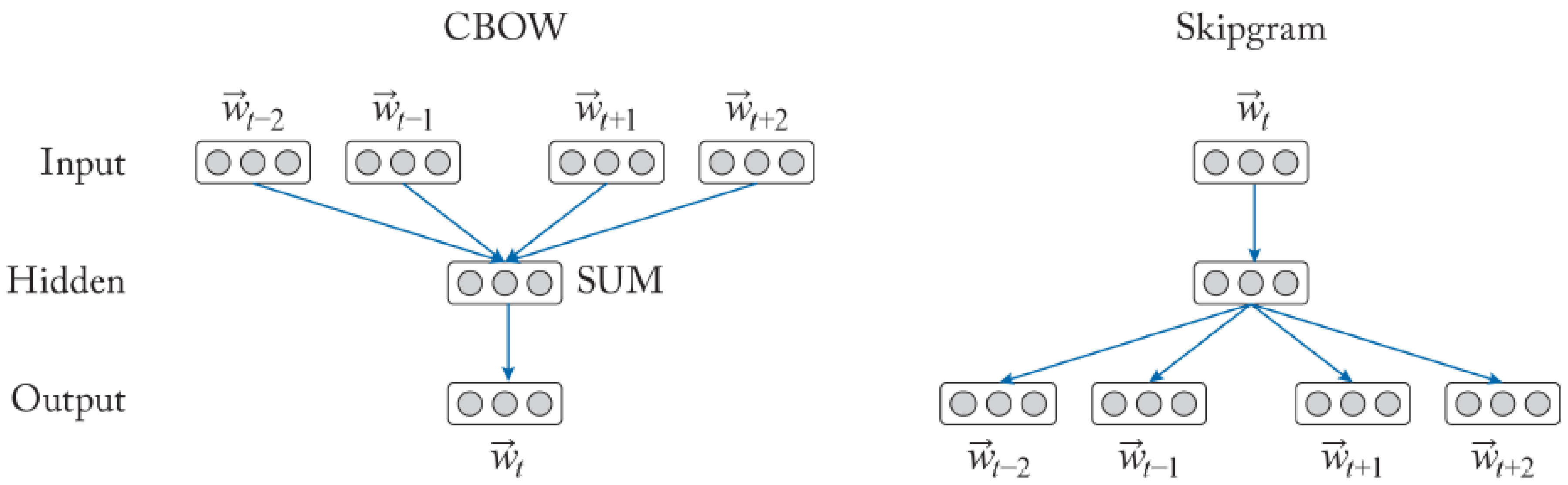
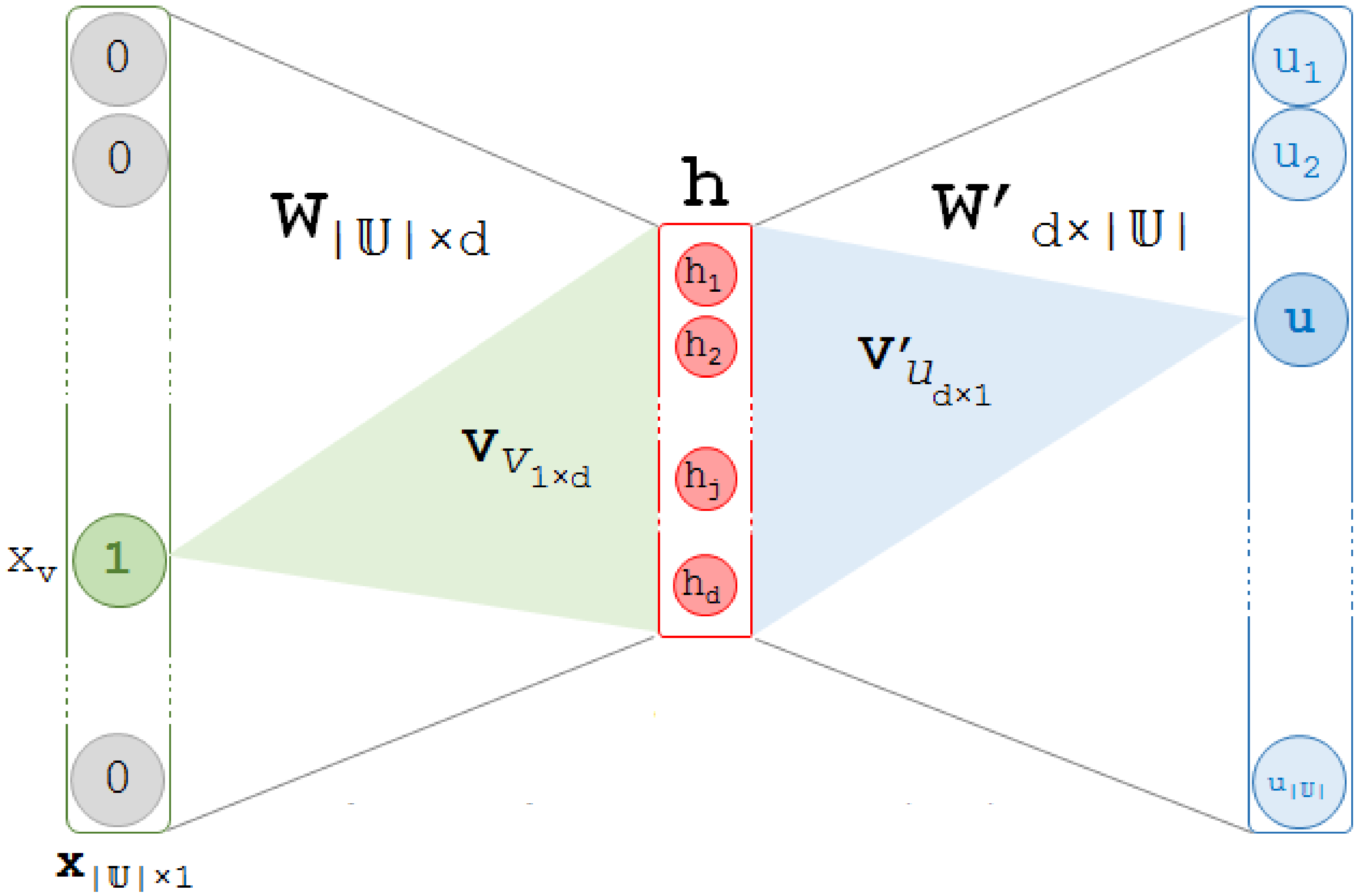


Figure 3.1: Learning architecture of the CBOW and Skip-gram models of Word2vec [Mikolov et al., 2013a].

Word2Vec

$$\boldsymbol{\sigma}\ ((\mathbf{h} = \mathbf{x}^{\mathrm{T}}\ \mathbf{W} + \mathbf{b})\ \mathbf{W'} + \mathbf{b}\ )$$



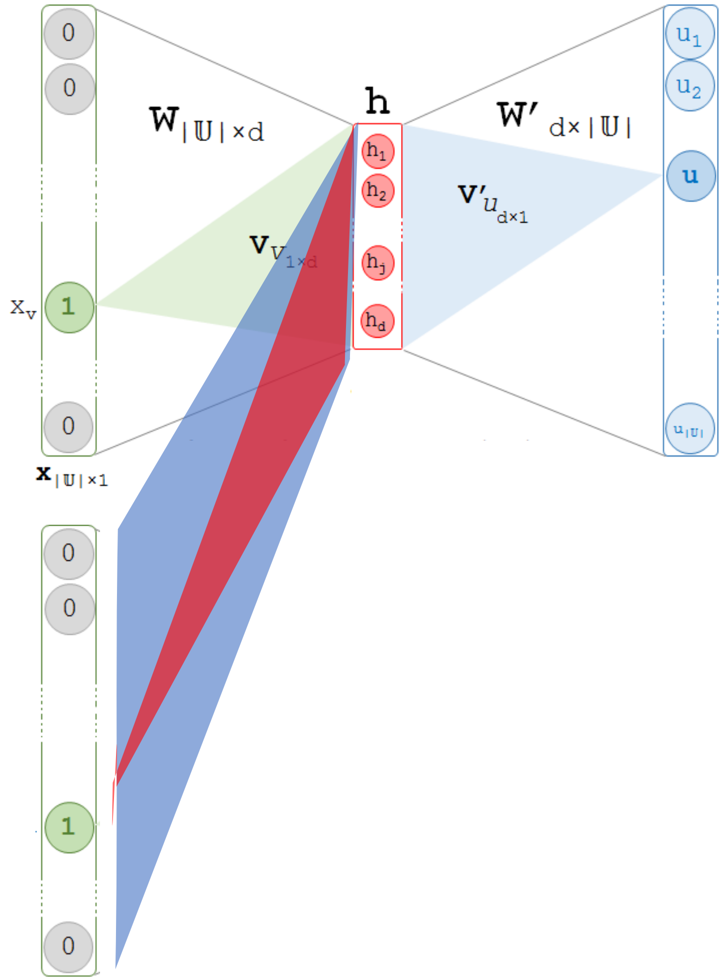
Word2Vec Predicts within a Context Window

The Context Window moves over the single stream of words.
No further context such as sentence, paragraph, or document is considered!
What if we say that the Window Context is moving within what document?

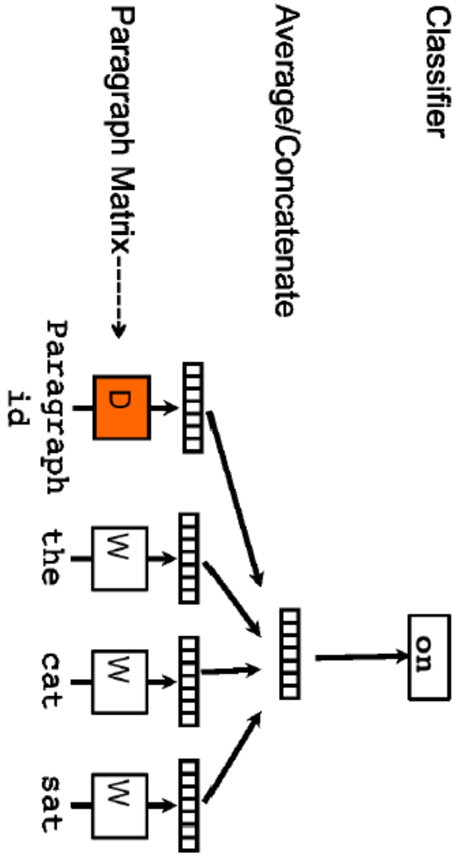
Distributed Memory Model of Paragraph Vectors (PV-DM)

$$\sigma\left(\left(\mathbf{h} = \mathbf{x}^{T}\, \mathbf{W} + \cancel{\mathbf{b}}\right) \mathbf{W}' + \cancel{\mathbf{b}}\right)$$

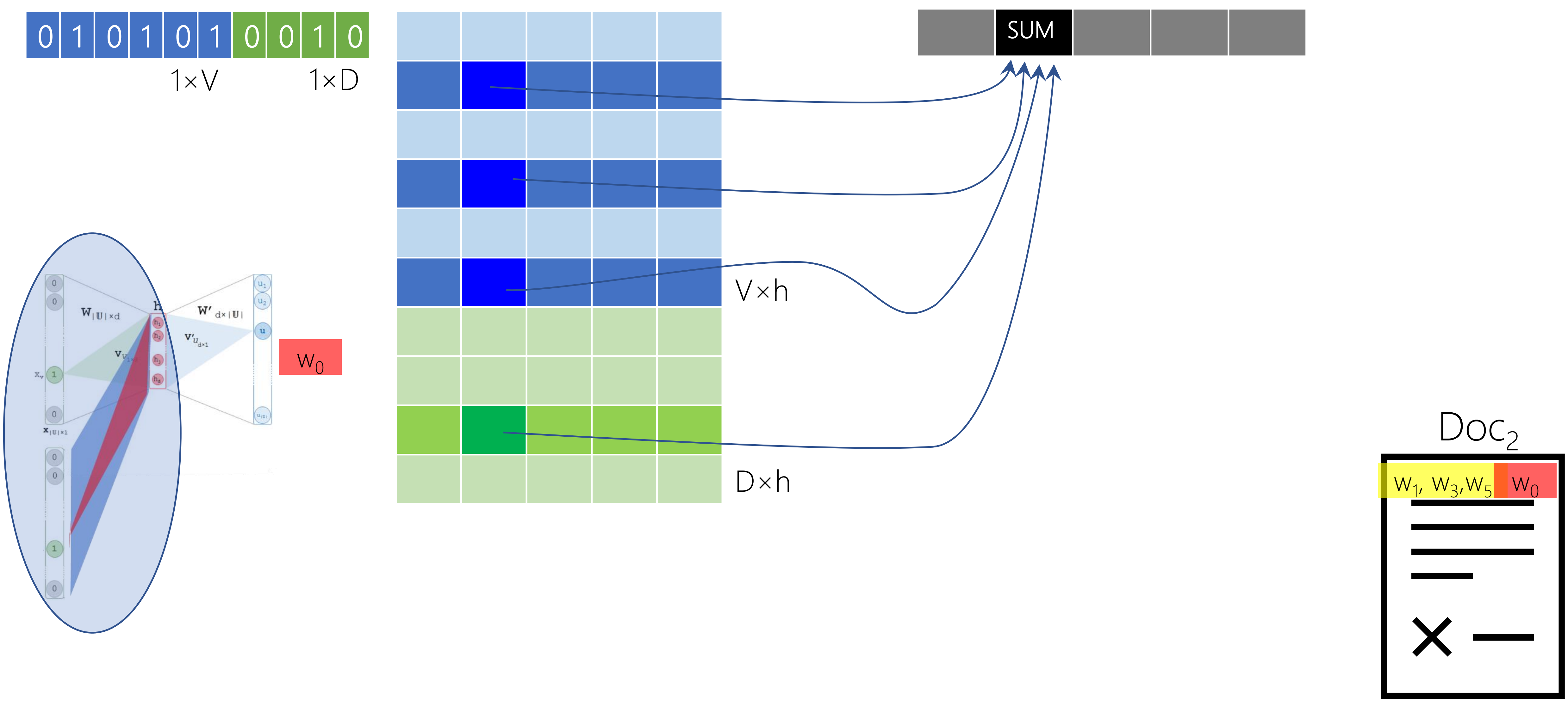
Window Context
1-hot
(bigrams of $w_i \rightarrow w_o$)
(occurrence vector)



Document
1-hot



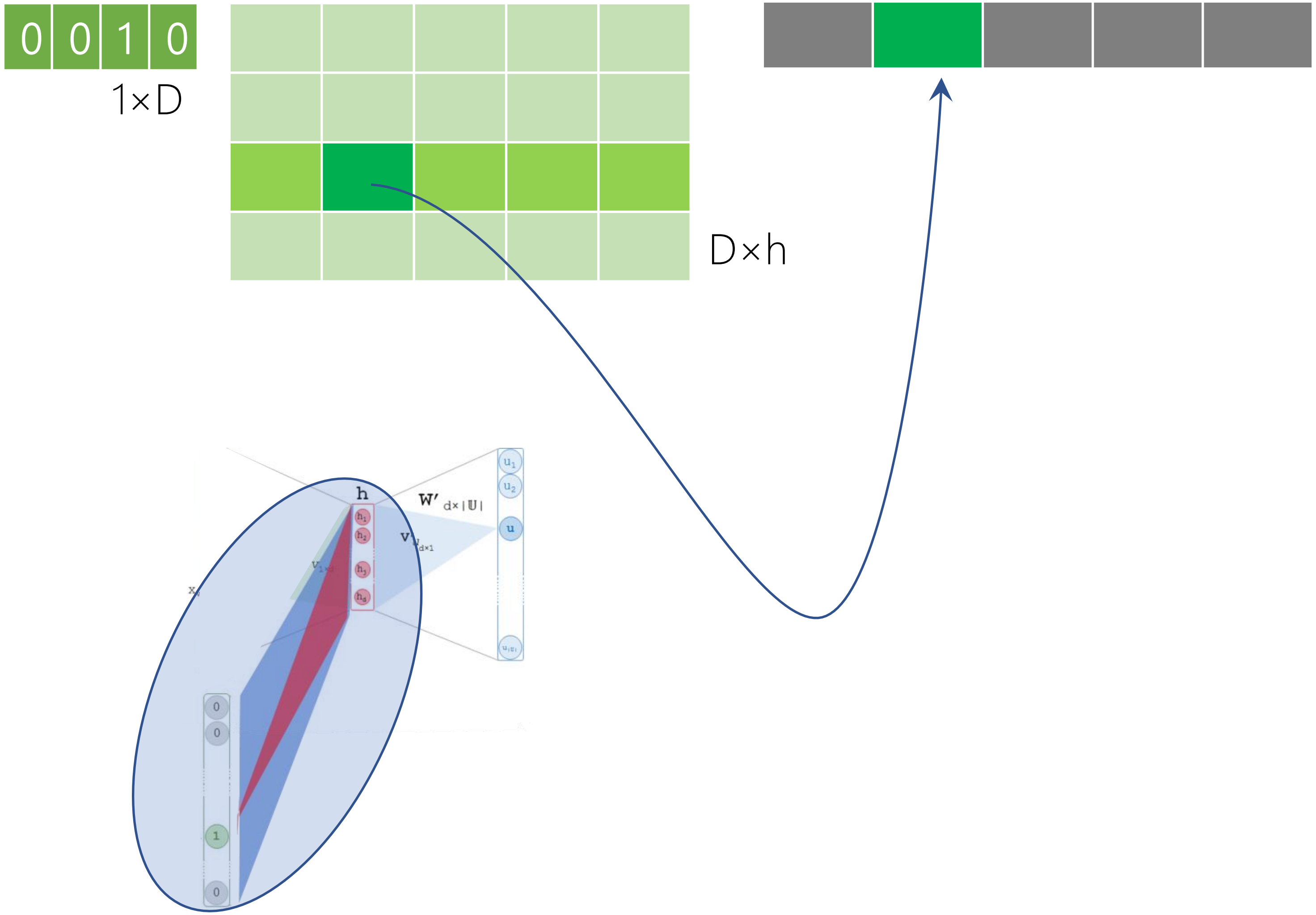
Distributed Memory Model of Paragraph Vectors (PV-DM)

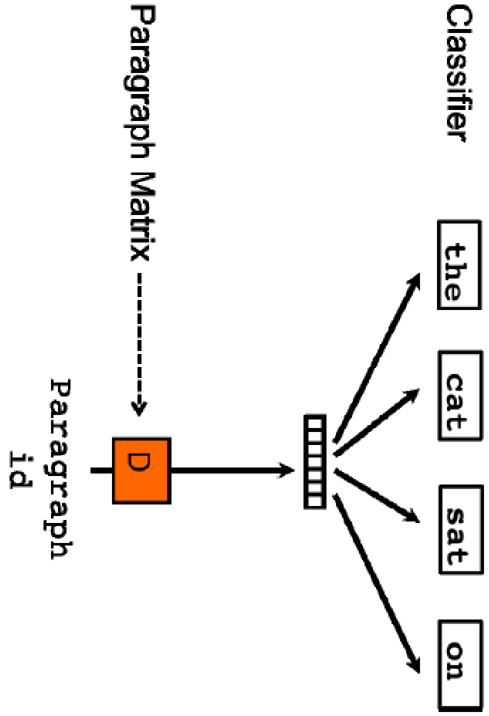


Paragraph Vector

Continuous dense vector representations for variable length of texts ranging from sentences to paragraph to documents.

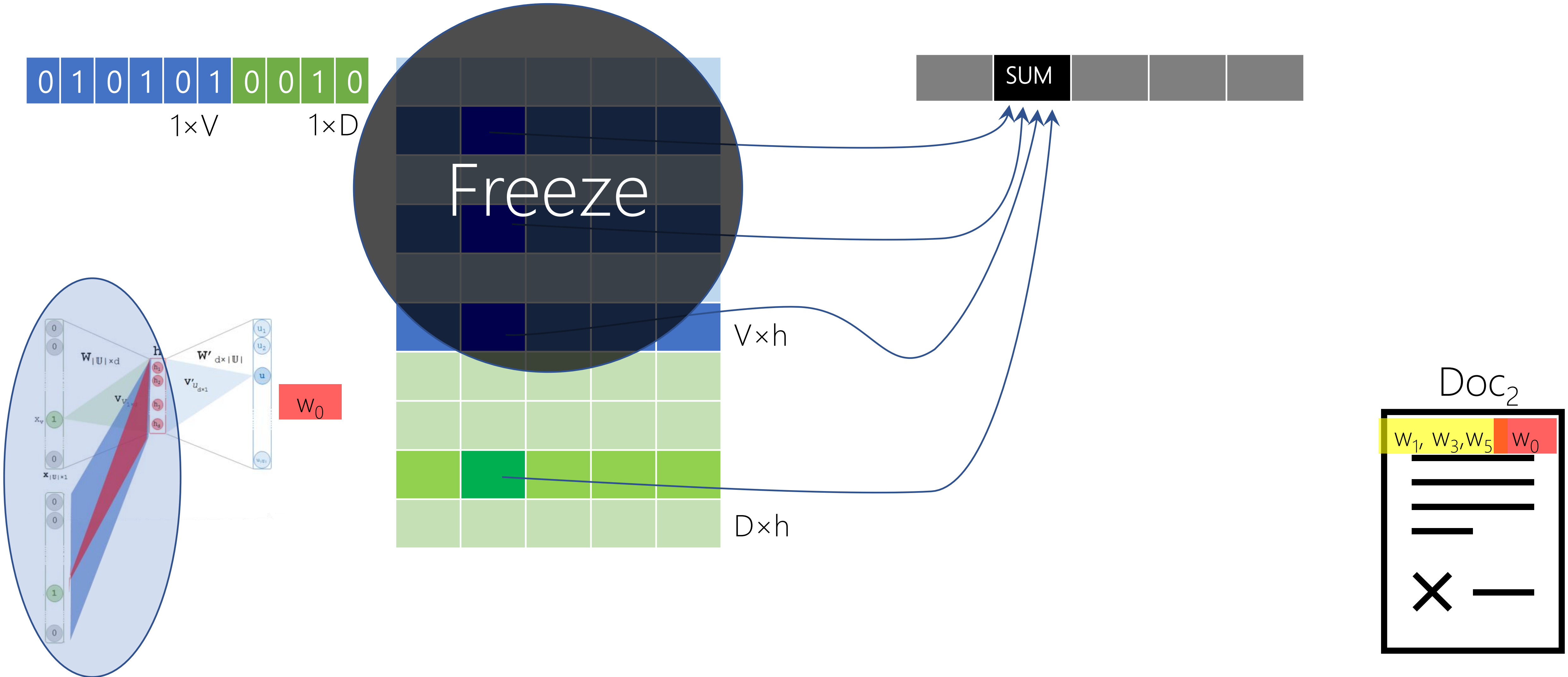
Distributed BoW of Paragraph Vectors (PV-DBOW)





Distributed Memory Model of Paragraph Vectors (PV-DM)

Pretrained Word Vectors for Unseen Documents



(PV-DM $\frown$ PV-DBOW), d=400, cw=8

fine-grained: {Very Negative, Negative, Neutral, Positive, Very Positive}
coarse-grained: {Negative, Positive}.

Table 1. The performance of our method compared to other approaches on the Stanford Sentiment Treebank dataset. The error rates of other methods are reported in (Socher et al., 2013b).

Model	Error rate (Positive/ Negative)	Error rate (Fine-grained)
Naïve Bayes (Socher et al., 2013b)	18.2 %	59.0%
SVMs (Socher et al., 2013b)	20.6%	59.3%
Bigram Naïve Bayes (Socher et al., 2013b)	16.9%	58.1%
Word Vector Averaging (Socher et al., 2013b)	19.9%	67.3%
Recursive Neural Network (Socher et al., 2013b)	17.6%	56.8%
Matrix Vector-RNN (Socher et al., 2013b)	17.1%	55.6%
Recursive Neural Tensor Network (Socher et al., 2013b)	14.6%	54.3%
Paragraph Vector	**12.2%**	**51.3%**

Table 2. The performance of Paragraph Vector compared to other approaches on the IMDB dataset. The error rates of other methods are reported in (Wang & Manning, 2012).

Model	Error rate
BoW (bnc) (Maas et al., 2011)	12.20 %
BoW (bΔt'c) (Maas et al., 2011)	11.77%
LDA (Maas et al., 2011)	32.58%
Full+BoW (Maas et al., 2011)	11.67%
Full+Unlabeled+BoW (Maas et al., 2011)	11.11%
WRRBM (Dahl et al., 2012)	12.58%
WRRBM + BoW (bnc) (Dahl et al., 2012)	10.77%
MNB-uni (Wang & Manning, 2012)	16.45%
MNB-bi (Wang & Manning, 2012)	13.41%
SVM-uni (Wang & Manning, 2012)	13.05%
SVM-bi (Wang & Manning, 2012)	10.84%
NBSVM-uni (Wang & Manning, 2012)	11.71%
NBSVM-bi (Wang & Manning, 2012)	8.78%
Paragraph Vector	**7.42%**

Paragraph Vector

- PV-DM is consistently better than PV-DBOW.
- The combination of PV-DM and PV-DBOW often work consistently better

- Using concatenation in PV-DM is often better than sum.

- A good guess of window size in many applications is between 5 and 12.

User Modeling

You are what you post!

Represent users by documents

a user = a document including all she said

You are what you post!

- Modeling User Personality (Computational Social Science)
  - Personality traits in psychology
  - Big Five: extraversion, emotional stability, agreeableness, conscientiousness, and openness to experience

- Modeling User Health Profile (Computational Epidemiology)
  - Privacy of the user, Ethical principles

- Modeling Gender and Ethnicity
  - First names → gender; Last names → ethnicity

- Modeling User Location

You are what you post! *within time.*

Predicting Personal Life Events from Streaming Social Content

Maryam Khodabakhsh
Ferdowsi University of Mashhad
maryamkhodabakhsh@stu.mail.ac.ir

Hossein Fani
University of New Brunswick
hfani@unb.ca

Fattane Zarrinkalam
Ryerson University
fzarrinkalam@ryerson.ca

Ebrahim Bagheri
Ryerson University
bagheri@ryerson.ca

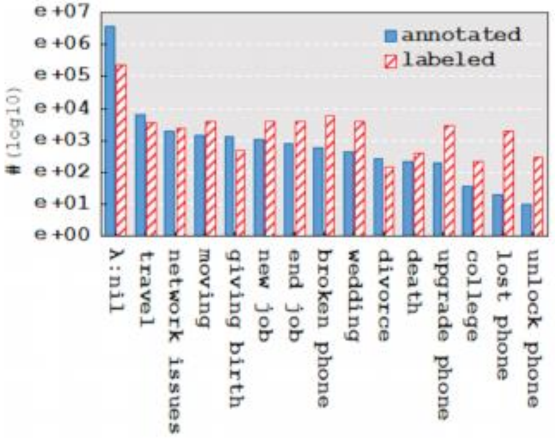
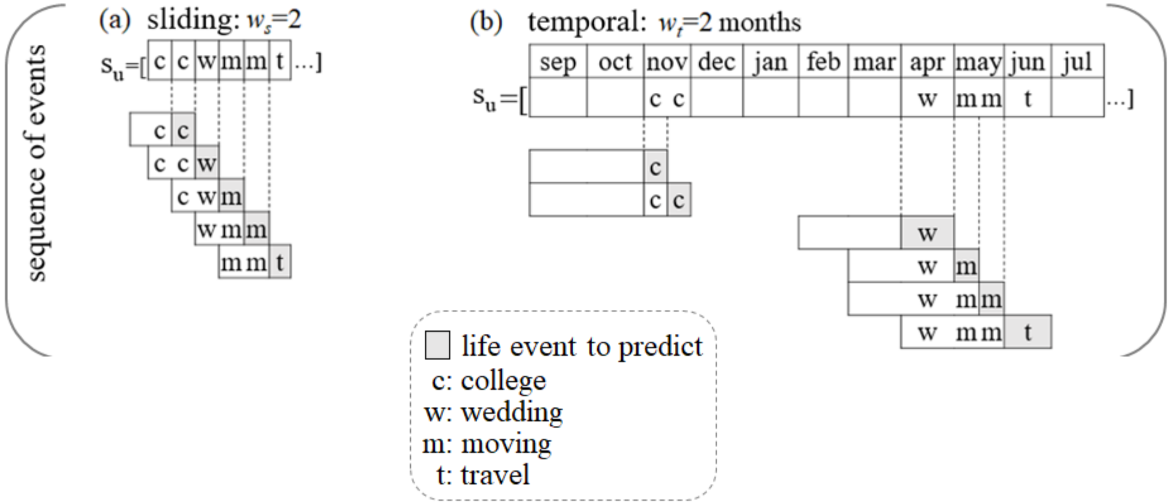


Figure 1: Distribution of personal life events by event class.



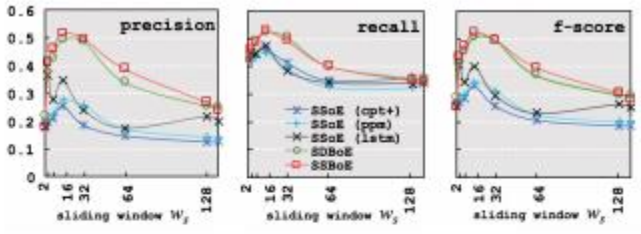


Figure 3: Comparative results of the *sliding* strategy.

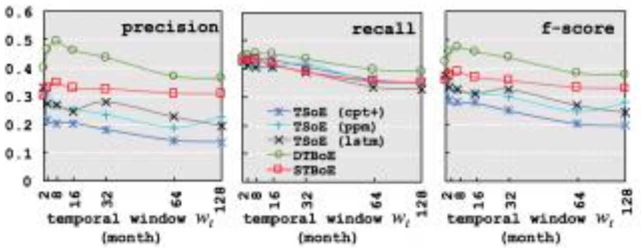


Figure 4: Comparative results of the *temporal* strategy.

You are what you post! *within time.*

User community detection via embedding of social network structure and temporal content☆

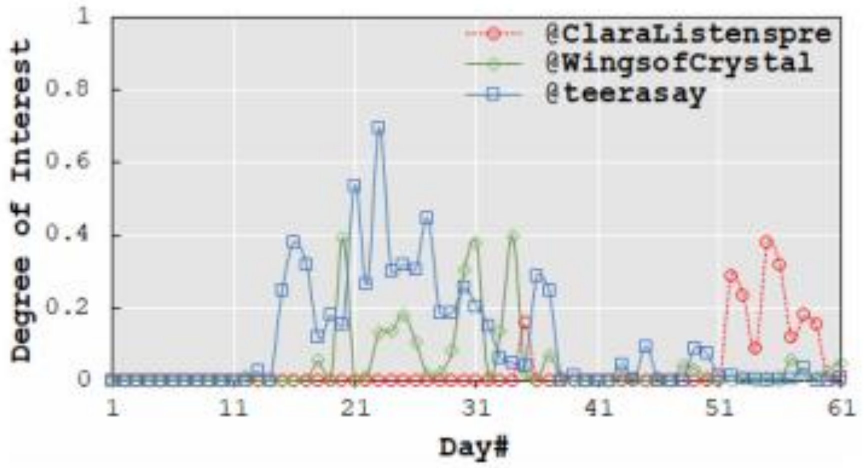


Fig. 1. Different temporal behaviour of three Twitter users with respect to the '*War in Afghanistan*' topic.

All users are interested in z_{44}='War in Afghanistan'

You are what you post! *within time.*

User community detection via embedding of social network structure and temporal content☆

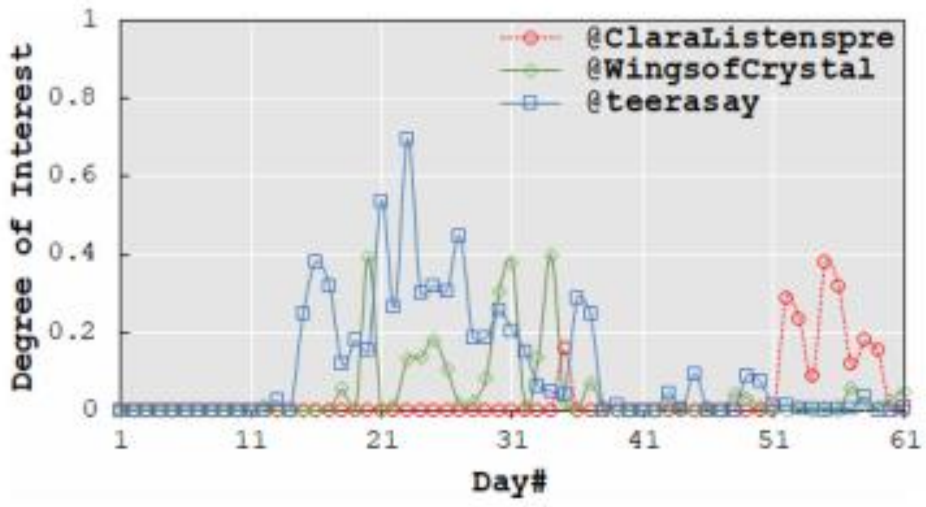
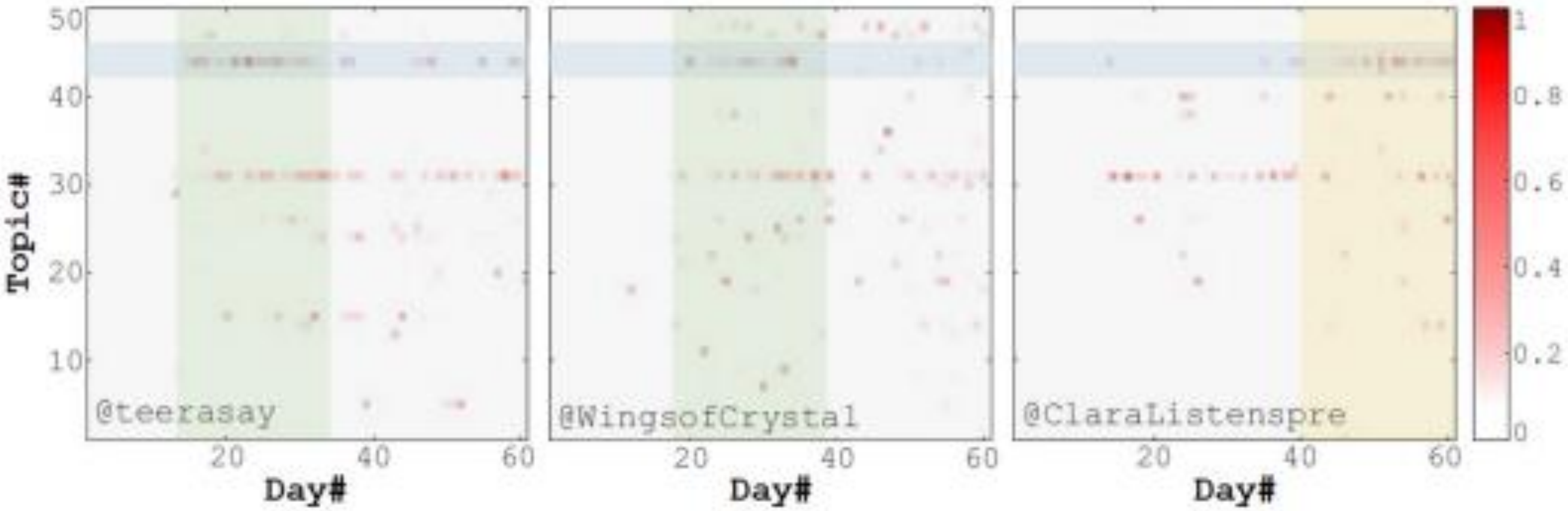


Fig. 1. Different temporal behaviour of three Twitter users with respect to the '*War in Afghanistan*' topic.

All users are interested in z_{44}='War in Afghanistan'
but not aligned in time!

You are what you post! *within time.*

User community detection via embedding of social network structure and temporal content☆



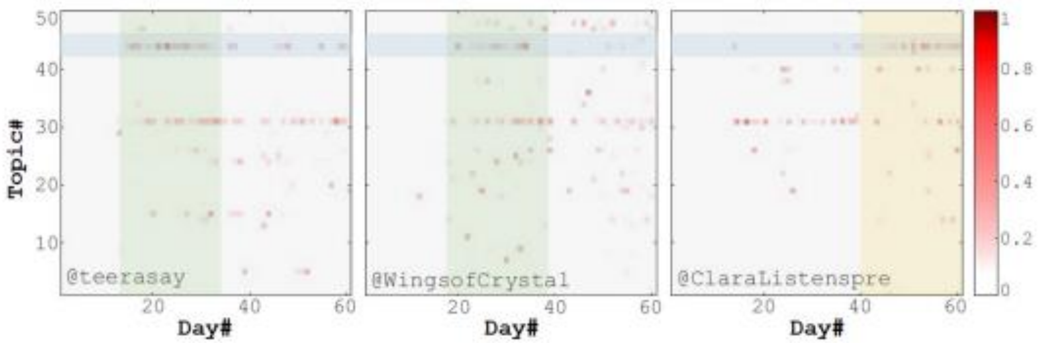
All users are interested in z_{44}='War in Afghanistan'
but not aligned in time!

Diachronically Like-minded User Community Detection

- User Clustering
  - Timeseries (Image) Clustering

User ⟷ Documents ⟶ User Vector ⟷ Document Vector

- How to include time?



Diachronically Like-minded User Community Detection

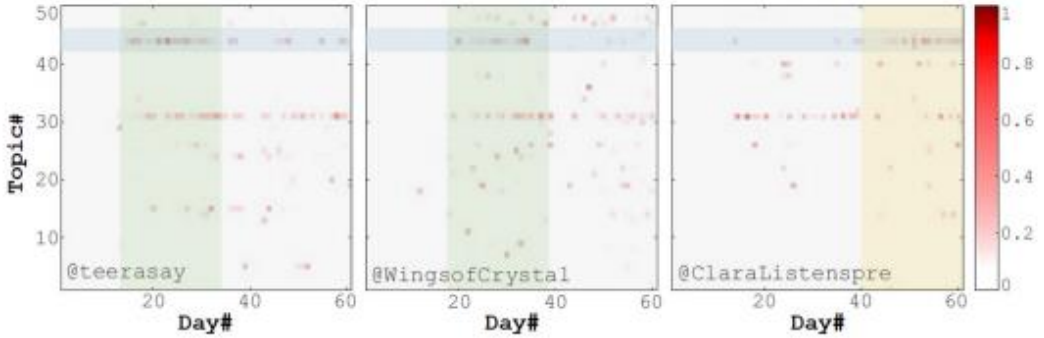
- User Clustering
  - User Vector Representation

User ⟷ Documents ⟶ User Vector ⟷ Document Vector

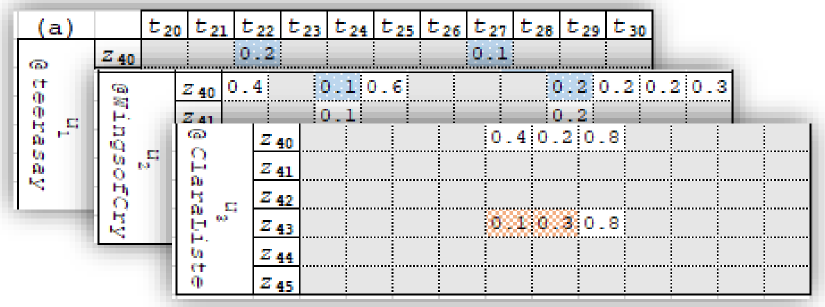
- How to include time?

User at time t ⟷ A document that has all she said at time t

User = [Doc_0, Doc_1, …, Doc_T]



Diachronically Like-minded User Community Detection



(a)		t_{20}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}
u_1 @teerasay	z_{40}			0.2					0.1			
	z_{41}											
	z_{42}											
	z_{43}	0.2	0.2	0.3								
	z_{44}	0.2	0.5	0.3	0.7	0.4	0.3	0.4	0.5	0.2	0.2	0.3
	z_{45}							0.2				
u_2 @WingsofCry	z_{40}	0.4		0.1	0.6				0.2	0.2	0.2	0.3
	z_{41}			0.1					0.2			
	z_{42}								0.2			
	z_{43}	0.3		0.1		0.9	0.5		0.2		0.4	
	z_{44}	0.4		0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.8	0.3
	z_{45}				0.5	0.2	0.5	0.2	0.2	0.4	0.2	
u_3 @ClaraListe	z_{40}					0.4	0.2	0.8				
	z_{41}											
	z_{42}											
	z_{43}					0.1	0.3	0.8				
	z_{44}											
	z_{45}											

(b)		t_{50}	t_{51}	t_{52}	t_{53}	t_{54}	t_{55}	t_{56}	t_{57}	t_{58}	t_{59}	t_{60}
u_3 @ClaraListe	z_{40}		0.1	0.2	0.1	0.2			0.1		0.2	
	z_{41}				0.1							
	z_{42}		0.1			0.1						
	z_{43}		0.2		0.1							
	z_{44}			0.3	0.2	0.1			0.1		0.1	
	z_{45}		0.1	0.1	0.1						0.1	

$$\text{User} = [\text{Doc}_0, \text{Doc}_1, \ldots, \text{Doc}_T]$$

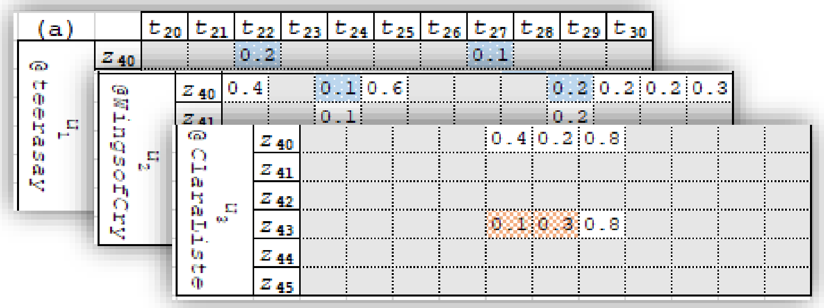
$$\downarrow$$

LDA

$$\downarrow$$

$$\text{User} = [[z^{(0)}_{1:K}], [z^{(1)}_{1:K}], \ldots, [z^{(T)}_{1:K}]]$$

Diachronically Like-minded User Community Detection



(a)		t_{20}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}
u_1 @teerasay	z_{40}			0.2					0.1			
	z_{41}											
	z_{42}											
	z_{43}	0.2	0.2	0.3								
	z_{44}	0.2	0.5	0.3	0.7	0.4	0.3	0.4	0.5	0.2	0.2	0.3
	z_{45}							0.2				
u_2 @WingsofCry	z_{40}	0.4		0.1	0.6				0.2	0.2	0.2	0.3
	z_{41}			0.1					0.2			
	z_{42}								0.2			
	z_{43}	0.3		0.1		0.9	0.5		0.2		0.4	
	z_{44}	0.4		0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.8	0.3
	z_{45}				0.5	0.2	0.5	0.2	0.2	0.4	0.2	
u_3 @ClaraListe	z_{40}					0.4	0.2	0.8				
	z_{41}											
	z_{42}											
	z_{43}					0.1	0.3	0.8				
	z_{44}											
	z_{45}											

(b)		t_{50}	t_{51}	t_{52}	t_{53}	t_{54}	t_{55}	t_{56}	t_{57}	t_{58}	t_{59}	t_{60}
u_3 @ClaraListe	z_{40}		0.1	0.2	0.1	0.2			0.1		0.2	
	z_{41}				0.1							
	z_{42}		0.1			0.1						
	z_{43}		0.2		0.1							
	z_{44}			0.3	0.2	0.1			0.1		0.1	
	z_{45}		0.1	0.1	0.1						0.1	

User = [Doc$_0$, Doc$_1$, …, Doc$_T$]

↓

LDA

↓

User = [[$z^{(0)}_{1:K}$], [$z^{(1)}_{1:K}$], …, [$z^{(T)}_{1:K}$]]

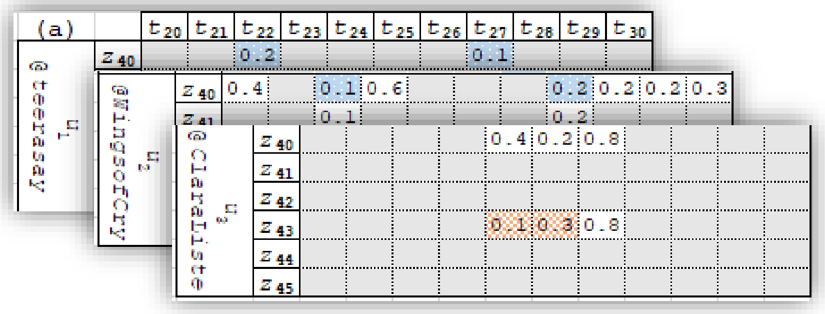
Two users are similar if they share more cells!

each cell = 1×1×1 cube $=\{u_i\}\times\{z_j\}\times\{t_k\}$

Shared cell = n×m×k cube

e.g., $\{u_1 u_2\}\times\{z_{44}\}\times\{t_{22}\ t_{23}\ \ldots\ t_{30}\}$

Diachronically Like-minded User Community Detection



(a)		t_{20}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}
@teerasay u_1	z_{40}			0.2					0.1			
	z_{41}											
	z_{42}											
	z_{43}	0.2	0.2	0.3								
	z_{44}	0.2	0.5	0.3	0.7	0.4	0.3	0.4	0.5	0.2	0.2	0.3
	z_{45}							0.2				
@WingsofCry u_2	z_{40}	0.4		0.1	0.6				0.2	0.2	0.2	0.3
	z_{41}			0.1					0.2			
	z_{42}								0.2			
	z_{43}	0.3		0.1		0.9	0.5		0.2		0.4	
	z_{44}	0.4		0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.8	0.3
	z_{45}				0.5	0.2	0.5	0.2	0.2	0.4	0.2	
@ClaraListe u_3	z_{40}					0.4	0.2	0.8				
	z_{41}											
	z_{42}											
	z_{43}					0.1	0.3	0.8				
	z_{44}											
	z_{45}											

(b)		t_{50}	t_{51}	t_{52}	t_{53}	t_{54}	t_{55}	t_{56}	t_{57}	t_{58}	t_{59}	t_{60}
@ClaraListe u_3	z_{40}		0.1	0.2	0.1	0.2			0.1		0.2	
	z_{41}				0.1							
	z_{42}		0.1			0.1						
	z_{43}		0.2		0.1							
	z_{44}			0.3	0.2	0.1			0.1		0.1	
	z_{45}		0.1	0.1	0.1						0.1	

Region of Like-mindedness (RoL) iff

$$y_t^u[z] \approx y_t^v[z]$$

Two users are similar if they share more cells!

each cell = 1×1×1 cube $=\{u_i\}\times\{z_j\}\times\{t_k\}$

Shared cell = n×m×k cube

e.g., $\{u_1 u_2\}\times\{z_{44}\}\times\{t_{22}\ t_{23}\ \ldots\ t_{30}\}$

Diachronically Like-minded User Community Detection

- User Clustering
  - ~~Timeseries (Image) Clustering~~
  - User2Vec: User Vector Representation: Two Similar Users → Similar Vectors