CSAIL

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
عضو في مؤسسة قطر
Member of Qatar Foundation

Semi-supervised Question Retrieval with Gated Convolutions

Tao Lei

joint work with Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti and Lluís Màrquez

NAACL 2016

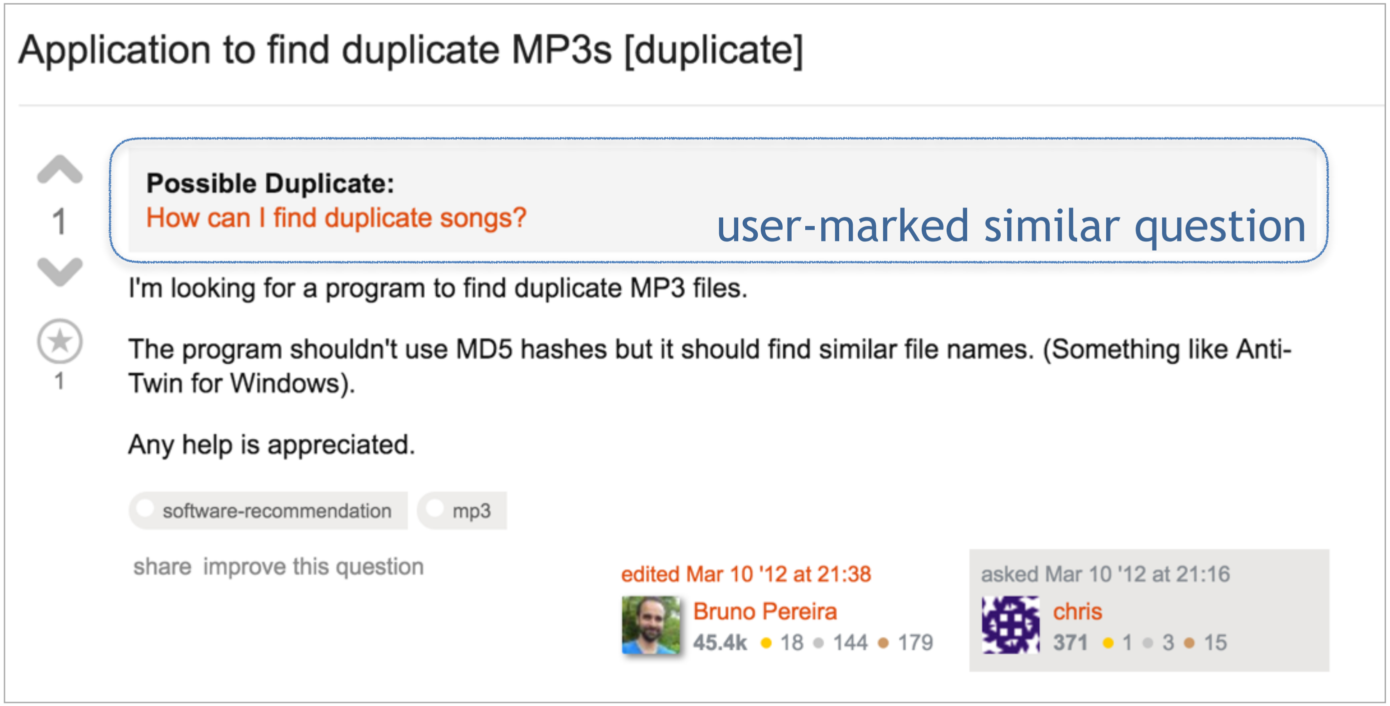
Our Task

Find similar questions given the user's input question

question from Stack Exchange AskUbuntu

Our Task

Find similar questions given the user's input question



question from Stack Exchange AskUbuntu

Our goal: automate this process as a solution for QA

Challenges

- Multi-sentence text contains irrelevant details

> Title: How can I boot Ubuntu from a USB ?
>
> Body: I bought a Compaq pc with Windows 8 a few months ago and now I want to install Ubuntu but still keep Windows 8. I tried Webi but when my pc restarts it read ERROR 0x000007b. I know that Windows 8 has a thing about not letting you have Ubuntu ...

> Title: When I want to install Ubuntu on my laptop I'll have to erase all my data. "Alonge side windows" doesnt appear
>
> Body: I want to install Ubuntu from a Usb drive. It says I have to erase all my data but I want to install it along side Windows 8. The "Install alongside windows" option doesn't appear …

- Forum user annotation is limited and noisy (more on this later)

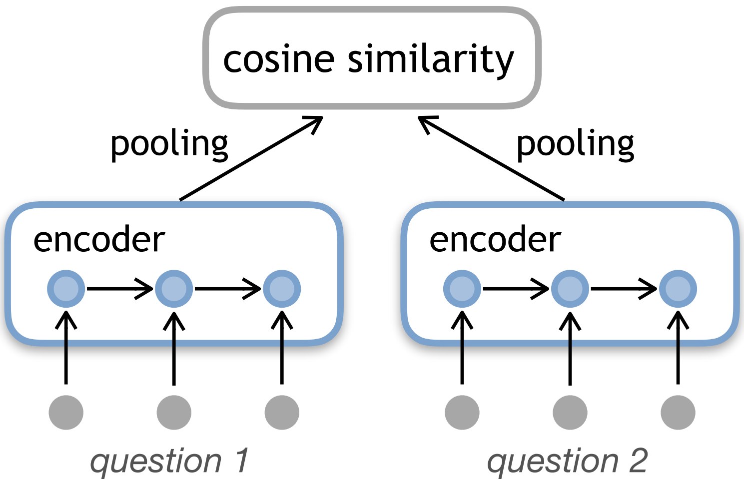
Solution

(1) a model to better represent the question text

(2) semi-supervised training to leverage raw text data

Model

Model Architecture*:



Choice of encoder:

LSTM, GRU, CNN … or:

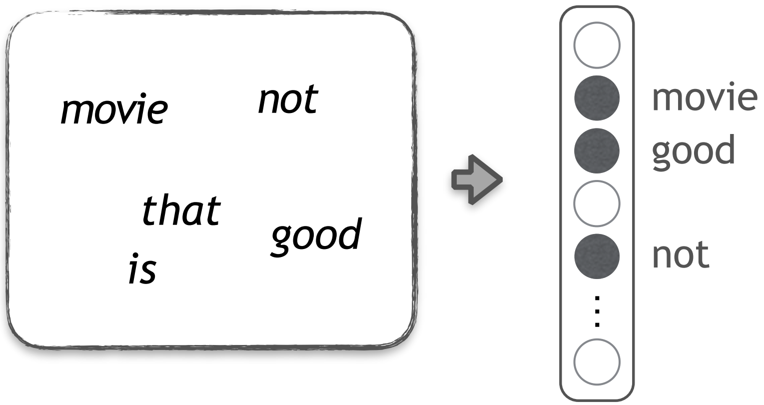
$$
\begin{aligned}
\mathbf{c}_t^{(3)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot \left(\mathbf{c}_{t-1}^{(1)} + \mathbf{W}_2 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1-\lambda_t) \odot (\mathbf{W}_1 \mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(3)} + \mathbf{b})
\end{aligned}
$$

Why this encoder (or equations)? How to understand it?

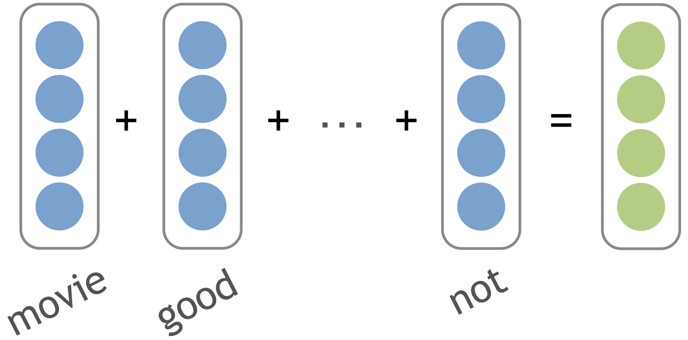
**Other architectures possible: (Feng et. al. 2015), (Tan et. al. 2015) etc.*

Sentence: *“the movie is not that good”*

Bag of words, TF-IDF



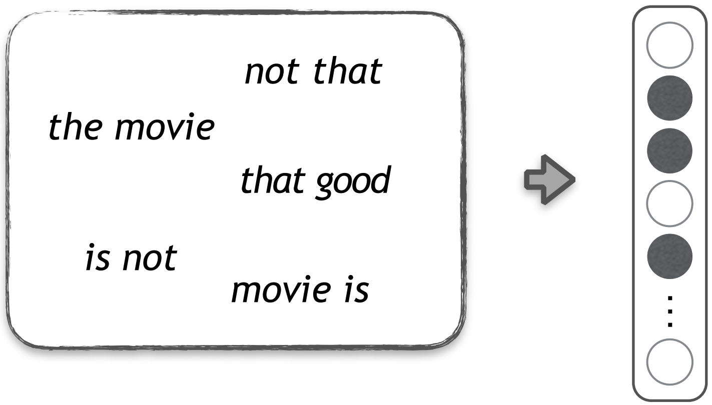
Neural Bag-of-words
(average embedding)



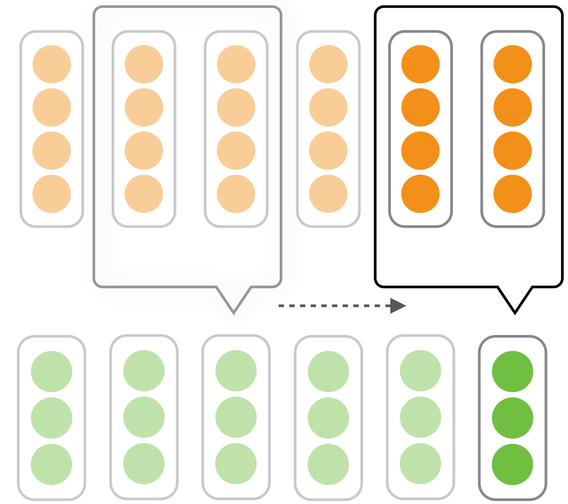
Sentence: *“the movie is not that good”*

Ngram Kernel

(N=2)



CNNs



Neural methods as a dimension-reduction of traditional methods

Sentence: ***"the movie is not that good"***

String Kernel

the movie
not _ good
is _ that
movie _ not
is not
not _ good
...

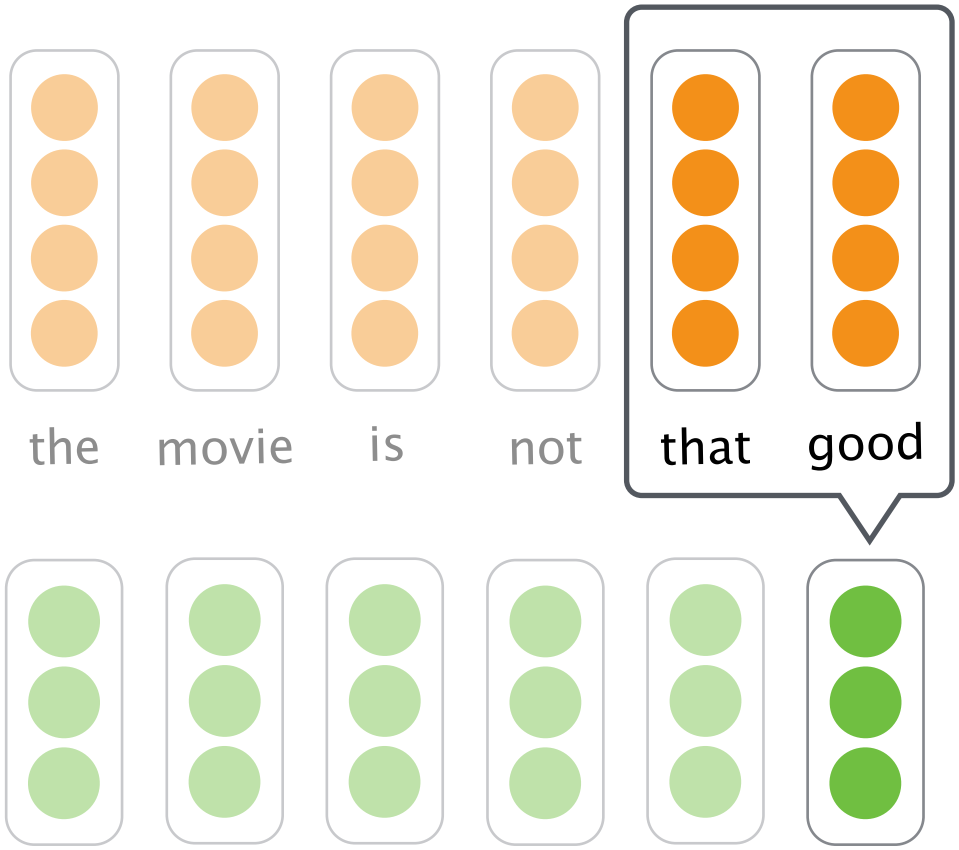
$$\Rightarrow \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{bmatrix} 0 \\ \lambda^0 \\ \lambda^2 \\ \vdots \\ \lambda^1 \\ 0 \end{bmatrix}\begin{array}{l} \\ \text{the movie} \\ \text{is \_ \_ good} \\ \\ \text{not \_ good} \\ \\ \end{array}$$

bigger feature space

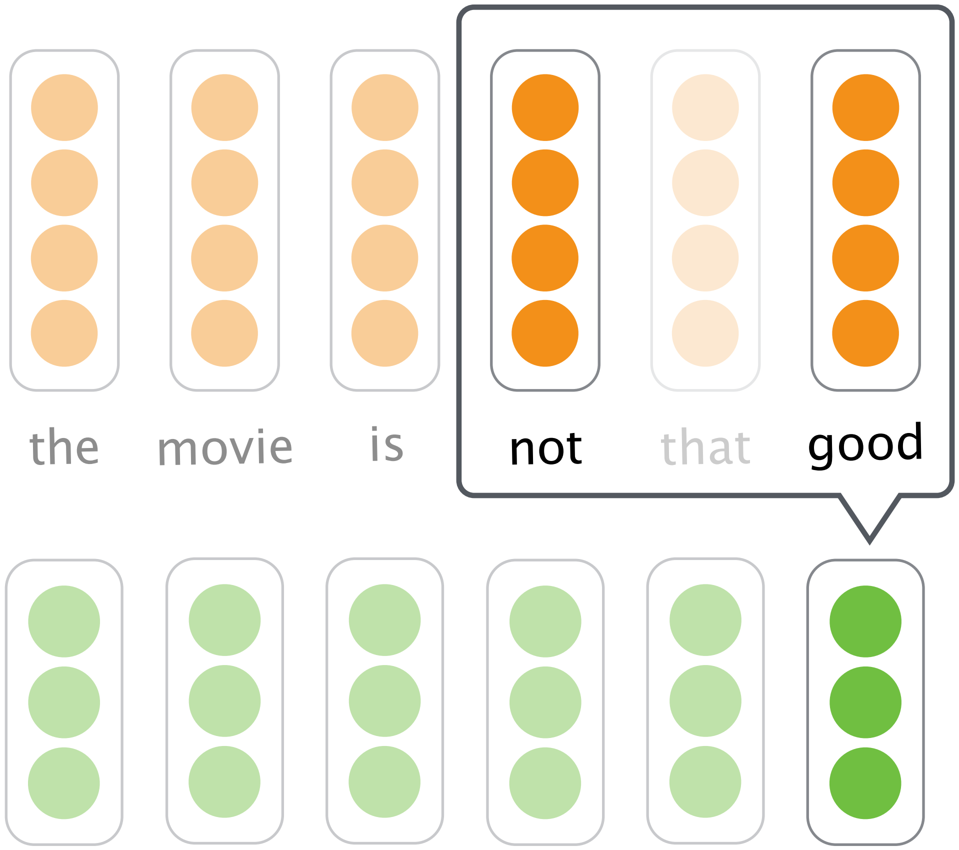
$\lambda \in (0, 1)$ penalize skips

Neural model inspired by this kernel method ?

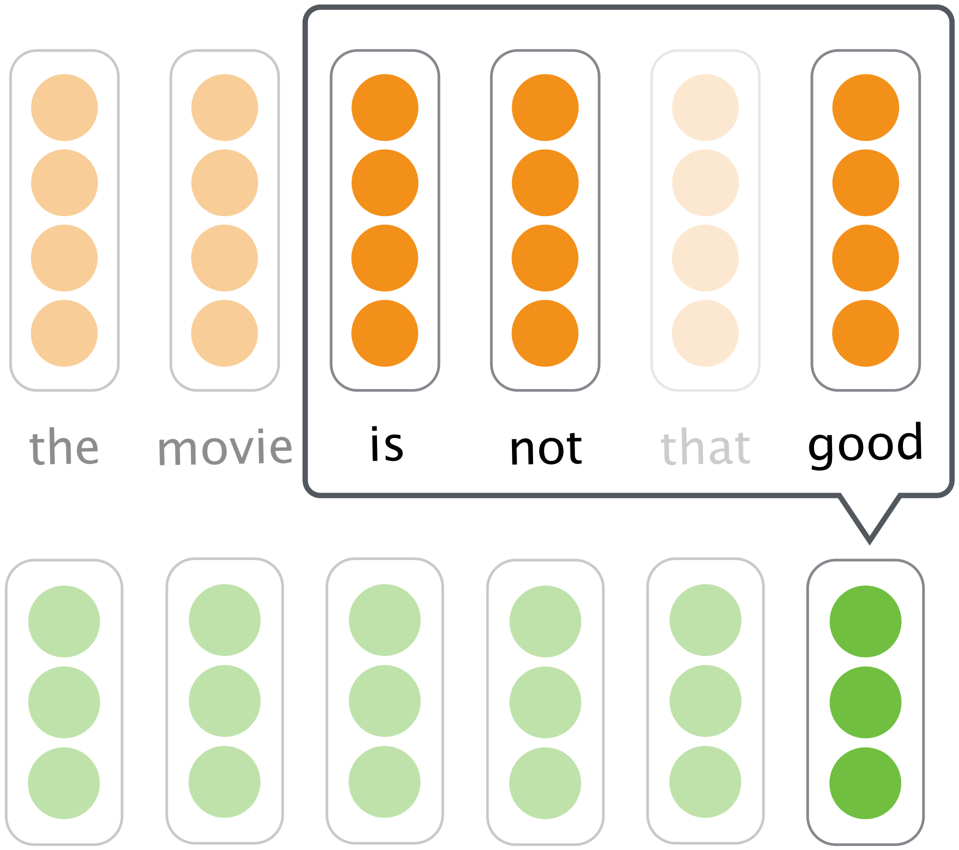
“string” convolution



“string” convolution



“string” convolution



Formulas in the case of 3gram

$$
\begin{aligned}
\mathbf{c}_t^{(3)} &= \lambda \cdot \mathbf{c}_{t-1}^{(3)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(2)} &= \lambda \cdot \mathbf{c}_{t-1}^{(2)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(1)} + \mathbf{W}_2 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(1)} &= \lambda \cdot \mathbf{c}_{t-1}^{(1)} + (1-\lambda) \cdot (\mathbf{W}_1 \mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(3)} + \mathbf{b})
\end{aligned}
$$

Formulas in the case of 3gram

$$
\begin{aligned}
\mathbf{c}_t^{(3)} &= \lambda \cdot \mathbf{c}_{t-1}^{(3)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(2)} &= \lambda \cdot \mathbf{c}_{t-1}^{(2)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(1)} + \mathbf{W}_2 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(1)} &= \boxed{\lambda} \cdot \mathbf{c}_{t-1}^{(1)} + (1-\lambda) \cdot \left(\mathbf{W}_1 \mathbf{x}_t\right) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(3)} + \mathbf{b})
\end{aligned}
$$

penalize skip grams

weighted average of 1grams (to 3grams) up to position *t*

Formulas

$$
\begin{aligned}
\mathbf{c}_t^{(3)} &= \lambda \cdot \mathbf{c}_{t-1}^{(3)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(2)} &= \lambda \cdot \mathbf{c}_{t-1}^{(2)} + (1-\lambda) \cdot \left(\mathbf{c}_{t-1}^{(1)} + \mathbf{W}_2 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(1)} &= \lambda \cdot \mathbf{c}_{t-1}^{(1)} + (1-\lambda) \cdot (\mathbf{W}_1 \mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(3)} + \mathbf{b})
\end{aligned}
$$

$$
\lambda = 0 : \quad \mathbf{c}_t^{(3)} = \mathbf{W}_1 \mathbf{x}_{t-2} + \mathbf{W}_2 \mathbf{x}_{t-1} + \mathbf{W}_3 \mathbf{x}_t \qquad \text{(one-layer CNN)}
$$

Gated version

$$
\begin{aligned}
\mathbf{c}_t^{(3)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot \left(\mathbf{c}_{t-1}^{(1)} + \mathbf{W}_2 \mathbf{x}_t\right) \\
\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1-\lambda_t) \odot (\mathbf{W}_1 \mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(3)} + \mathbf{b})
\end{aligned}
$$

$$
\lambda_t = \sigma\left(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}'\right)
$$

adaptive decay controlled by gate

Training

- Amount of annotation is scarce

# of unique questions	*167,765*
# of marked questions	*12,584*
# of marked pairs	*16,391*

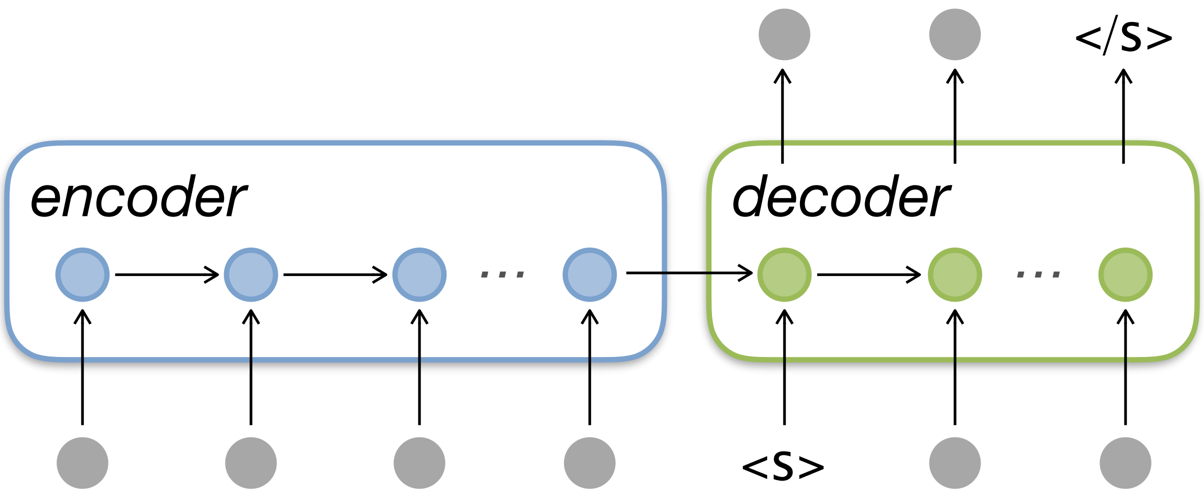
forum users only identify a few similar pairs

only 10% of the number unique questions

Ideally, want to use all questions available

Pre-training Encoder-Decoder Network

Encoder trained to pull out important (summarized) information



encode question body/title re-generate question title

Pre-training recently applied to classification task

- *Semi-supervised Sequence Learning. Dai and Le. 2015*

Evaluation Set-up

Dataset: AskUbuntu 2014 dump

pre-train on 167k, fine-tune on 16k

evaluate using 8k pairs (50/50 split for dev/test)

Baselines: TF-IDF, BM25 and SVM reranker

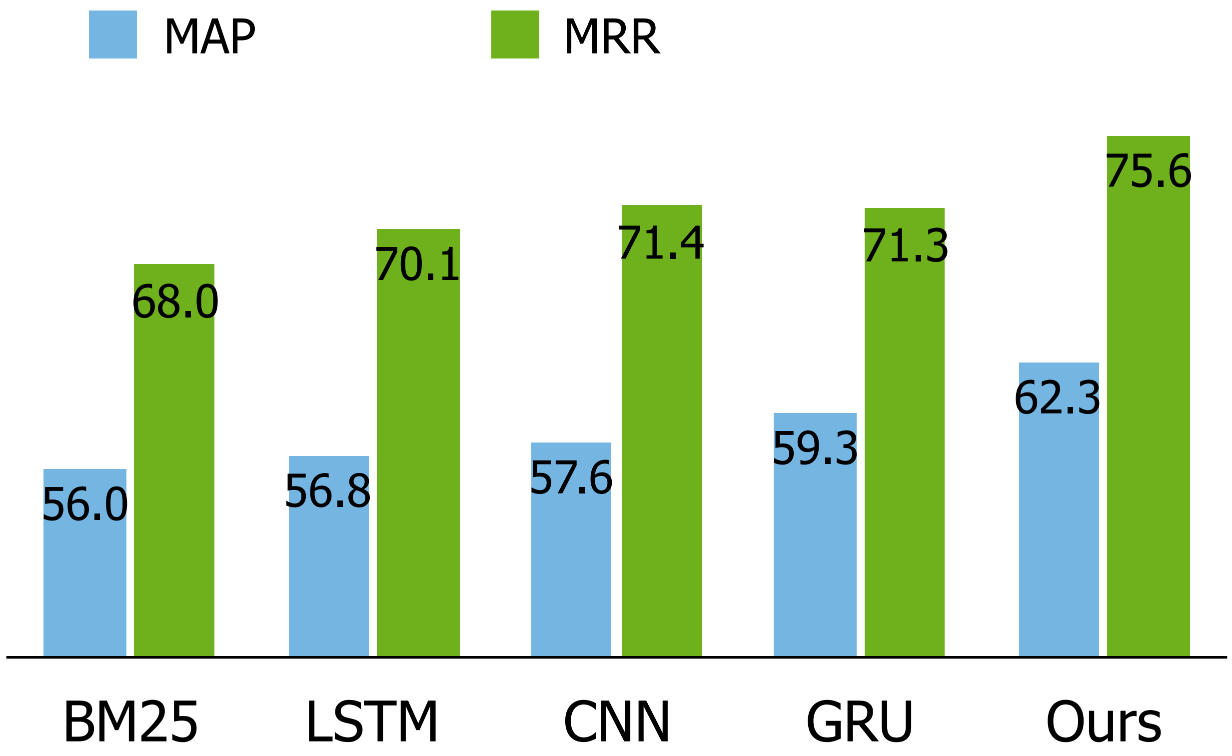
CNNs, LSTMs and GRUs

Grid-search: learning rate, dropout, pooling, filter size, pre-training, …

5 independent runs for each config.

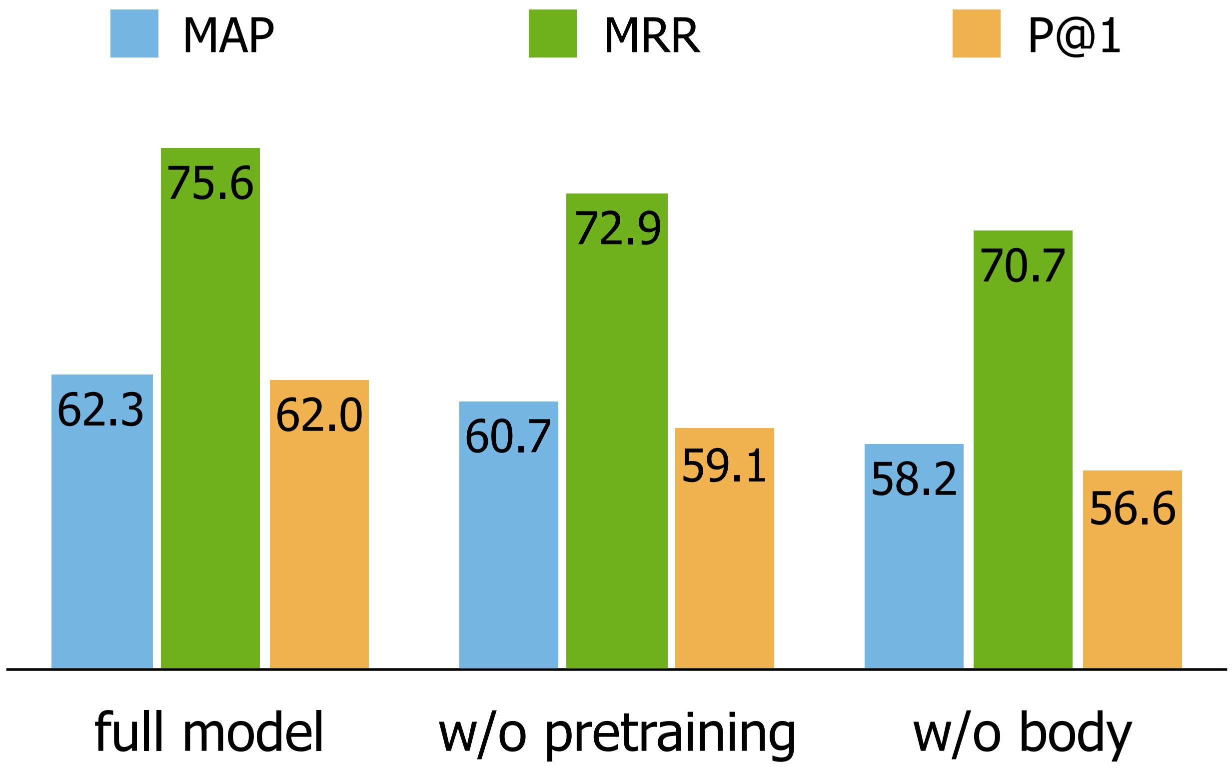
> 500 runs in total

Overall Results

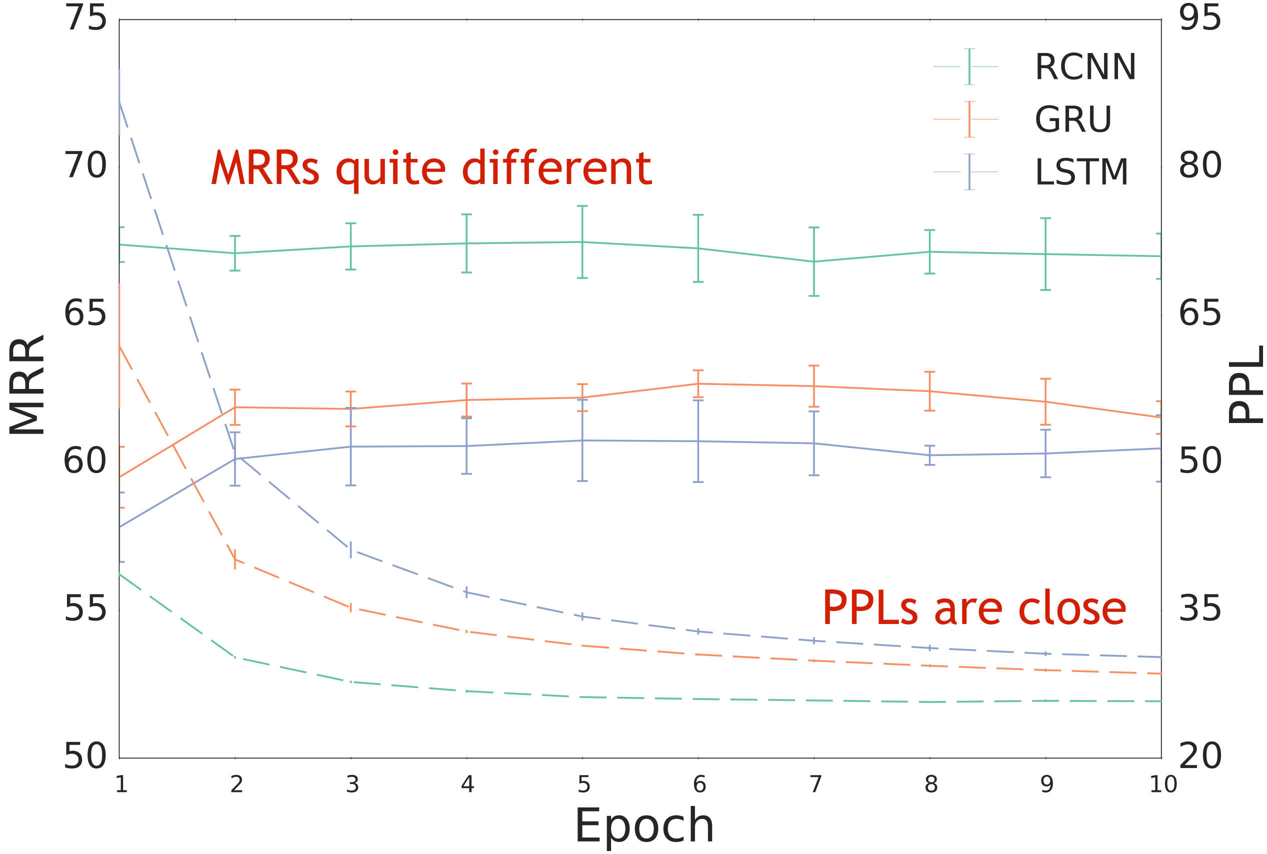


Our improvement is significant

Analysis

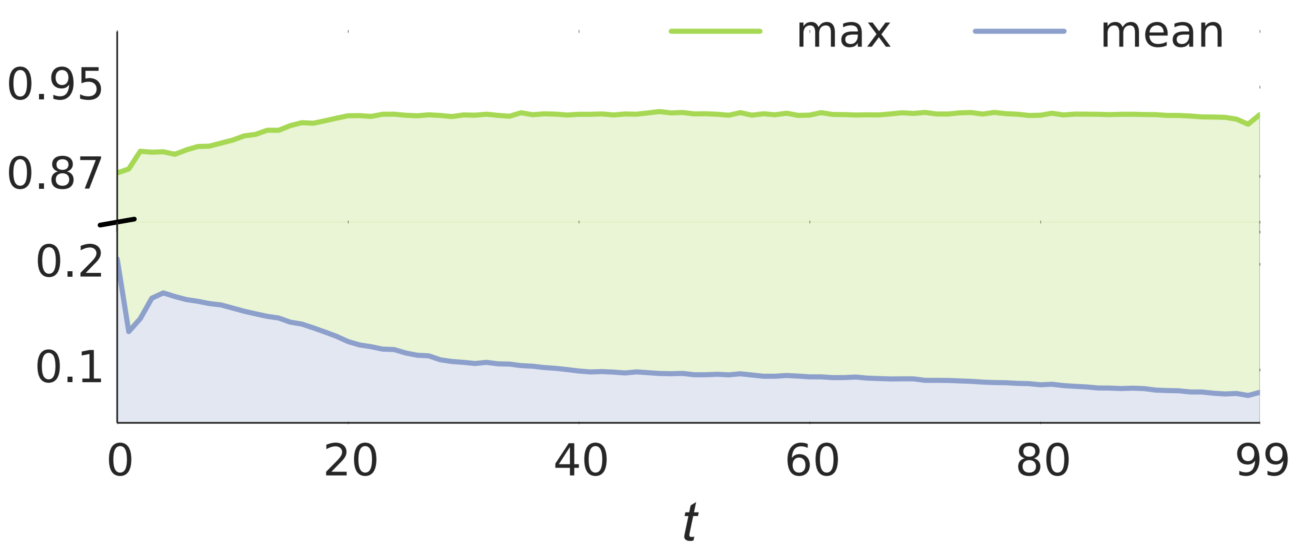

MAP
MRR
P@1
75.6
72.9
70.7
62.3
62.0
60.7
59.1
58.2
56.6
full model
w/o pretraining
w/o body

Pre-training



MRR on the dev set versus Perplexity on a heldout corpus

Decay Factor (Neural Gate)

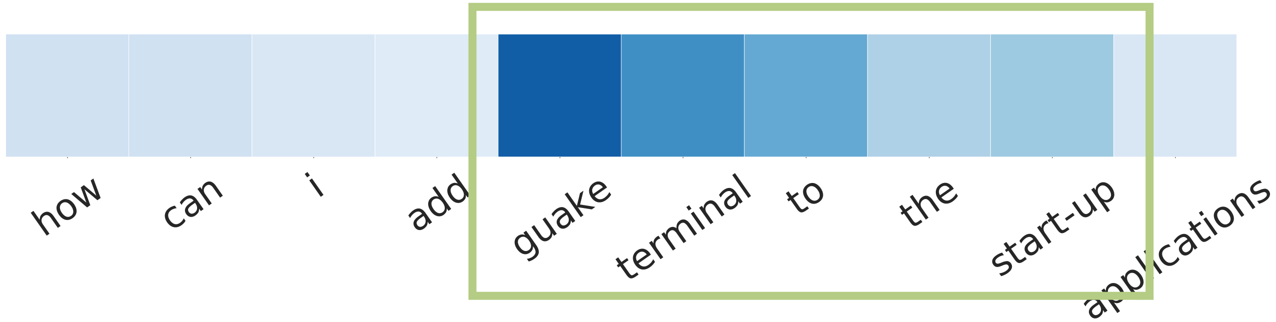


$$\mathbf{c}_t^{(3)} = \lambda \odot \mathbf{c}_{t-1}^{(3)} + \boxed{(1-\lambda)} \odot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right)$$

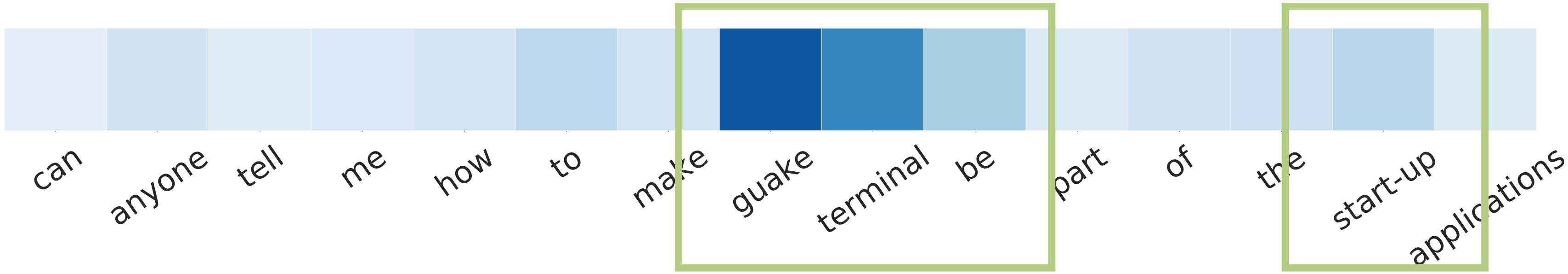
Analyze the weight vector over time

Case Study (using a scalar decay)

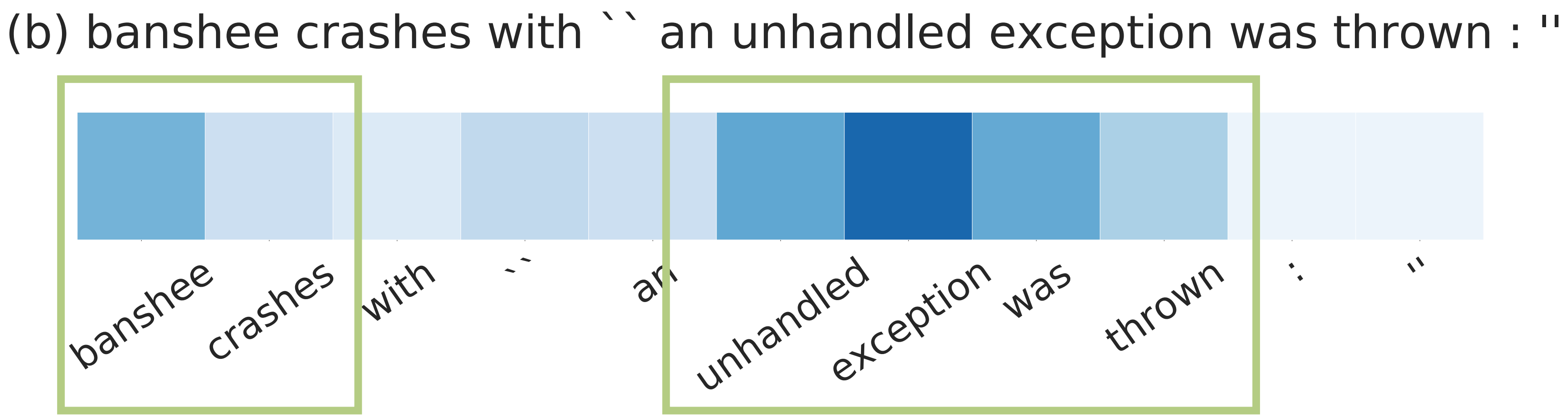
(a) how can i add guake terminal to the start-up applications



(f) can anyone tell me how to make guake terminal be part of the start-up applications

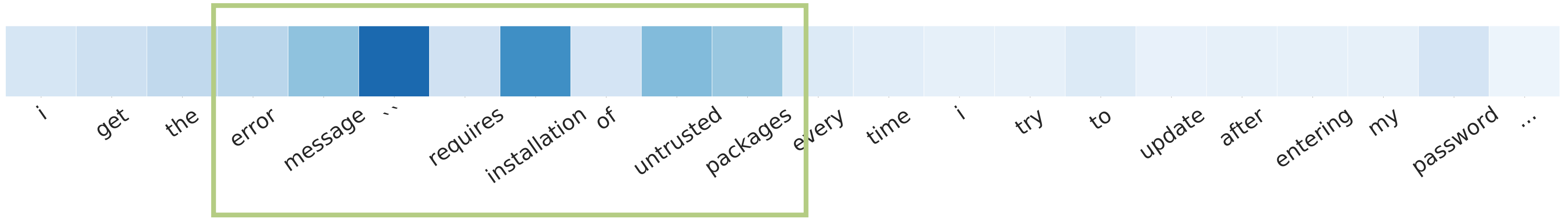


Case Study (using a scalar decay)

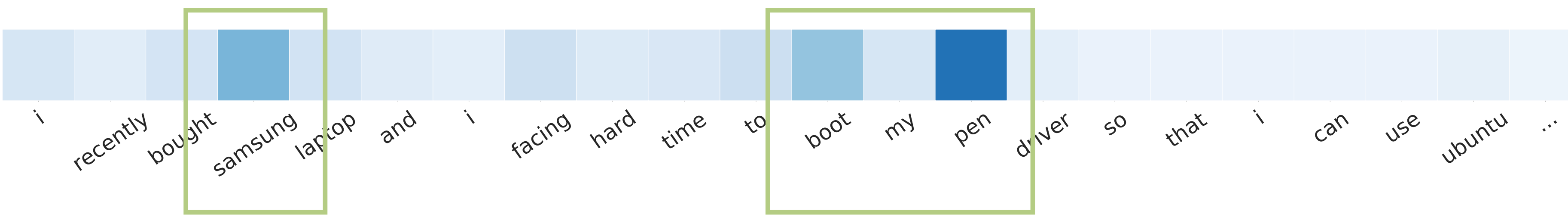


Case Study (using a scalar decay)

(c) i get the error message `` requires installation of untrusted packages every time i try to update after entering my password ...



(d) i recently bought samsung laptop and i facing hard time to boot my pen driver so that i can use ubuntu ...



Conclusions

- AskUbuntu data as a natural benchmark for retrieval and summarization tasks
- Neural model with good intuition and understanding (e.g. attention) can potentially lead to good performance

https://github.com/taolei87/askubuntu

https://github.com/taolei87/rcnn

Method	Pooling	Dev				Test			
		MAP	MRR	P@1	P@5	MAP	MRR	P@1	P@5
BM25	-	52.0	66.0	51.9	42.1	56.0	68.0	53.8	42.5
TF-IDF	-	54.1	68.2	55.6	45.1	53.2	67.1	53.8	39.7
SVM	-	53.5	66.1	50.8	43.8	57.7	71.3	57.0	43.3
CNNs	mean	58.5	71.1	58.4	46.4	57.6	71.4	57.6	43.2
LSTMs	mean	58.4	72.3	60.0	46.4	56.8	70.1	55.8	43.2
GRUs	mean	59.1	74.0	62.6	47.3	57.1	71.4	57.3	43.6
RCNNs	last	59.9	74.2	63.2	48.0	60.7	72.9	59.1	45.0
LSTMs + pre-train	mean	58.3	71.5	59.3	47.4	55.5	67.0	51.1	43.4
GRUs + pre-train	last	59.3	72.2	59.8	48.3	59.3	71.3	57.2	44.3
RCNNs + pre-train	last	**61.3***	**75.2**	**64.2**	**50.3***	**62.3***	**75.6***	**62.0**	**47.1***

Classification Result

Model	Fine	Binary
(Kalchbrener et al. 2014)	48.5	86.9
(Kim 2014)	47.4	88.1
(Tai et al. 2015)	51.0	88.0
(Kumar et al. 2016)	52.1	88.6
Constant, scalar decay	52.7	88.6
Gated decay	52.9	89.2

Table 1: Results on Stanford Sentiment Treebank.

Analysis

Does it help to model non-consecutive patterns?