Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks

Hua He
University of Maryland College Park

Kevin Gimpel
Toyota Technological Institute at Chicago

Jimmy Lin
University of Waterloo

Problem: Sentence Similarity Measurement

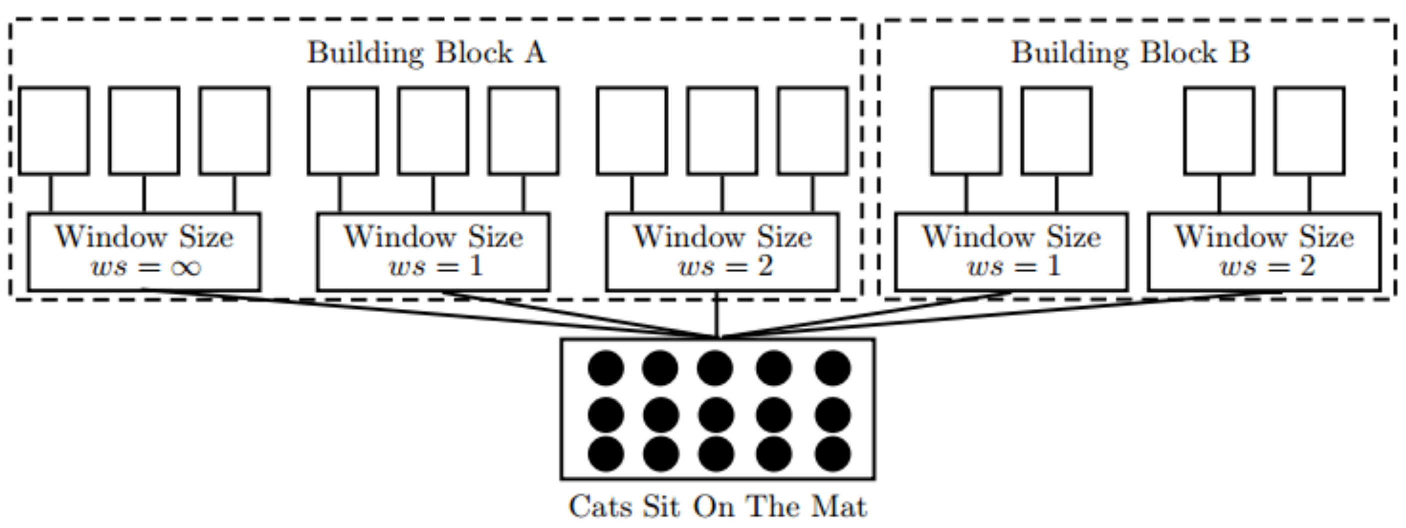
Given two sentences, measure their similarity:

The product also streams internet radio and comes with a 30-day free trial for realnetworks' rhapsody music subscription.
The device plays internet radio streams and comes with a 30-day trial of realnetworks rhapsody music service.

Approach: Multi-Perspective Sentence Representation and Structured Similarity Measurement

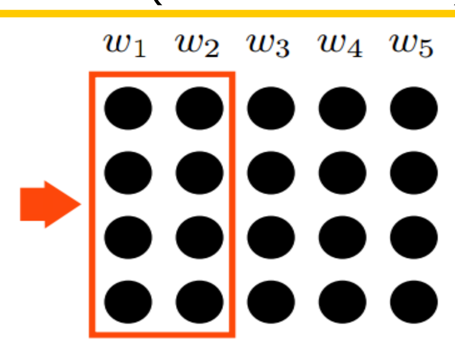
Part 1: Sentence Representation

to represent each sentence, we use **multiple types of convolution and pooling**

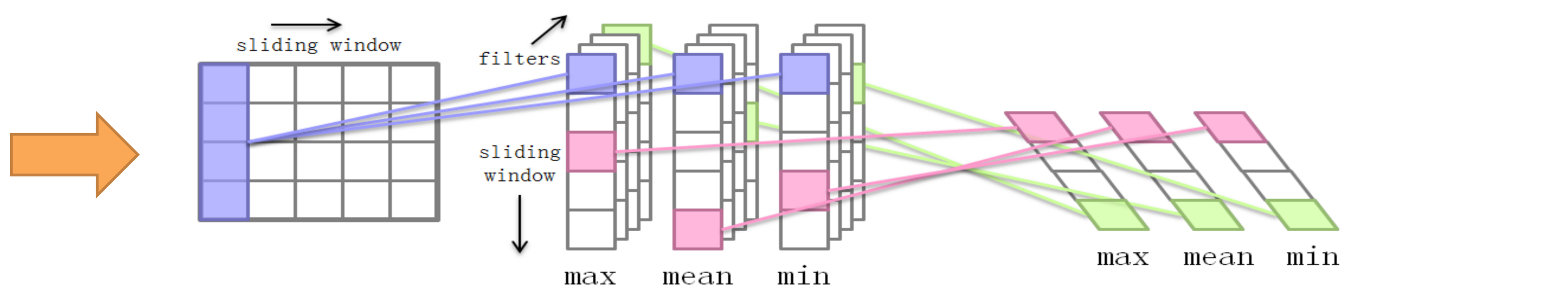


two types of convolution filters

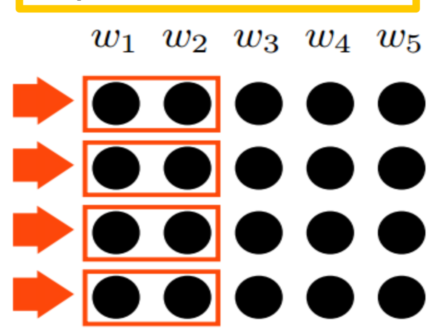
Building Block A: holistic (all dimensions)

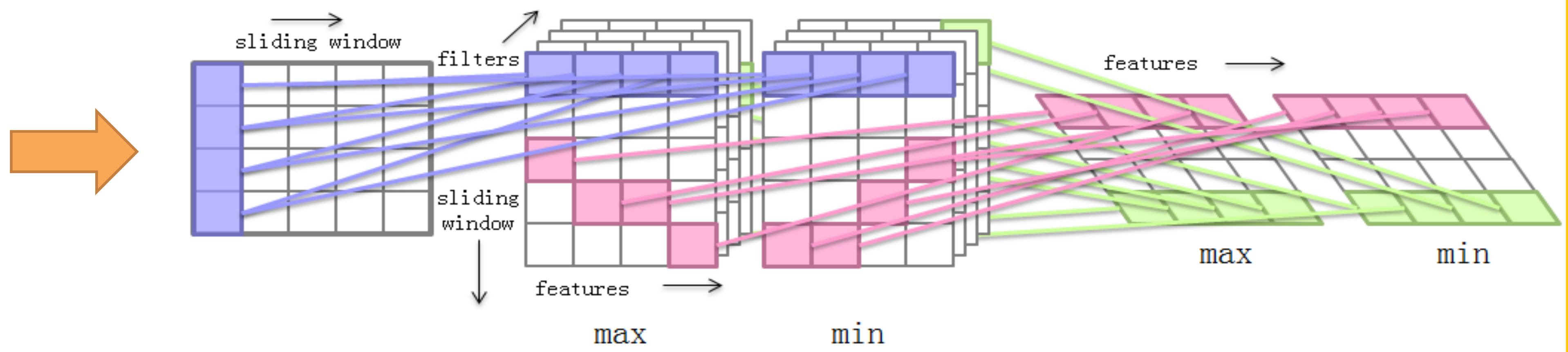


three types of pooling: max/min/mean
- each pooling group has **multiple window sizes** (1,2,3, ∞)
- each pooling group has independent underlying filters



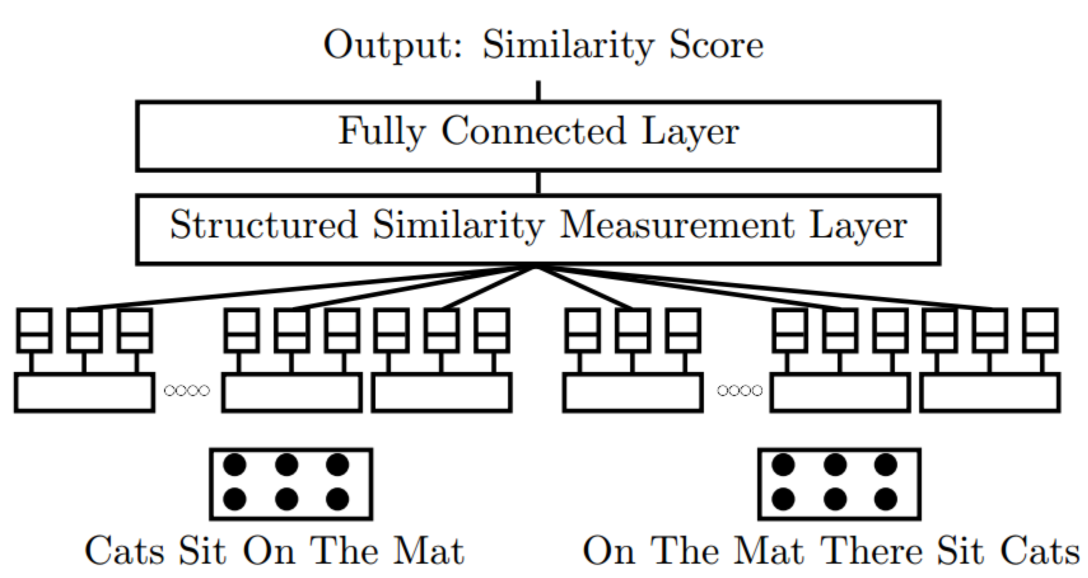
Building Block B: per-dimension





Part 2: Structured Similarity Measurement

sentence representations compared by **structured similarity measurement** layer



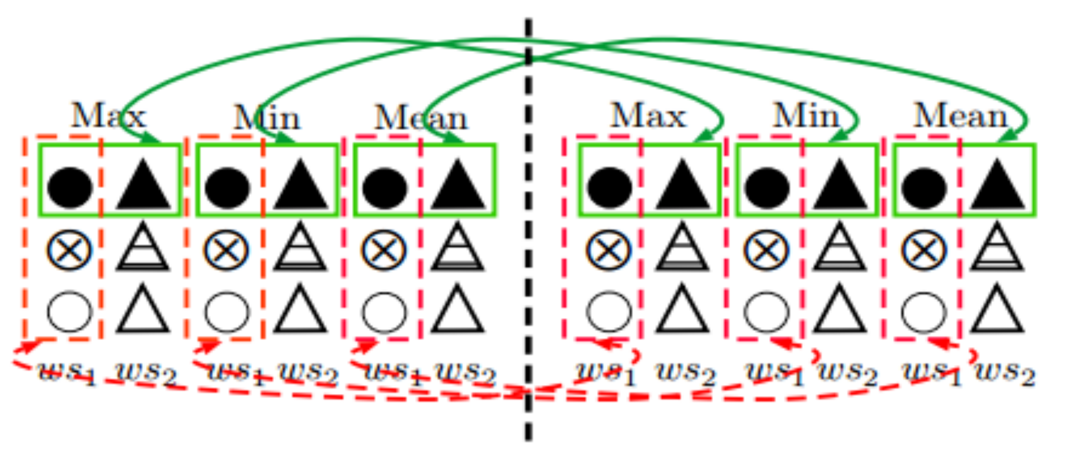
two algorithms **compare multiple pairs of local regions** of sentence representations

Algorithm 1 Horizontal Comparison

```
for each pooling p = max, min, mean do
    for each width ws_1 = 1...n, ∞ do
        regM_1[*][ws_1] = group_A(ws_1, p, S_1)
        regM_2[*][ws_1] = group_A(ws_1, p, S_2)
    end for
    for each i = 1...numFilter_A do
        fea_h = comU_2(regM_1[i], regM_2[i])
        accumulate fea_h for final layer
    end for
end for
```

Algorithm 2 Vertical Comparison

```
for each pooling p = max, min, mean do
    for each width ws_1 = 1...n, ∞ do
        oG_1A = group_A(ws_1, p, S_1)
        for each width ws_2 = 1...n, ∞ do
            oG_2A = group_A(ws_2, p, S_2)
            fea_a = comU_1(oG_1A, oG_2A)
            accumulate fea_a for final layer
        end for
    end for
    for each width ws_1 = 1...n do
        oG_1B = group_B(ws_1, p, S_1)
        oG_2B = group_B(ws_1, p, S_2)
        for each i = 1...numFilter_B do
            fea_b = comU_1(oG_1B[*][i], oG_2B[*][i])
            accumulate fea_b for final layer
        end for
    end for
end for
```



simplified example of local region comparison for two sentences (Block A with 3 filters)
- **green solid** lines: Horizontal Comparison (Alg. 1)
- **red dotted** lines: Vertical Comparison (Alg. 2)

each uses **multiple similarity metrics** for vector comparison

$$comU_1(\vec{x}, \vec{y}) = \{\cos(\vec{x}, \vec{y}), L_2Euclid(\vec{x}, \vec{y}), |\vec{x} - \vec{y}|\}$$
$$comU_2(\vec{x}, \vec{y}) = \{\cos(\vec{x}, \vec{y}), L_2Euclid(\vec{x}, \vec{y})\}$$

Experimental Results on Three Datasets

Experimental Setup

- Classification: Microsoft Research Paraphrase Corpus (MSRP)
- Similarity: Sentences Involving Compositional Knowledge (SICK)
- Similarity: Microsoft Video Paraphrase Corpus (MSRVID)
- multiple embeddings:
  - 300-dim GloVe (all tasks)
  - 200-dim POS (MSRP only)
  - 25-dim PARAGRAM (MSRP only)
- number of filters in Block A:
  - 525 (GloVe+POS+PARAGRAM) for MSRP
  - 300 for SICK/MSRVID
- embedding updating for MSRP only
- hinge loss for MSRP, KL-divergence loss (Tai et al., 2015) for SICK/MSRVID

MSRP

Model	Acc.	F1
Hu et al. (2014) ARC-I	69.6%	80.3%
Hu et al. (2014) ARC-II	69.9%	80.9%
Blacoe and Lapata (2012)	73.0%	82.3%
Fern and Stevenson (2008)	74.1%	82.4%
Finch (2005)	75.0%	82.7%
Das and Smith (2009)	76.1%	82.7%
Wan et al. (2006)	75.6%	83.0%
Socher et al. (2011)	76.8%	83.6%
Madnani et al. (2012)	77.4%	84.1%
Ji and Eisenstein (2013)	**80.41%**	**85.96%**
Yin and Schütze (2015) (without pretraining)	72.5%	81.4%
Yin and Schütze (2015) (with pretraining)	78.1%	84.4%
Yin and Schütze (2015) (pretraining+sparse features)	78.4%	84.6%
This work	**78.60%**	**84.73%**

SICK

Model	r	ρ	MSE
Socher et al. (2014) DT-RNN	0.7863	0.7305	0.3983
Socher et al. (2014) SDT-RNN	0.7886	0.7280	0.3859
Lai and Hockenmaier (2014)	0.7993	0.7538	0.3692
Jimenez et al. (2014)	0.8070	0.7489	0.3550
Bjerva et al. (2014)	0.8268	0.7721	0.3224
Zhao et al. (2014)	0.8414	-	-
LSTM	0.8477	0.7921	0.2949
Bi-LSTM	0.8522	0.7952	0.2850
2-layer LSTM	0.8411	0.7849	0.2980
2-layer Bidirectional LSTM	0.8488	0.7926	0.2893
Tai et al. (2015) Const. LSTM	0.8491	0.7873	0.2852
Tai et al. (2015) Dep. LSTM	0.8676	**0.8083**	**0.2532**
This work	**0.8686**	0.8047	0.2606

MSRVID

Model	Pearson's r
Rios et al. (2012)	0.7060
Wang and Cer (2012)	0.8037
Beltagy et al. (2014)	0.8300
Bär et al. (2012)	0.8730
Šarić et al. (2012)	0.8803
This work	**0.9090**

Ablation Study

Ablation Component	MSRP Accuracy Diff.	MSRVID Pearson Diff.	SICK Pearson Diff.
Remove POS embeddings	-0.81	NA	NA
Remove PARAGRAM embeddings	-1.33	NA	NA
Remove per-dimension embeddings, building block A only	-0.75	-0.0067	-0.0014
Remove min and mean pooling, use max pooling only	-0.58	-0.0112	+0.0001
Remove multiple widths, $ws = 1$ and $ws = \infty$ only	-2.14	-0.0048	-0.0012
Remove cosine and $L_2Euclid$ distance in $comU_*$	-2.31	-0.0188	-0.0309
Remove Horizontal Algorithm	-0.92	-0.0097	-0.0117
Remove Vertical Algorithm	-2.15	-0.0063	-0.0027
Remove similarity layer (completely flatten)	-1.90	-0.0121	-0.0288

Nine components in four groups:
(1) input embeddings
(2) sentence representation
(3) similarity measurement metrics
(4) similarity measurement layer

code available: hohocode.github.io/textSimilarityConvNet/

References

- *J. Wieting, M. Bansal, K. Gimpel, K. Livescu, and D. Roth. 2015. From paraphrase database to compositional paraphrase model and back. TACL.*
- *K. S. Tai, R. Socher, and C. D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. ACL.*
- *W. Yin and H. Schutze. 2015. Convolutional neural network for paraphrase identification. NAACL.*