Automatic Paraphrase Collection and Identification in Twitter

Wuwei Lan, Siyu Qiu, Hua He, Wei Xu
What is paraphrase?
What is paraphrase?

Willy Wonka was famous for his delicious candy. Children and adults loved to eat it.
What is paraphrase?

Willy Wonka was famous for his delicious candy. Children and adults loved to eat it.

Willy Wonka was known throughout the world because people enjoyed eating the tasty candy he made.
What is paraphrase?

Willy Wonka was **famous** for his delicious candy.
Children and adults loved to eat it.

Willy Wonka was **known throughout the world** because people enjoyed eating the tasty candy he made.
What is paraphrase?

Willy Wonka was famous for his delicious candy. Children and adults loved to eat it.

Willy Wonka was known throughout the world because people enjoyed eating the tasty candy he made.
What is paraphrase?

Willy Wonka was **famous** for his **delicious** candy. Children and adults **loved to eat it**.

Willy Wonka was **known throughout the world** because people **enjoyed eating the tasty** candy he made.
Paraphrase Application: duplicate question identification
Paraphrase Application: duplicate question identification
Paraphrase Application: duplicate question identification

[Search bar on Stack Overflow]

Search
python how to sort dictionary by value

search
Paraphrase Application: duplicate question identification

Q: Sort a Python dictionary by value

Q: How to sort a Python dictionary by value?

Q: Python how to sort a dictionary by value in reverse order
Paraphrase Application: duplicate question identification

Search

python how to sort dictionary by value

Q: Sort a Python dictionary by value

Q: How to sort a Python dictionary by value?

Q: Python how to sort a dictionary by value in reverse order
Paraphrase Application: duplicate question identification

Search

python how to sort dictionary by value

2477 votes  38 answers  Q: Sort a Python dictionary by value

12 votes  2 answers  Q: How to sort a Python dictionary by value?

-4 votes  3 answers  Q: Python how to sort a dictionary by value in reverse order
Paraphrase Application: duplicate question identification

**Q:** Sort a Python dictionary by value

2477 votes, 38 answers

**Q:** How to sort a Python dictionary by value?

12 votes, 2 answers

**Q:** Python how to sort a dictionary by value in reverse order

-4 votes, 3 answers
Paraphrase Application: question answering
Paraphrase Application: question answering
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India
<table>
<thead>
<tr>
<th>[Question]</th>
<th>[Supporting Evidence]</th>
</tr>
</thead>
<tbody>
<tr>
<td>In May 1898 Portugal celebrated the 400th anniversary of this explorer’s arrival in India</td>
<td>On the 27th of May 1498, Vasco da Gama landed in Kappad Beach</td>
</tr>
</tbody>
</table>
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India.

On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India.

On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India. On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer’s arrival in India.

On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India. On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India. On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
In May 1898 Portugal celebrated the 400th anniversary of this explorer’s arrival in India. On the 27th of May 1498, Vasco da Gama landed in Kappad Beach.
Paraphrases?

Paraphrases?


The New York Times @nytimes · 12 Oct 2016

Worries over the health of King Bhumibol Adulyadej are shaking Thailand

nyti.ms/2dRzPcr
Paraphrases?


The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPcr
Paraphrases?


The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand nyti.ms/2dRzPcr

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King's Health Shake Thailand ift.tt/2d7frGd
Paraphrases?

The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPcr

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King’s Health Shake Thailand ift.tt/2d7frGd

Paraphrases?

[Image of The New York Times article about Thailand's King Bhumibol Adulyadej]


The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPcr

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King’s Health Shake Thailand ift.tt/2d7frGd
Paraphrases?

- The New York Times via @nytimes · 12 Oct 2016
  Worries over the health of King Bhumibol Adulyadej are shaking Thailand
  nyti.ms/2dRzPcr

- Career Synchronicity via @careersync_now · 12 Oct 2016
  Fears for King’s Health Shake Thailand
  ift.tt/2d7frGd

- Herbert Buchsbaum via @herbertnyt · 12 Oct 2016
  New bulletin from Thai palace: King is still on a ventilator and in unstable
ccondition.  nyti.ms/2dW1A37
Paraphrases?

The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPc

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King’s Health Shake Thailand ift.tt/2d7frGd

Herbert Buchsbaum @herbertnyt · 12 Oct 2016
New bulletin from Thai palace: King is still on a ventilator and in unstable condition. nyti.ms/2dW1A37

Paraphrase

Non-Paraphrase
Paraphrases? We can get many in Twitter


The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPcr

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King’s Health Shake Thailand ift.tt/2d7frGd

Herbert Buchsbaum @herbertnyt · 12 Oct 2016
New bulletin from Thai palace: King is still on a ventilator and in unstable condition. nyti.ms/2dW1A37
Paraphrases? We can get many in Twitter


The New York Times @nytimes · 12 Oct 2016
Worries over the health of King Bhumibol Adulyadej are shaking Thailand
nyti.ms/2dRzPcr

Career Synchronicity @careersync_now · 12 Oct 2016
Fears for King’s Health Shake Thailand
ift.tt/2d7frGd

Herbert Buchsbaum @herbertnyt · 12 Oct 2016
New bulletin from Thai palace: King is still on a ventilator and in unstable condition.
nyti.ms/2dW1A37
Paraphrases? We can get many in Twitter


nyti.ms/2dRzPcr


fft.tt/2d7frGd

same URL

nyti.ms/2dW1A37
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

- [MSRP\textsuperscript{[1]}
  - clustered news articles
  - 5,801 annotated pairs

- [PIT-2015\textsuperscript{[2]}
  - Twitter trending topics
  - 14,035 annotated pairs

\textsuperscript{[1]} Dolan et al., 2004
\textsuperscript{[2]} Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:

- **[MSRP][1]**
  - clustered news articles
  - 5,801 annotated pairs

- **[PIT-2015][2]**
  - Twitter trending topics
  - 14,035 annotated pairs

[1] Dolan et al., 2004
[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space

[MSRP][1]
clustered news articles
5,801 annotated pairs

[PIT-2015][2]
Twitter trending topics
14,035 annotated pairs

[1] Dolan et al., 2004
[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space
• ensure diversity among sentences

[MSRP\textsuperscript{[1]}]
clustered news articles
5,801 annotated pairs

[PIT-2015\textsuperscript{[2]}]
Twitter trending topics
14,035 annotated pairs

[1] Dolan et al., 2004
[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

**Key for success:**
- narrow the search space
- ensure diversity among sentences

Also **Pitfalls** …

**[MSRP][1]**
- clustered news articles
- 5,801 annotated pairs

**[PIT-2015][2]**
- Twitter trending topics
- 14,035 annotated pairs

[1] Dolan et al., 2004
[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space
• ensure diversity among sentences

Also Pitfalls …

[MSRP\textsuperscript{1}]
clustered news articles
5,801 annotated pairs

needed a SVM classifier to select sentences before data annotation

[PIT-2015\textsuperscript{2}]
Twitter trending topics
14,035 annotated pairs

\textsuperscript{1} Dolan et al., 2004
\textsuperscript{2} Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space
• ensure diversity among sentences

Also Pitfalls …

[MSRP\textsuperscript{1}]
clustered news articles
5,801 annotated pairs

needed a SVM classifier to select sentences before data annotation

[1] Dolan et al., 2004

[PIT-2015\textsuperscript{2}]
Twitter trending topics
14,035 annotated pairs

needed human-in-the-loop to avoid “bad” topics

[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

-needed human-in-the-loop to avoid “bad” topics

[MSRP]

clustered news articles 5801 sentence pairs

[PIT-2015][2]

Twitter trending topics 14,035 annotated pairs
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success: 5801 sentence pairs needed human-in-the-loop to avoid "bad" topics

[MSRP]

[Twitter]

PIT-2015

Twitter trending topics

14,035 annotated pairs
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space

5801 sentence pairs needed human-in-the-loop to avoid “bad” topics

[Twitter trending topics 14,035 annotated pairs]
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

- Narrow the search space
- Ensure diversity among sentences

5801 sentence pairs needed human-in-the-loop to avoid "bad" topics

[PIT-2015[2]

Twitter trending topics

14,035 annotated pairs
Only exist two sentential paraphrase corpora
(which contain meaningful non-paraphrases)

Key for success:
• narrow the search space
• ensure diversity among sentences

Also Pitfalls …

5801 sentence pairs needed human-in-the-loop to avoid “bad” topics

Twitter trending topics
14,035 annotated pairs
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

[MSRP]

[1] clustered news articles needed a SVM classifier to select sentences before data annotation.

Key for success:

• narrow the search space
• ensure diversity among sentences

Also Pitfalls …

5801 sentence pairs needed human-in-the-loop to avoid “bad” topics.

[PIT-2015][2]

Twitter trending topics
14,035 annotated pairs
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
• narrow the search space
• ensure diversity among sentences

Also Pitfalls:

[MSRP][1]
clustered news articles
5,801 annotated pairs

[MSRP][1]
needed a SVM classifier to select sentences before data annotation

[PIT-2015][2]
Twitter trending topics
14,035 annotated pairs

[PIT-2015][2]
needed human-in-the-loop to avoid “bad” topics

[1] Dolan et al., 2004
[2] Xu et al., 2014
Only exist two sentential paraphrase corpora (which contain meaningful non-paraphrases)

Key for success:
- narrow the search space
- ensure diversity among sentences

Also Pitfalls: cause over-identification when applied to unlabeled data

[MSRP$^1$]
- clustered news articles
- 5,801 annotated pairs

[MSRP$^1$] needed a SVM classifier to select sentences before data annotation

[PIT-2015$^2$]
- Twitter trending topics
- 14,035 annotated pairs

[PIT-2015$^2$] needed human-in-the-loop to avoid “bad” topics

[1] Dolan et al., 2004
[2] Xu et al., 2014
We created the 3rd paraphrase corpora
(largest annotated corpus to date)

Key for success:
• narrow the search space
• ensure diversity among sentences
• the simpler the better!

[MSRP\textsuperscript{[1]}]
clustered news articles
5,801 annotated pairs

[Twitter URL Corpus]
URL-linked Tweets
51,524 annotated pairs

[PIT-2015\textsuperscript{[2]}]
Twitter trending topics
14,035 annotated pairs

\textsuperscript{[1]} Dolan et al., 2004
\textsuperscript{[2]} Xu et al., 2014

no clustering or topic detection needed
no data selection steps needed
We created the 3rd paraphrase corpora (largest annotated corpus to date)

**Key for success:**
- narrow the search space
- ensure diversity among sentences
- the simpler the better!

[MSRP]$^1$
- clustered news articles
- 5,801 annotated pairs

[Twitter URL Corpus]
- URL-linked Tweets
- 51,524 annotated pairs
- largest up-to-date

[PIT-2015]$^2$
- Twitter trending topics
- 14,035 annotated pairs

---

$^1$ Dolan et al., 2004
$^2$ Xu et al., 2014

no clustering or topic detection needed
no data selection steps needed
We created the 3rd paraphrase corpora (which also dynamically updates!)

Key for success:
• narrow the search space
• ensure diversity among sentences
• the simpler the better! more effective automatic paraphrase identification

[MSRP\textsuperscript{[1]}]
clustered news articles
5,801 annotated pairs

[Twitter URL Corpus]
URL-linked Tweets
51,524 annotated pairs

30,000 new sentential paraphrases every month

[PIT-2015\textsuperscript{[2]}]
Twitter trending topics
14,035 annotated pairs

\textsuperscript{[1]} Dolan et al., 2004
\textsuperscript{[2]} Xu et al., 2014
Once we have a lot of up-to-date sentential paraphrases (we can, for example, learn name variations fully automatically)
Once we have a lot of up-to-date sentential paraphrases (we can, for example, learn name variations fully automatically)

Donald Trump, DJT, Drumpf, Mr Trump, Idiot Trump, Chump, Evil Donald, #OrangeHitler, Donald @realTrump, D*nald Tr*mp, Comrade #Trump, Crooked #Trump, CryBaby Trump, Daffy Trump, Donald KKKrump, Dumb Trump, GOPTrump, Incompetent Trump, He-Who-Must-Not-Be-Named, President-elect Trump, President-Elect Trump, President-elect Donald J. Trump, PEOTUS Trump, Emperor Trump
Once we have a lot of up-to-date sentential paraphrases (we can, of course, learn other synonyms in large quantity via word alignment)

- FBI Director backs CIA finding
- FBI agrees with CIA
- FBI backs CIA view
- FBI finally backs CIA view
- FBI now backs CIA view
- FBI supports CIA assertion
- FBI Clapper back CIA's view
- The FBI backs the CIA’s assessment
- FBI Backs CIA …
How different from existing paraphrase corpora?

Model Performance → Dataset Difference
Automatic Paraphrase Identification
Automatic Paraphrase Identification

- **LEX-OrMF**[1] (Orthogonal Matrix Factorization[2])

---

[1] Xu et al., 2014
Automatic Paraphrase Identification

- **LEX-OrMF**[1] (Orthogonal Matrix Factorization[2])
- **DeepPairwiseWord**[3] (Deep Neural Networks)

[1] Xu et al., 2014
[3] He et al., 2016
Automatic Paraphrase Identification

- **LEX-OrMF**\(^1\) (Orthogonal Matrix Factorization\(^2\))
- **DeepPairwiseWord**\(^3\) (Deep Neural Networks)
- **MultiP**\(^4\) (Multiple Instance Learning)

\[^1\] Xu et al., 2014
\[^2\] Guo et al., 2014
\[^3\] He et al., 2016
\[^4\] Xu et al., 2014

\[
P(z_i, y_i|w_i; \theta) = \prod_{j=1}^{m} \exp(\theta \cdot f(z_j, w_j)) \times \sigma(z_i, y_i)
\]
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model

1. Context Modeling

Cats Sit On the Mat

On the Mat There Sit Cats
Deep Pairwise Word Model

Glove

1. Context Modeling

Cats Sit On the Mat
On the Mat There Sit Cats
Deep Pairwise Word Model

Bi-LSTM

Glove

Cats Sit On the Mat  On the Mat There Sit Cats

1. Context Modeling
Deep Pairwise Word Model

Decompose sentence input into word context to reduce modeling difficulty
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model

\[ coU(\vec{h}_1, \vec{h}_2) = \{ \cos(\vec{h}_1, \vec{h}_2), L_2 \text{Euclid}(\vec{h}_1, \vec{h}_2), \text{DotProduct}(\vec{h}_1, \vec{h}_2) \} \]
Deep Pairwise Word Model

Multiple vector similarity measurement used to capture word pair relationship

\[ coU(\vec{h}_1, \vec{h}_2) = \{\cos(\vec{h}_1, \vec{h}_2), L_2\text{Euclid}(\vec{h}_1, \vec{h}_2), DotProduct(\vec{h}_1, \vec{h}_2)\} \]
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model

More attention added to top ranked word pairs.
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model
Deep Pairwise Word Model

4. 19-Layer
Deep ConvNet
### Deep ConvNet Configurations

<table>
<thead>
<tr>
<th>Input Size: 32 by 32</th>
<th>Input Size: 48 by 48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Conv 128: size $3 \times 3$, stride 1, pad 1</td>
<td></td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Max Pooling: size $2 \times 2$, stride 2</td>
<td></td>
</tr>
<tr>
<td>Spatial Conv 164: size $3 \times 3$, stride 1, pad 1</td>
<td></td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Max Pooling: size $2 \times 2$, stride 2</td>
<td></td>
</tr>
<tr>
<td>Spatial Conv 192: size $3 \times 3$, stride 1, pad 1</td>
<td></td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Max Pooling: size $2 \times 2$, stride 2</td>
<td></td>
</tr>
<tr>
<td>Spatial Conv 128: size $3 \times 3$, stride 1, pad 1</td>
<td></td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Max Pooling: $2 \times 2$, s2</td>
<td>Max Pooling: $3 \times 3$, s1</td>
</tr>
<tr>
<td>Fully-Connected Layer</td>
<td></td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Fully-Connected Layer</td>
<td></td>
</tr>
<tr>
<td>LogSoftMax</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Deep ConvNet architecture given two padding size configurations for final classification.
Sentence pair relationship can be identified by pattern recognition through ConvNet.

Table 1: Deep ConvNet architecture given two padding size configurations for final classification.
Deep Pairwise Word Model

• From **Sentence Representation** to **Word Representation**
Deep Pairwise Word Model

- From Sentence Representation to Word Representation
- From Word Representation to Word Pair Interaction
Deep Pairwise Word Model

- From Sentence Representation to Word Representation
- From Word Representation to Word Pair Interaction
- From Normal Interaction to Attentive Interaction
Deep Pairwise Word Model

- From Sentence Representation to Word Representation
- From Word Representation to Word Pair Interaction
- From Normal Interaction to Attentive Interaction
- From Interaction to Pattern Recognition
Automatic Paraphrase Identification

Performance F1

- MSRP
- URL (this work)
- PIT-2015

Skewed over-identification in MSRP
Automatic Paraphrase Identification

- Performance F1: 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
- MSRP
- URL (this work)
- PIT-2015

- Random Baseline
- LR N-gram
- LEX-OrMF
- MultiP
- Deep PairwiseWord

Skewed over-identification in MSRP
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP

- MSRP
- URL (this work)
- PIT-2015

- Random Baseline
- LR N-gram
- LEX-OrMF
- MultiP
- Deep PairwiseWord
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP

- Random Baseline
- LR N-gram
- LEX-OrMF
- MultiP
- Deep PairwiseWord

- MSRP
- URL (this work)
- PIT-2015
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP
- easily setup & works well

- MSRP
- URL (this work)
- PIT-2015

Performance F1

- Random Baseline
- LR N-gram
- LEX-OrMF
- MultiP
- Deep PairwiseWord
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP
- easily setup & works well

Performance F1

- MSRP
- URL (this work)
- PIT-2015

Random Baseline LR N-gram LEX-OrMF MultiP Deep PairwiseWord
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

Performance F1

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP
- easily setup & works well

- best performance

- MSRP
- URL (this work)
- PIT-2015

Random Baseline
LR N-gram
LEX-OrMF
MultiP
Deep PairwiseWord
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP
- easily setup & works well

MSRP
URL (this work)
PIT-2015

Random Baseline
LR N-gram
LEX-OrMF
MultiP
Deep PairwiseWord

Performance F1
MSRP used a SVM classifier before data annotation.

- Skewed over-identification in MSRP
- Too much n-gram overlap in MSRP
- Easily setup & works well
- Best performance
- Best performance
- MSRP
- URL (this work)
- PIT-2015

Performance F1:
- Random Baseline
- LR N-gram
- LEX-OrMF
- MultiP
- Deep PairwiseWord

Performance F1 ranges from 0.3 to 0.9.
Automatic Paraphrase Identification

MSRP used a SVM classifier before data annotation

- skewed over-identification in MSRP
- too much n-gram overlap in MSRP
- easily setup & works well
- best performance
- too less n-gram overlap in PIT-2015

MSRP
URL (this work)
PIT-2015

PIT-2015 covered too broad content under the topic
System Performance v.s. Human Upper-bound
System Performance v.s. Human Upper-bound

Twitter URL Dataset

Performance F1

Amazon Mechanical Turk: 0.827
Deep PairWiseWord: 0.749
This newly discovered species of moth has been named after Donald Trump.

New #moth named in honor of Donald Trump @realDonaldTrump
This newly discovered species of moth has been named after Donald Trump.
This newly discovered species of moth has been named after Donald Trump.

New moth named in honor of Donald Trump @realDonaldTrump
Out-of-Vocabulary Word Problem
## Out-of-Vocabulary Word Problem

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training Size</th>
<th>Test Size</th>
<th># INV</th>
<th># OOV</th>
<th>OOV Ratio</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIT-2015</td>
<td>11530</td>
<td>838</td>
<td>7771</td>
<td>1238</td>
<td>13.7%</td>
<td>Twitter trends</td>
</tr>
<tr>
<td>Twitter-URL</td>
<td>42200</td>
<td>9324</td>
<td>24905</td>
<td>11440</td>
<td>31.5%</td>
<td>Twitter/news</td>
</tr>
<tr>
<td>MSRP</td>
<td>4076</td>
<td>1725</td>
<td>16226</td>
<td>1614</td>
<td>9.0%</td>
<td>news</td>
</tr>
</tbody>
</table>
# Out-of-Vocabulary Word Problem

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training Size</th>
<th>Test Size</th>
<th># INV</th>
<th># OOV</th>
<th>OOV Ratio</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIT-2015</td>
<td>11530</td>
<td>838</td>
<td>7771</td>
<td>1238</td>
<td>13.7%</td>
<td>Twitter trends</td>
</tr>
<tr>
<td>Twitter-URL</td>
<td>42200</td>
<td>9324</td>
<td>24905</td>
<td>11440</td>
<td>31.5%</td>
<td>Twitter/news</td>
</tr>
<tr>
<td>MSRP</td>
<td>4076</td>
<td>1725</td>
<td>16226</td>
<td>1614</td>
<td>9.0%</td>
<td>news</td>
</tr>
</tbody>
</table>
## Out-of-Vocabulary Word Problem

Randomly initialized word embeddings fail to capture word syntax and semantics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training Size</th>
<th>Test Size</th>
<th># INV</th>
<th># OOV</th>
<th>OOV Ratio</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIT-2015</td>
<td>11530</td>
<td>838</td>
<td>7771</td>
<td>1238</td>
<td>13.7%</td>
<td>Twitter trends</td>
</tr>
<tr>
<td>Twitter-URL</td>
<td>42200</td>
<td>9324</td>
<td>24905</td>
<td>11440</td>
<td>31.5%</td>
<td>Twitter/news</td>
</tr>
<tr>
<td>MSRP</td>
<td>4076</td>
<td>1725</td>
<td>16226</td>
<td>1614</td>
<td>9.0%</td>
<td>News</td>
</tr>
</tbody>
</table>
Representing Word with Smaller Units
## Representing Word with Smaller Units

<table>
<thead>
<tr>
<th>Unit</th>
<th>Output of $\sigma(\text{brexit})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>b, r, e, x, i, t</td>
</tr>
<tr>
<td>bigram w overlap</td>
<td>br, re, ex, xi, it</td>
</tr>
<tr>
<td>bigram w/o overlap</td>
<td>br, ex, it</td>
</tr>
<tr>
<td>trigram w overlap</td>
<td>bre, rex, exi, xit</td>
</tr>
<tr>
<td>trigram w/o overlap</td>
<td>bre, xit</td>
</tr>
<tr>
<td>whole word</td>
<td>brexit</td>
</tr>
</tbody>
</table>

Table 1: Ngram examples for word *brexit*. 

LSTM Based Character Embedding (C2W)[1]
LSTM Based Character Embedding (C2W)[1]

[1] Ling et al., 2015
LSTM Based Character Embedding (C2W) [1]

[1] Ling et al., 2015
CNN Based Character Embedding[1]
CNN Based Character Embedding

CNN Based Character Embedding [1]

Embedding Concatenation

CNN Based Character Embedding\[1\]

Embedding Concatenation

Convolution with multiple filters

\[
f^k[i] = \tanh(\langle C^k[*, i : i + w - 1], H \rangle + b)
\]

CNN Based Character Embedding \cite{kim2016}

Embedding Concatenation

Convolution with multiple filters

\[ f^k[i] = \tanh(\langle C^k[*, i : i + w - 1], H \rangle + b) \]

max pooling

\[ y^k = \max_i f^k[i] \]

\cite{kim2016} Kim et al., 2016
CNN Based Character Embedding \[1\]

Embedding Concatenation

Convolution with multiple filters

\[ f^k[i] = \tanh(C^k[*, i : i + w - 1], H) + b \]

max pooling

\[ y^k = \max_i f^k[i] \]

highway network

\[ t = \sigma(W_T y + b_T) \]
\[ z = t \odot g(W_H y + b_H) + (1 - t) \odot y \]

Subword Based Pairwise Word Interaction Model
Subword Based Pairwise Word Interaction Model
Word Embedding v.s. Subword Embedding
## Word Embedding v.s. Subword Embedding

<table>
<thead>
<tr>
<th>Model Variations</th>
<th>pre-train</th>
<th>#parameters</th>
<th>Twitter URL</th>
<th>PIT-2015</th>
<th>MSRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression (Lan et al., 2017)</td>
<td>Yes</td>
<td>9.5M</td>
<td>0.749</td>
<td>0.667</td>
<td>0.834</td>
</tr>
<tr>
<td>Word Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pretrained, fixed</td>
<td>Yes</td>
<td>2.2M</td>
<td>0.753</td>
<td>0.632</td>
<td>0.834</td>
</tr>
<tr>
<td>pretrained, updated</td>
<td>Yes</td>
<td>9.5M</td>
<td>0.756</td>
<td>0.656</td>
<td>0.832</td>
</tr>
<tr>
<td>randomized, fixed</td>
<td>–</td>
<td>2.2M</td>
<td>0.728</td>
<td>0.456</td>
<td>0.821</td>
</tr>
<tr>
<td>randomized, updated</td>
<td>–</td>
<td>9.5M</td>
<td>0.735</td>
<td>0.625</td>
<td>0.834</td>
</tr>
<tr>
<td>Subword Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2W, unigram</td>
<td>–</td>
<td>2.6M</td>
<td>0.742</td>
<td>0.534</td>
<td>0.816</td>
</tr>
<tr>
<td>C2W, bigram</td>
<td>–</td>
<td>2.7M</td>
<td>0.742</td>
<td>0.563</td>
<td>0.825</td>
</tr>
<tr>
<td>C2W, trigram</td>
<td>–</td>
<td>3.1M</td>
<td>0.729</td>
<td>0.576</td>
<td>0.824</td>
</tr>
<tr>
<td>CNN, unigram</td>
<td>–</td>
<td>6.5M</td>
<td>0.756</td>
<td>0.589</td>
<td>0.820</td>
</tr>
<tr>
<td>CNN, bigram</td>
<td>–</td>
<td>6.5M</td>
<td>0.760</td>
<td>0.646</td>
<td>0.814</td>
</tr>
<tr>
<td>CNN, trigram</td>
<td>–</td>
<td>6.7M</td>
<td>0.753</td>
<td>0.667</td>
<td>0.818</td>
</tr>
</tbody>
</table>
Multi-task Language Model
Multi-task Language Model
Multi-task Language Model

\[ E_{joint} = E + \gamma (\overrightarrow{E_{LM}} + \overleftarrow{E_{LM}}) \]
New State-of-the-art with Multi-task Language Model
# New State-of-the-art with Multi-task Language Model

<table>
<thead>
<tr>
<th>Model Variations</th>
<th>Pre-train</th>
<th>#Parameters</th>
<th>Twitter URL</th>
<th>PIT-2015</th>
<th>MSRP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression (Lan et al., 2017)</td>
<td>Yes</td>
<td>9.5M</td>
<td>0.683</td>
<td>0.749</td>
<td>0.645</td>
</tr>
<tr>
<td>pre-trained, fixed</td>
<td>Yes</td>
<td>2.2M</td>
<td>0.753</td>
<td>2.2M</td>
<td>0.632</td>
</tr>
<tr>
<td>pre-trained, updated</td>
<td>Yes</td>
<td>9.5M</td>
<td>0.756</td>
<td>0.656</td>
<td>0.832</td>
</tr>
<tr>
<td>randomized, fixed</td>
<td>–</td>
<td>2.2M</td>
<td>0.728</td>
<td>0.456</td>
<td>0.821</td>
</tr>
<tr>
<td>randomized, updated</td>
<td>–</td>
<td>9.5M</td>
<td>0.735</td>
<td>0.625</td>
<td>0.834</td>
</tr>
<tr>
<td><strong>Subword Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2W, unigram</td>
<td>–</td>
<td>2.6M</td>
<td>0.742</td>
<td>0.534</td>
<td>0.816</td>
</tr>
<tr>
<td>C2W, bigram</td>
<td>–</td>
<td>2.7M</td>
<td>0.742</td>
<td>0.563</td>
<td>0.825</td>
</tr>
<tr>
<td>C2W, trigram</td>
<td>–</td>
<td>3.1M</td>
<td>0.729</td>
<td>0.576</td>
<td>0.824</td>
</tr>
<tr>
<td>CNN, unigram</td>
<td>–</td>
<td>6.5M</td>
<td>0.756</td>
<td>0.589</td>
<td>0.820</td>
</tr>
<tr>
<td>CNN, bigram</td>
<td>–</td>
<td>6.5M</td>
<td>0.760</td>
<td>0.646</td>
<td>0.814</td>
</tr>
<tr>
<td>CNN, trigram</td>
<td>–</td>
<td>6.7M</td>
<td>0.753</td>
<td>0.667</td>
<td>0.818</td>
</tr>
<tr>
<td><strong>Subword+LM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM, C2W, unigram</td>
<td>–</td>
<td>3.5M</td>
<td>0.760</td>
<td><strong>0.691</strong></td>
<td>0.831</td>
</tr>
<tr>
<td>LM, C2W, bigram</td>
<td>–</td>
<td>3.6M</td>
<td><strong>0.768</strong></td>
<td>0.651</td>
<td>0.830</td>
</tr>
<tr>
<td>LM, C2W, trigram</td>
<td>–</td>
<td>4.0M</td>
<td>0.765</td>
<td>0.659</td>
<td>0.831</td>
</tr>
<tr>
<td>LM, CNN, unigram</td>
<td>–</td>
<td>7.4M</td>
<td>0.754</td>
<td>0.665</td>
<td><strong>0.840</strong></td>
</tr>
<tr>
<td>LM, CNN, bigram</td>
<td>–</td>
<td>7.4M</td>
<td>0.761</td>
<td>0.667</td>
<td>0.835</td>
</tr>
<tr>
<td>LM, CNN, trigram</td>
<td>–</td>
<td>7.6M</td>
<td>0.759</td>
<td>0.667</td>
<td>0.831</td>
</tr>
</tbody>
</table>
Takeaways

• Simple but effective paraphrase collection method
• Largest annotated paraphrase corpora to date
• Continuously growing, providing up-to-date data
• Subword embedding for paraphrase identification

• Data and Code: https://github.com/lanwuwei/paraphrase-dataset
Backup slides: Lexical Dissimilarity

% of Paraphrase pairs

PINC

MSRP
URL
PIT-2015

% of Paraphrase pairs

0 10 20 30 40

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Backup slides: Lexical Dissimilarity

% of Paraphrase pairs

PINC

MSRP
URL
PIT-2015

% of Paraphrase pairs

0 10 20 30 40

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0