Split and Rephrase: Better Evaluation and a Stronger Baseline

Roee Aharoni and Yoav Goldberg
NLP Lab, Bar Ilan University, Israel
ACL 2018
Motivation
Motivation

• Processing long, complex sentences is hard!
Motivation

- Processing long, complex sentences is hard!
- Children, people with reading disabilities, L2 learners…

*Simple English Wikipedia*

When writing articles here:

- Use **Basic English** vocabulary and **shorter sentences**. This allows people to understand normally complex terms or phrases.
Motivation

• Processing long, complex sentences is hard!

• Children, people with reading disabilities, L2 learners…

• Sentence level NLP systems:

  When writing articles here:
  • Use Basic English vocabulary and shorter sentences. This allows people to understand normally complex terms or phrases.
Motivation

- Processing long, complex sentences is hard!
- Children, people with reading disabilities, L2 learners...
- Sentence level NLP systems:
  - Dependency Parsers

McDonald & Nivre, 2011
Motivation

• Processing long, complex sentences is hard!

• Children, people with reading disabilities, L2 learners…

• Sentence level NLP systems:
  • Dependency Parsers
  • Neural Machine Translation

Koehn & Knowles, 2017
Motivation

• Processing long, complex sentences is hard!

• Children, people with reading disabilities, L2 learners…

• Sentence level NLP systems:
  • Dependency Parsers
  • Neural Machine Translation

• Can we automatically break a complex sentence into several simple ones while preserving its meaning?

Koehn & Knowles, 2017
The Split and Rephrase Task
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
- Dataset, evaluation method, baseline models
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
- Dataset, evaluation method, baseline models
- Task definition: complex sentence -> several simple sentences with the same meaning
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
- Dataset, evaluation method, baseline models
- Task definition: complex sentence -> several simple sentences with the same meaning

Alan Bean joined NASA in 1963 where he became a member of the Apollo 12 mission along with Alfred Worden as back up pilot and David Scott as commander.
The Split and Rephrase Task

• Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
• Dataset, evaluation method, baseline models
• Task definition: complex sentence -> several simple sentences with the same meaning

Alan Bean joined NASA in 1963 where he became a member of the Apollo 12 mission along with Alfred Worden as back up pilot and David Scott as commander.
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
- Dataset, evaluation method, baseline models
- Task definition: complex sentence -> several simple sentences with the same meaning

Alan Bean joined NASA in 1963 where he became a member of the Apollo 12 mission along with Alfred Worden as back up pilot and David Scott as commander.

Alan Bean served as a crew member of Apollo 12.
Alfred Worden was the backup pilot of Apollo 12.
Apollo 12 was commanded by David Scott.
Alan Bean was selected by Nasa in 1963.
The Split and Rephrase Task

- Narayan, Gardent, Cohen & Shimorina, EMNLP 2017
- Dataset, evaluation method, baseline models
- Task definition: complex sentence -> several simple sentences with the same meaning
- Requires (a) identifying independent semantic units (b) rephrasing those units to single sentences

Alan Bean served as a crew member of Apollo 12.
Alfred Worden was the backup pilot of Apollo 12.
Apollo 12 was commanded by David Scott.
Alan Bean was selected by NASA in 1963.

Alan Bean joined NASA in 1963 where he became a member of the Apollo 12 mission along with Alfred Worden as back up pilot and David Scott as commander.

Alan Bean served as a crew member of Apollo 12.
Alfred Worden was the backup pilot of Apollo 12.
Apollo 12 was commanded by David Scott.
Alan Bean was selected by NASA in 1963.
This Work
This Work

• We show that simple neural models seem to perform very on the original benchmark due to memorization of the training set
This Work

- We show that simple neural models seem to perform very on the original benchmark due to **memorization** of the training set.

- We propose a **more challenging data split** for the task to discourage memorization.
This Work

• We show that simple neural models seem to perform very on the original benchmark due to **memorization** of the training set

• We propose a **more challenging data split** for the task to discourage memorization

• We perform automatic evaluation and error analysis on the new benchmark, showing that the task is **still far from being solved**
WebSplit Dataset Construction
(Narayan et al. 2017)
WebSplit Dataset Construction
(Narayan et al. 2017)

Simple RDF Triples
(facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>
WebSplit Dataset Construction
(Narayan et al. 2017)

Simple RDF Triples (facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>

Simple Sentences

Alan Bean is a US national.

Alan Bean was on the crew of Apollo 12.

Alan Bean was hired by NASA in 1963.
WebSplit Dataset Construction
(Narayanan et al. 2017)

Simple RDF Triples
(facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>

Simple Sentences

Alan Bean is a US national.

Alan Bean was on the crew of Apollo 12.

Alan Bean was hired by NASA in 1963.

Sets of RDF triples

<Alan_Bean | nationality | United_States,
Alan_Bean | mission | Apollo_12,
Alan_Bean | NASA selection | 1963>
WebSplit Dataset Construction (Narayan et al. 2017)

Simple RDF Triples (facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>

Simple Sentences

Alan Bean is a US national.

Alan Bean was on the crew of Apollo 12.

Alan Bean was hired by NASA in 1963.

Sets of RDF triples

<Alan_Bean | nationality | United_States, Alan_Bean | mission | Apollo_12, Alan_Bean | NASA selection | 1963>

Complex Sentences

Alan Bean, born in the United States, was selected by NASA in 1963 and served as a crew member of Apollo 12.
WebSplit Dataset Construction  
(Narayan et al. 2017)

Simple RDF Triples  
(facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>

Simple Sentences

Alan Bean is a US national.

Alan Bean was on the crew of Apollo 12.

Alan Bean was hired by NASA in 1963.

Sets of RDF triples

<Alan_Bean | nationality | United_States,  
Alan_Bean | mission | Apollo_12,  
Alan_Bean | NASA selection | 1963>

Complex Sentences

Alan Bean, born in the United States, was selected by NASA in 1963 and served as a crew member of Apollo 12.
WebSplit Dataset Construction
(Narayan et al. 2017)

Simple RDF Triples
(facts from DBpedia)

<Alan_Bean | nationality | United_States>

<Alan_Bean | mission | Apollo_12>

<Alan_Bean | NASA selection | 1963>

Simple Sentences

Alan Bean is a US national.

Alan Bean was on the crew of Apollo 12.

Alan Bean was hired by NASA in 1963.

Matching via RDFs ~1M examples

Complex Sentences

Alan Bean, born in the United States, was selected by NASA in 1963 and served as a crew member of Apollo 12.
Preliminary Experiments
Preliminary Experiments

- ~1M training examples
Preliminary Experiments

- ~1M training examples
- “Vanilla” LSTM seq2seq with attention
Preliminary Experiments

- ~1M training examples
- “Vanilla” LSTM seq2seq with attention
- Shared vocabulary between the encoder and the decoder
Preliminary Experiments

- ~1M training examples
- “Vanilla” LSTM seq2seq with attention
- Shared vocabulary between the encoder and the decoder
- Simple sentences predicted as a single sequence
Preliminary Experiments

• ~1M training examples

• “Vanilla” LSTM seq2seq with attention

• Shared vocabulary between the encoder and the decoder

• Simple sentences predicted as a single sequence

• Evaluated using single-sentence, multi-reference BLEU as in Narayan et al. 2017
Preliminary Results
Preliminary Results

- Our simple seq2seq baseline outperform all but one of the baselines from Narayan et al. 2017
Preliminary Results

• Our simple seq2seq baseline outperforms all but one of the baselines from Narayan et al. 2017

• Their best baselines were using the RDF structures as additional information
Preliminary Results

- Our simple seq2seq baseline outperform all but one of the baselines from Narayan et al. 2017.
- Their best baselines were using the RDF structures as additional information.
- Do the simple seq2seq model really performs so well?

The bar chart shows the performance comparison between different models:
- **Text Only**: seq2seq (ours), seq2seq, split-multi
- **Text + RDFs**: hybrid, multi-seq2seq, split-seq2seq
BLEU can be Misleading
BLEU can be Misleading

• In spite of the high BLEU scores, our neural models suffer from:
BLEU can be Misleading

- In spite of the high BLEU scores, our neural models suffer from:
  - **Missing facts** - appeared in the input but not in the output

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>The address, 11 Diagonal Street is located in South Africa where the leader is Cyril Ramaphosa and some Asian South Africans live.</td>
<td>A Fortress of Grey Ice has 672 pages.</td>
</tr>
<tr>
<td></td>
<td>The address, 11 Diagonal Street is located in South Africa.</td>
</tr>
<tr>
<td></td>
<td>The leader of South Africa is called Cyril Ramaphosa.</td>
</tr>
<tr>
<td></td>
<td>The leader of South Africa is called Cyril Ramaphosa.</td>
</tr>
</tbody>
</table>
BLEU can be Misleading

- In spite of the high BLEU scores, our neural models suffer from:
  - **Missing facts** - appeared in the input but not in the output
  - **Unsupported facts** - appeared in the output but not in the input
BLEU can be Misleading

- In spite of the high BLEU scores, our neural models suffer from:
  - **Missing facts** - appeared in the input but not in the output
  - **Unsupported facts** - appeared in the output but not in the input
  - **Repeated facts** - appeared several times in the output
A Closer Look
A Closer Look

- Visualizing the attention weights we find an unexpected pattern
A Closer Look

- Visualizing the attention weights we find an unexpected pattern.
- The network mainly attends to a single token instead of spreading the attention.
A Closer Look

- Visualizing the attention weights we find an unexpected pattern
- The network mainly attends to a single token instead of spreading the attention
- This token was usually a part of the first mentioned entity
A Closer Look

- Visualizing the attention weights we find an unexpected pattern.

- The network mainly **attends to a single token** instead of spreading the attention.

- This token was usually a part of the **first mentioned entity**.

- Consistent among different input examples.
A Closer Look

- Visualizing the attention weights we find an unexpected pattern
- The network mainly attends to a single token instead of spreading the attention
- This token was usually a part of the first mentioned entity
- Consistent among different input examples
A Closer Look

- Visualizing the attention weights we find an unexpected pattern
- The network mainly attends to a single token instead of spreading the attention
- This token was usually a part of the first mentioned entity
- Consistent among different input examples
Testing for Over-Memorization
Testing for Over-Memorization

- In this stage we suspect that the network heavily *memorizes* entity-fact pairs
Testing for Over-Memorization

- In this stage we suspect that the network heavily **memorizes** entity-fact pairs
- We test this by introducing it with inputs consisting of repeated entities alone

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan Shepard Alan Shepard Alan Shepard</td>
<td></td>
</tr>
</tbody>
</table>
Testing for Over-Memorization

- In this stage we suspect that the network heavily **memorizes** entity-fact pairs.
- We test this by introducing it with inputs consisting of repeated entities alone.
- The network indeed generates facts it memorized about those specific entities.

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan Shepard Alan Shepard Alan Shepard</td>
<td>Alan Shepard is dead.</td>
</tr>
<tr>
<td></td>
<td>Alan Shepard was a test pilot.</td>
</tr>
</tbody>
</table>
Testing for Over-Memorization

- In this stage we suspect that the network heavily **memorizes** entity-fact pairs
- We test this by introducing it with inputs consisting of repeated entities alone
- The network indeed generates facts it memorized about those specific entities

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan Shepard Alan Shepard Alan Shepard</td>
<td>Alan Shepard is dead.</td>
</tr>
<tr>
<td></td>
<td>Alan Shepard was a test pilot.</td>
</tr>
<tr>
<td>AFC Ajax AFC Ajax AFC Ajax AFC Ajax</td>
<td>AFC Ajax ’s manager is Jong Ajax.</td>
</tr>
<tr>
<td></td>
<td>AFC Ajax N.V. own Sportpark De Toekomst.</td>
</tr>
</tbody>
</table>
Searching for the Cause: Dataset Artifacts
Searching for the Cause: Dataset Artifacts

- The original dataset included overlap between the training/development/test sets
Searching for the Cause: Dataset Artifacts

- The original dataset included overlap between the training/development/test sets
- When looking at the complex sentences side, there is no overlap
Searching for the Cause: Dataset Artifacts

- The original dataset included overlap between the training/development/test sets.
- When looking at the complex sentences side, there is no overlap.
- On the other hand, most of the simple sentences did overlap (~90%).
Searching for the Cause: Dataset Artifacts

- The original dataset included overlap between the training/development/test sets
- When looking at the complex sentences side, there is no overlap
- On the other hand, **most of the simple sentences** did overlap (~90%)
- Makes memorization very effective - “leakage” from train on the target side
New Data Split
New Data Split

• To remedy this, we construct a new data split by using the RDF information:
New Data Split

• To remedy this, we construct a new data split by using the RDF information:
  • Ensuring that all RDF relation types appear in the training set (enable generalization)
New Data Split

• To remedy this, we construct a new data split by using the RDF information:
  • Ensuring that all RDF relation types appear in the training set (enable generalization)
  • Ensuring that no RDF triple (fact) appears in two different sets (reduce memorization)
New Data Split

- To remedy this, we construct a new data split by using the RDF information:
  - Ensuring that all RDF relation types appear in the training set (enable generalization)
  - Ensuring that no RDF triple (fact) appears in two different sets (reduce memorization)
- The resulting dataset has no overlapping simple sentences

<table>
<thead>
<tr>
<th></th>
<th>Original Split</th>
<th>New Split</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique dev simple sentences in train</td>
<td>90.9%</td>
<td>0.09%</td>
</tr>
<tr>
<td>unique test simple sentences in train</td>
<td>89.8%</td>
<td>0%</td>
</tr>
</tbody>
</table>
New Data Split

- To remedy this, we construct a new data split by using the RDF information:
  - Ensuring that all RDF **relation types** appear in the training set (enable generalization)
  - Ensuring that no RDF triple (fact) appears in two different sets (reduce memorization)
- The resulting dataset has no overlapping simple sentences
- Has more unknown symbols in dev/test - **need better models!**

<table>
<thead>
<tr>
<th></th>
<th>Original Split</th>
<th>New Split</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique dev simple sentences in train</td>
<td>90.9%</td>
<td>0.09%</td>
</tr>
<tr>
<td>unique test simple sentences in train</td>
<td>89.8%</td>
<td>0%</td>
</tr>
<tr>
<td>% dev vocabulary in train</td>
<td>97.2%</td>
<td>63%</td>
</tr>
<tr>
<td>% test vocabulary in train</td>
<td>96.3%</td>
<td>61.7%</td>
</tr>
</tbody>
</table>
Copy Mechanism
Copy Mechanism

• To help with the increase in unknown words in the harder split, we incorporate a copy mechanism
Copy Mechanism

• To help with the increase in unknown words in the harder split, we incorporate a copy mechanism
  • Gu et al. 2016, See et al. 2017, Merity et al. 2017
Copy Mechanism

• To help with the increase in unknown words in the harder split, we incorporate a copy mechanism
  • Gu et al. 2016, See et al. 2017, Merity et al. 2017
• Uses a “copy switch” - feed-forward NN component with a sigmoid-activated scalar output
Copy Mechanism

- To help with the increase in unknown words in the harder split, we incorporate a copy mechanism
  - Gu et al. 2016, See et al. 2017, Merity et al. 2017
- Uses a “copy switch” - feed-forward NN component with a sigmoid-activated scalar output
- Controls the interpolation of the softmax probabilities and the copy probabilities over the input tokens in each decoder step

\[
p(w) = p(z = 1)p_{copy}(w) + p(z = 0)p_{softmax}(w)
\]
Results - New Split
Results - New Split

- Baseline seq2seq models completely break (BLEU < 7) on the new split
Results - New Split

- Baseline seq2seq models completely break (BLEU < 7) on the new split
- Copy mechanism helps to generalize
Results - New Split

• Baseline seq2seq models completely break (BLEU < 7) on the new split

• Copy mechanism helps to generalize

• Much lower than the original benchmark - memorization was crucial for the high BLEU
Copying and Attention
Copying and Attention

The copy-enhanced models spread the attention across the input tokens while improving results.
Error Analysis
Error Analysis

- On the original split the models did very well (due to memorization) with up to 91% correct simple sentences
Error Analysis

• On the original split the models did very well (due to memorization) with up to 91% correct simple sentences

• On the new benchmark the best model got only up to 20% correct simple sentences
Error Analysis

• On the original split the models did very well (due to memorization) with up to 91% correct simple sentences.

• On the new benchmark the best model got only up to 20% correct simple sentences.

• The task is much more challenging than previously demonstrated.
Conclusions
Conclusions

• Simple neural models seem to perform well due to memorization
Conclusions

• Simple neural models seem to perform well due to memorization

• We propose a more challenging data split for the task to discourage this
Conclusions

- Simple neural models seem to perform well due to **memorization**
- We propose a **more challenging data split** for the task to discourage this
  - A similar update was proposed by Narayan et al. in parallel to our work (*WebSplit v1.0*)
Conclusions

- Simple neural models seem to perform well due to **memorization**
- We propose a **more challenging data split** for the task to discourage this
  - A similar update was proposed by Narayan et al. in parallel to our work (**WebSplit v1.0**)
- We perform automatic evaluation and error analysis on the new benchmarks, showing that the task is **still far from being solved**
More Broadly
More Broadly

- Creating datasets is hard!
More Broadly

• Creating datasets is hard!
  • Think how models can “cheat"
More Broadly

• Creating datasets is hard!
  • Think how models can “cheat"
  • Create a challenging evaluation environment to capture generalization
More Broadly

- Creating datasets is hard!
  - Think how models can “cheat"
  - Create a challenging evaluation environment to capture generalization
  - Look for leakage of train to dev/test
More Broadly

• Creating datasets is hard!
  • Think how models can “cheat"
  • Create a challenging evaluation environment to capture generalization
  • Look for leakage of train to dev/test

• Numbers can be misleading!
More Broadly

• **Creating datasets is hard!**
  • Think how models can “cheat"
  • Create a challenging evaluation environment to capture generalization
  • Look for leakage of train to dev/test

• **Numbers can be misleading!**
  • Look at the data
More Broadly

- **Creating datasets is hard!**
  - Think how models can “cheat"
  - Create a challenging evaluation environment to capture generalization
  - Look for leakage of train to dev/test

- **Numbers can be misleading!**
  - Look at the data
  - Look at the model
More Broadly

• **Creating datasets is hard!**
  • Think how models can "cheat"
  • Create a challenging evaluation environment to capture generalization
  • Look for leakage of train to dev/test

• **Numbers can be misleading!**
  • Look at the data
  • Look at the model
  • Error analysis
Thank You!

Link to code and data is available in the paper :)