SPLIT AND REPHRASE

Shashi Narayan, Claire Gardent, Shay B. Cohen and Anastasia Shimorina
John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

Labour politician, John Clancy is the leader of Birmingham.

John Madin was born in Birmingham.

He was the architect of 103 Colmore Row.
John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

Labour politician, John Clancy is the leader of Birmingham. John Madin was born in Birmingham. He was the architect of 103 Colmore Row.

John Clancy is a labor politician who leads Birmingham. The architect of 103 Colmore Row was born here. His name was John Madin.
Our Contributions

**Split-and-Rephrase**: A *new* sentence rewriting task

<table>
<thead>
<tr>
<th></th>
<th>Split</th>
<th>Delete</th>
<th>Rephrase</th>
<th>Meaning-preserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split-and-Rephrase</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*A new benchmark* for this task

**Semantically-motivated split model** is a key factor in generating fluent and meaning preserving rephrasings
Split-and-Rephrase: Comparisons with Other Tasks

- Compression
- Paraphrasing
- Fusion
- Simplification
### Split-and-Rephrase: Comparisons with Other Tasks

<table>
<thead>
<tr>
<th></th>
<th>Compression</th>
<th>Paraphrasing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compression</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rephrase</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Meaning-preserve</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td><strong>Split-and-Rephrase</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(Knight and Marcu, 2000; Filippova and Strube, 2008; Cohn and Lapata, 2008; Pitler, 2010; Filippova et al, 2015)
### Split-and-Rephrase: Comparisons with Other Tasks

<table>
<thead>
<tr>
<th></th>
<th>Compression</th>
<th>Paraphrasing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fusion</strong></td>
<td>✗</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><strong>Split-and-Rephrase</strong></td>
<td>✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Split</th>
<th>Delete</th>
<th>Rephrase</th>
<th>Meaning-preserve</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fusion</strong></td>
<td>✗</td>
<td>often</td>
<td>✓</td>
<td>often</td>
</tr>
<tr>
<td><strong>Split-and-Rephrase</strong></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(McKeown et al., 2010; Filippova, 2010; Thadani and McKeown, 2013)
Split-and-Rephrase: Comparisons with Other Tasks

<table>
<thead>
<tr>
<th></th>
<th>Split</th>
<th>Delete</th>
<th>Rephrase</th>
<th>Meaning-preserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paraphrasing</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Split-and-Rephrase</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(Dras, 1999; Barzilay and McKeown, 2001; Bannard and Callison-Burch, 2005; Wubben et al., 2010; Mallinson et al., 2017)
## Split-and-Rephrase: Comparisons with Other Tasks

<table>
<thead>
<tr>
<th></th>
<th>Compression</th>
<th>Paraphrasing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Split</td>
<td>Delete</td>
</tr>
<tr>
<td>Simplification</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Split-and-Rephrase</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

_Fusion_  
(Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Wubben et al., 2012;)
(Siddharthan and Mandya, 2014; Narayan and Gardent, 2014, Xu et al., 2015; Zhang and Lapata, 2017)
Limitations of the Current Simplification Datasets

- Ill-suited for syntactic simplification related to splitting.
Split-and-Rephrase: Applications

- Shorter sentences are generally better processed by NLP systems (NLP applications).

- Reduced syntactic complexity will improve readability (Societal applications).
Split-and-Rephrase: Applications

- Shorter sentences are generally better processed by NLP systems (NLP applications).

- Reduced syntactic complexity will improve readability (Societal applications).

More beneficial than sentence simplification!
Split-and-Rephrase Benchmark
The Split-and-Rephrase Benchmark

Extracted from our large scale generation (WebNLG) corpus (Gardent et al., ACL 2017)

The WebNLG Corpus

RDF (Resource Description Framework) triple

\{ Birmingham | leaderName | John_Clancy_(Labour_politician) \}

Text  Labour politician, John Clancy is the leader of Birmingham.

Meaning representations (MRs, a set of RDF triples) paired with one or more texts verbalising those triples using crowdsourcing.
The Split-and-Rephrase Benchmark

Extracted from our large scale generation (WebNLG) corpus (Gardent et al., ACL 2017)

The WebNLG Corpus

RDF triples

\{John\_Madin | birthPlace | Birmingham, 103\_Colmore\_Row | architect | John\_Madin\}\n
Text-1  John Madin was born in Birmingham.

He was the architect of 103 Colmore Row.

Text-2  John Madin who was born in Birmingham, was the architect of 103 Colmore Row.
The Split-and-Rephrase Benchmark

Extracted from our large scale generation (WebNLG) corpus (Gardent et al., ACL 2017)

The WebNLG Corpus

- 13,308 MR-Text pairs, 7,049 distinct MRs, 8 DBpedia categories and 1-to-7 RDF triples in MRs.

The Split-and-Rephrase Benchmark

Extracted from our large scale generation (WebNLG) corpus (Gardent et al., ACL 2017)

The WebNLG Corpus

- 13,308 MR-Text pairs, 7,049 distinct MRs, 8 DBpedia categories and 1-to-7 RDF triples in MRs.

**Pivot approach:** Meaning representation (MR) as pivot for the extraction of paraphrases with splits.
MR

\{ Birmingham | leaderName | John_Clancy (Labour_politician),
   John_Madin | birthPlace | Birmingham,
   103_Colmore_Row | architect | John_Madin \}
Paraphrase Extraction with MRs as Pivot

MR

\{
Birmingham | leaderName | John_Clancy_(Labour_politician),

John_Madin | birthPlace | Birmingham,

103_Colmore_Row | architect | John_Madin \}

T-1  John Clancy is a labor politician who leads Birmingham, where architect
     John Madin, who designed 103 Colmore Row, was born.

T-2  Labour politician, John Clancy is the leader of Birmingham.
     John Madin was born in this city.
     He was the architect of 103 Colmore Row.
MR

\{ Birmingham | leaderName | John_Clancy_(Labour_politician),

John_Madin | birthPlace | Birmingham,

103_Colmore_Row | architect | John_Madin \}

T-1  John Clancy is a labor politician who leads Birmingham, where architect
John Madin, who designed 103 Colmore Row, was born.

T-2  Labour politician, John Clancy is the leader of Birmingham.
John Madin was born in this city.
He was the architect of 103 Colmore Row.
MR

\{ Birmingham | leaderName | John\_Clancy\_(Labour\_politician),  
    John\_Madin | birthPlace | Birmingham,  
    103\_Colmore\_Row | architect | John\_Madin \}

**T-1**  John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

**T-2**  Labour politician, John Clancy is the leader of Birmingham.
    John Madin was born in this city.
    He was the architect of 103 Colmore Row.

**S-1**  Labour politician, John Clancy is the leader of Birmingham.
MR

{ Birmingham | leaderName | John_Clancy_(Labour_politician),
  John_Madin | birthPlace | Birmingham,
  103_Colmore_Row | architect | John_Madin }

T-1  John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

T-2  Labour politician, John Clancy is the leader of Birmingham.
  John Madin was born in this city.
  He was the architect of 103 Colmore Row.

S-1  Labour politician, John Clancy is the leader of Birmingham.

S-2  John Madin was born in Birmingham.
  He was the architect of 103 Colmore Row.
Across Entries $\{ (MR, T-1), (MR-1, S-1) (MR-2, S-2) \}$

**T-1**  
*John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.*

**S-1**  
*Labour politician, John Clancy is the leader of Birmingham.*

**S-2**  
*John Madin was born in Birmingham.  
He was the architect of 103 Colmore Row.*
Across Entries  \{(MR,T-1), (MR-1, S-1) (MR-2, S-2)\}

T-1  John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

S-1  Labour politician, John Clancy is the leader of Birmingham.

S-2  John Madin was born in Birmingham.

He was the architect of 103 Colmore Row.

Within Entries \{(MR, T-1), (MR, T-2)\}

T-1  John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

T-2  Labour politician, John Clancy is the leader of Birmingham.

John Madin was born in this city.

He was the architect of 103 Colmore Row.
The Split-and-Rephrase Benchmark

- 1,100,166 pairs of the form
  \[ \{(M_C, C), \{(M_1, S_1) \ldots (M_n, S_n)\}\} \]
- 5,546 distinct complex sentences
- The vocabulary size is 3,311
The Split-and-Rephrase Benchmark

- 1,100,166 pairs of the form
  \{ (M_C, C), (M_1, S_1), \ldots, (M_n, S_n) \}\}
- 5,546 distinct complex sentences
- The vocabulary size is 3,311
- Number of sentences in the rephrasings varies between 2 and 7 with an average of 4.99
Split-and-Rephrase Models
Encoder-decoder Framework for NMT (SEQ2SEQ)

- Optimizes $p(S|C)$

(Sutskever et al., 2011; Bahdanau et al., 2014)
Multi-source NMT (MULTISEQ2SEQ)

\[ p(S|C) = \sum_{M_C} p(S|C; M_C)p(M_C|C) = p(S|C; M_C), \text{ if } M_C \text{ is known,} \]

where \( M_C \) is the meaning representation (RDF tuples) of \( C \).
John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

Inspired from ideas in

Hybrid Simplification using Deep Semantics and Machine Translation,
Shashi Narayan and Claire Gardent, ACL 2014.
John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

\[
\{ \text{Birmingham} | \text{leaderName} | \text{John Clancy (Labour politician)}, \\
\text{Birmingham} | \text{birthPlace} | \text{John Madin}, \\
\text{John Madin} | \text{architect} | \text{103 Colmore Row} \}
\]

Semantic Representation
Semantically-motivated Partition and Generate

\{
  Birmingham | leaderName | John_Clancy_(Labour_politician),
  Birmingham | birthPlace | John_Madin,
  John_Madin | architect | 103_Colmore_Row
\}

\{
  Birmingham | leaderName | John_Clancy_(Labour_politician)
\}

\{
  Birmingham | birthPlace | John_Madin,
  John_Madin | architect | 103_Colmore_Row
\}
Labour politician, John Clancy is the leader of Birmingham.

John Madin, the architect of 103 Colmore Row, was born in Birmingham.
John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

Labour politician, John Clancy is the leader of Birmingham.

John Madin, the architect of 103 Colmore Row, was born in Birmingham.
Semantically-motivated Partition and Generate

\[
p(S|C; M_C) = \sum_{M_{1:n}} p(S|C; M_C; M_{1:n}) \times p(M_{1:n}|C; M_C)
\]

Rephrase Partition

\(M_C\) is the meaning representation (RDF tuples) of \(C\).

\(M_{1:n} = M_1, \ldots, M_n\) is the partition of \(M_C\).
Semantically-motivated Partition and Generate

\[ p(S|C; M_C) = \sum_{M_{1:n}} p(S|C; M_C; M_{1:n}) \times p(M_{1:n}|C; M_C) \]

- A probabilistic model trained on the training set \( \{(M_C, C), \{(M_1, S_1) \ldots (M_n, S_n)\}\} \)
Semantically-motivated Partition and Generate

\[ p(S|C; M_C) = \sum_{M_{1:n}} p(S|C; M_C; M_{1:n}) \times p(M_{1:n}|C; M_C) \]

Learn to Rephrase \[ p(S|C; M_C; M_{1:n}) \]

\[ p(S|C; M_C; M_1, \ldots, M_n) \approx \prod_{i=1}^{n} p(S_i|C; M_i), \quad \text{(multi-seq2seq)} \]

\[ \approx \prod_{i=1}^{n} p(S_i|M_i), \quad \text{(seq2seq)} \]
• Training set (4,438, 80%), Validation set (554, 10%) and Test set (554, 10%)

• We evaluate on
  • **Meaning Preservation**: Multi-reference BLEU-4 scores
  • **Splits**:
    • #S/C: Average number of sentences in the output texts
    • #Tokens/S: Average number of tokens per output sentences
## RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>#S/C</th>
<th>#Tokens/S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT</strong></td>
<td>55.67</td>
<td>1.0</td>
<td>21.11</td>
</tr>
</tbody>
</table>

**INPUT** Alan Shepard was born in New Hampshire and he served as the Chief of the Astronaut Office.
### Results

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>#S/C</th>
<th>#Tokens/S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT</strong></td>
<td>55.67</td>
<td>1.0</td>
<td>21.11</td>
</tr>
<tr>
<td><strong>SEQ2SEQ</strong></td>
<td>48.92</td>
<td>2.51</td>
<td>10.32</td>
</tr>
<tr>
<td><strong>MULTISeq2Seq</strong></td>
<td>42.18</td>
<td>2.53</td>
<td>10.69</td>
</tr>
</tbody>
</table>

**INPUT**: Alan Shepard was born in New Hampshire and he served as the Chief of the Astronaut Office.

**SEQ2SEQ**: Alan Shepard’s occupation was a **test pilot**. Alan Shepard was born in New Hampshire. Alan Shepard was **born on Nov 18, 1923**.

**MULTISeq2Seq**: Alan Shepard served as a **test pilot**. Alan Shepard’s birth place was New Hampshire.
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>#S/C</th>
<th>#Tokens/S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT</strong></td>
<td>55.67</td>
<td>1.0</td>
<td>21.11</td>
</tr>
<tr>
<td><strong>Seq2Seq</strong></td>
<td>48.92</td>
<td>2.51</td>
<td>10.32</td>
</tr>
<tr>
<td><strong>MultiSeq2Seq</strong></td>
<td>42.18</td>
<td>2.53</td>
<td>10.69</td>
</tr>
<tr>
<td><strong>Split-Seq2Seq</strong></td>
<td>78.77</td>
<td>2.84</td>
<td>9.28</td>
</tr>
<tr>
<td><strong>Split-MultiSeq2Seq</strong></td>
<td>77.27</td>
<td>2.84</td>
<td>11.63</td>
</tr>
</tbody>
</table>

**INPUT**

Alan Shepard was born in New Hampshire and he served as the Chief of the Astronaut Office.

**Split-Seq2Seq**

Alan Shepard served as the Chief of the Astronaut Office. Alan Shepard’s birth place was New Hampshire.

**Split-MultiSeq2Seq**

Alan Shepard served as the Chief of the Astronaut Office. Alan Shepard was born in New Hampshire.
<table>
<thead>
<tr>
<th>Model</th>
<th>Task</th>
<th>Training Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SEQ2SEQ</strong></td>
<td>Given $C$, predict $S$</td>
<td>886,857</td>
</tr>
<tr>
<td><strong>MULTISEQ2SEQ</strong></td>
<td>Given $C$ and $M_C$, predict $S$</td>
<td>886,866</td>
</tr>
<tr>
<td><strong>SPLIT-MULTISEQ2SEQ</strong></td>
<td>Given $C$ and $M_C$, predict $M_1\ldots M_n$</td>
<td>13,051</td>
</tr>
<tr>
<td></td>
<td>Given $C$ and $S_i$, predict $S_i$</td>
<td>53,470</td>
</tr>
<tr>
<td><strong>SPLIT-SEQ2SEQ</strong></td>
<td>Given $C$ and $T_C$, predict $M_1\ldots M_n$</td>
<td>13,051</td>
</tr>
<tr>
<td></td>
<td>Given $M_i$, predict $T_i$</td>
<td>53,470</td>
</tr>
</tbody>
</table>
Future work

- Jointly learn to partition and rephrase

\[ p(S|C; M_C) = \sum_{M_{1:n}} p(S|C; M_C; M_{1:n}) \times p(M_{1:n}|C; M_C) \]

- Coverage based encoder-decoder models
Future work

- Jointly learn to partition and rephrase

\[ p(S|C; M_C) = \sum_{M_{1:n}} p(S|C; M_C; M_{1:n}) \times p(M_{1:n}|C; M_C) \]

- Coverage based encoder-decoder models

- Limitations of the Split-and-Rephrase benchmark
  - Notion of semantics simplifies with RDF triples: text is restricted to entity descriptions
  - Lexical diversity (portability to a new domain)
Where are we?
Conclusion

- We presented a new task for sentence splitting and rephrasing.
- Our experiments indicate that the semantically-motivated split model is a key factor in generating fluent and meaning preserving rephrasings.