COMP 786 (Fall 2020)
Natural Language Processing

Week 9: Summarization; Guest Talk; Machine Translation 1

Mohit Bansal

(various slides adapted/borrowed from courses by Dan Klein, JurafskyMartin-SLP3, Manning/Socher, others)
Automatic Document Summarization
Single-Document Summarization

Full document to a salient, non-redundant summary of ~100 words
Multi-Document Summarization

Several news sources with articles on the same topic (can use overlapping info across articles as a good feature for summarization)

... 27,000+ more
Extractive Summarization

- Directly selecting existing sentences from input document instead of rewriting them

\[
\begin{align*}
S_1 & : \text{The health care bill is a major test for the Obama administration.} \\
S_2 & : \text{Universal health care is a divisive issue.} \\
S_3 & : \text{President Obama remained calm.} \\
S_4 & : \text{Obama addressed the House on Tuesday.}
\end{align*}
\]
Graph-based Extractive Summ

Stationary distribution represents node centrality

Nodes are sentences

Edges are similarities

[Mihalcea et al., 2004, 2005; inter alia]
Maximize Concept Coverage

The health care bill is a major test for the Obama administration.

Universal health care is a divisive issue.

President Obama remained calm.

Obama addressed the House on Tuesday.

<table>
<thead>
<tr>
<th>concept</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>obama</td>
<td>3</td>
</tr>
<tr>
<td>health</td>
<td>2</td>
</tr>
<tr>
<td>house</td>
<td>1</td>
</tr>
</tbody>
</table>

Length limit: 18 words

<table>
<thead>
<tr>
<th>summary</th>
<th>length</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>{s_1, s_3}</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>{s_2, s_3, s_4}</td>
<td>17</td>
<td>6</td>
</tr>
</tbody>
</table>
Maximize Concept Coverage

- A set coverage optimization problem

\[
\max_{s \in S(D)} \sum_{c \in C(s)} v_c
\]

- Set of extractive summaries of document set D
- Set of concepts present in summary s
- Value of concept c

Results

<table>
<thead>
<tr>
<th></th>
<th>Bigram Recall</th>
<th>Pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
<td>2009</td>
</tr>
<tr>
<td>Baseline</td>
<td>4.00</td>
<td>6.85</td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Gillick and Favre, 2009]
Maximize Concept Coverage

- Can be solved using an integer linear program with constraints:

Maximize: \[ \sum_{i} w_i c_i \quad \text{total concept value} \]

Subject to: \[ \sum_{j} l_j s_j \leq L \quad \text{summary length limit} \]

\[ s_j \text{Occ}_{ij} \leq c_i, \quad \forall i, j \quad \text{maintain consistency between selected sentences and concepts} \]

\[ \sum_{j} s_j \text{Occ}_{ij} \geq c_i \quad \forall i \]

\[ c_i \in \{0, 1\} \quad \forall i \]

\[ s_j \in \{0, 1\} \quad \forall j \]

\( c_i \) an indicator for the presence of concept \( i \) in the summary, and \( s_j \) an indicator for the presence of sentence \( j \) in the summary. We add \( \text{Occ}_{ij} \) to indicate the occurrence of concept \( i \) in sentence \( j \). Equations (1) and (2) ensure the logical consistency of the solution: selecting a sentence necessitates selecting all the concepts it contains and selecting a concept is only possible if it is present in at least one selected sentence.

[Gillick et al., 2008] [Gillick and Favre, 2009]
Beyond Extraction: Compression

If you had to write a concise summary, making effective use of the 100-word limit, you would remove some information from the lengthy sentences in the original article.

What would a human do?

It is therefore unsurprising that Lindsay pleaded not guilty yesterday afternoon to the charges filed against her, according to her publicist.
Beyond Extraction: Compression

- Model should learn the subtree deletions/cuts that allow compression

```
yn1 = 1
yn2 = 1
yn3 = 0
yn4 = 0
yn5 = 1
yn6 = 1
yn7 = 1
yn8 = 1
```

```
She flew to France.
In France she remained.
```

```
c[ she flew ] = 0
```

```
c[ she’s in France ] = 1
```

```
c[ she remained ] = 1
```

[Berg-Kirkpatrick et al., 2011]
Beyond Extraction: Compression

- Model should learn the subtree deletions/cuts that allow compression

![Diagram](image)

She flew to France. In France she remained.

\[y_{n1} = 1\]
\[y_{n2} = 1\]
\[y_{n3} = 1\]
\[y_{n4} = 1\]
\[y_{n5} = 1\]
\[y_{n6} = 0\]
\[y_{n7} = 1\]
\[y_{n8} = 1\]

\[C[\text{she flew}] = 1\]
\[C[\text{she’s in France}] = 1\]
\[C[\text{she remained}] = 1\]
The new optimization problem looks to maximize the concept values as well as safe deletion values in the candidate summary:

\[
\max_{s \in S(D)} \left[ \sum_{c \in C(s)} v_c + \sum_{d \in D(s)} v_d \right]
\]

To decide the value/cost of a deletion, we decide relevant deletion features and the model learns their weights:

\[
v_d = w^\top f(d)
\]
### Beyond Extraction: Compression

Some example features for concept bigrams and cuts/deletions:

<table>
<thead>
<tr>
<th>Bigram Features $f(b)$</th>
<th>Cut Features $f(c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNT: Bucketed document counts</td>
<td>COORD: Coordinated phrase, four versions: NP, VP, S, SBAR</td>
</tr>
<tr>
<td>STOP: Stop word indicators</td>
<td>S-ADJUNCT: Adjunct to matrix verb, four versions: CC, PP, ADVP, SBAR</td>
</tr>
<tr>
<td>POSITION: First document position indicators</td>
<td>REL-C: Relative clause indicator</td>
</tr>
<tr>
<td>CONJ: All two- and three-way conjunctions of above</td>
<td>ATTR-C: Attribution clause indicator</td>
</tr>
<tr>
<td>BIAS: Always one</td>
<td>ATTR-PP: PP attribution indicator</td>
</tr>
<tr>
<td></td>
<td>TEMP-PP: Temporal PP indicator</td>
</tr>
<tr>
<td></td>
<td>TEMP-NP: Temporal NP indicator</td>
</tr>
<tr>
<td></td>
<td>BIAS: Always one</td>
</tr>
</tbody>
</table>

[ Berg-Kirkpatrick et al., 2011 ]
Neural Abstractive Summarization

- Mostly based on sequence-to-sequence RNN models
- Later added attention, coverage, pointer/copy, hierarchical encoder/attention, metric rewards RL, etc.
- Examples: Rush et al., 2015; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017
Feature-Augmented Encoder-Decoder

2 Models

In this section, we first describe the basic encoder-decoder RNN that serves as our baseline and then propose several novel models for summarization, each addressing a specific weakness in the baseline.

2.1 Encoder-Decoder RNN with Attention and Large Vocabulary Trick

Our baseline model corresponds to the neural machine translation model used in Bahdanau et al. (2014). The encoder consists of a bidirectional GRU-RNN (Chung et al., 2014), while the decoder consists of a uni-directional GRU-RNN with the same hidden-state size as that of the encoder, and an attention mechanism over the source-hidden states and a soft-max layer over target vocabulary to generate words. In the interest of space, we refer the reader to the original paper for a detailed treatment of this model. In addition to the basic model, we also adapted to the summarization problem, the large vocabulary 'trick' (LVT) described in Jean et al. (2014). In our approach, the decoder vocabulary of each mini-batch is restricted to words in the source documents of that batch. In addition, the most frequent words in the target dictionary are added until the vocabulary reaches a fixed size. The aim of this technique is to reduce the size of the soft-max layer of the decoder which is the main computational bottleneck. In addition, this technique also speeds up convergence by focusing the modeling effort only on the words that are essential to a given example. This technique is particularly well suited to summarization since a large proportion of the words in the summary come from the source document in any case.

2.2 Capturing Keywords using Feature-rich Encoder

In summarization, one of the key challenges is to identify the key concepts and key entities in the document, around which the story revolves. In order to accomplish this goal, we may need to go beyond the word-embeddings-based representation of the input document and capture additional linguistic features such as parts-of-speech tags, named-entity tags, and TF and IDF statistics of the words. We therefore create additional look-up based embedding matrices for the vocabulary of each tag-type, similar to the embeddings for words. For continuous features such as TF and IDF, we convert them into categorical values by discretizing them into a fixed number of bins, and use one-hot representations to indicate the bin number they fall into. This allows us to map them into an embeddings matrix like any other tag-type. Finally, for each word in the source document, we simply look-up its embeddings from all of its associated tags and concatenate them into a single long vector, as shown in Fig. 1. On the target side, we continue to use only word-based embeddings as the representation.

Figure 1: Feature-rich-encoder: We use one embedding vector each for POS, NER tags and discretized TF and IDF values, which are concatenated together with word-based embeddings as input to the encoder.

2.3 Modeling Rare/Unseen Words using Switching Generator-Pointer

Often-times in summarization, the keywords or named-entities in a test document that are central to the summary may actually be unseen or rare with respect to training data. Since the vocabulary of the decoder is fixed at training time, it cannot emit these unseen words. Instead, a most common way of handling these out-of-vocabulary (OOV) words is to emit an 'UNK' token as a placeholder. However, this does not result in legible summaries. In summarization, an intuitive way to handle such OOV words is to simply point to their location in the source document instead. We model this no-[Nallapati et al., 2016]
where conditional log-likelihood shown below, with ad-

At training time, we optimize the OOV word in summary occurs in multiple doc-

word does not exist in the target vocabulary. When

pointing to the probability of the

word-positions

the attention distribution

i

P

tions in the document as the distribution to sample

W

turning on at the

the

h

s

j

i

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

j

P

s

h

j

P

...
Hierarchical Attention

In Another Very Recent Work, Cheng And Lapata (2016) Have Illustrated Extractive Approaches, Consisting Of Identifying Keywords Extracted From The Article Onto The Compressed Output. Some Of The Other Notable Work In This Domain Consist Of Reproduction Of Original Sentences, As Well As Keywords Within Those Sentences.

2 Related Work

This Architecture Therefore Models Keyword Sentences As Well As Keywords Within Those Sentences. This Framework As An Attractive, Fully Data-Driven Alternative For Many NLP Tasks (Collobert et al., 2011), Researchers Have Started Considering This Mar. Approaches (Woodsend Et Al., 2010).

With The Emergence Of Deep Learning As A Viable Alternatives For Many NLP Tasks (Collobert Et Al., 2011), Researchers Have Started Considering This Mar. Approaches (Woodsend Et Al., 2010).

Hierarchical attention operates at both levels simultaneously. The attention at the sentence level, represented by the dotted arrows, is used to compute the attention at the word level, represented by the solid arrows. The word-level attention is further re-weighted by the sentence-level attention and rescaled output. The re-scaled output is then used to compute the attention at the word level, represented by the solid arrows. The sentence-level attention is then used to compute the attention at the sentence level, represented by the dotted arrows. The attention at the word level, represented by the solid arrows, is further re-weighted by the sentence-level attention and rescaled output. The re-scaled output is then used to compute the attention at the word level, represented by the solid arrows. The sentence-level attention is then used to compute the attention at the sentence level, represented by the dotted arrows. The attention at the word level, represented by the solid arrows, is further re-weighted by the sentence-level attention and rescaled output. The re-scaled output is then used to compute the attention at the word level, represented by the solid arrows. The sentence-level attention is then used to compute the attention at the sentence level, represented by the dotted arrows.

The data for these tasks consists of news stories from various domains. The task of abstractive summarization has been standardized using the DUC-2003 And DUC-2004 Competitions. In An Extension To This Work, Chopra Et Al. (2016) Used A Similar Convolutional Model For The Encoder, But Replaced The Attentional Feed-Forward Neural Network To Generate The Summary, Producing State-Of-The-Art Results On The DUC-2004 Task, Called TOPIARY (Zajic Et Al., 2004), Used A Combination Of Linguistically Motivated Compression Techniques, And An Unsupervised Topic Detection Algorithm That Appends Supervised Topics With Multiple Reference Summaries Per Story.

Humans On The Other Hand, Tend To Paraphrase Summaries. This Model Consist Of Reproduction Of Original Sentences, As Well As Keywords Within Those Sentences.

In Another Very Recent Work, Cheng And Lapata (2016) Have Illustrated Extractive Approaches, Consisting Of Identifying Keywords Extracted From The Article Onto The Compressed Output. Some Of The Other Notable Work In This Domain Consist Of Reproduction Of Original Sentences, As Well As Keywords Within Those Sentences.

2 Related Work

This Architecture Therefore Models Keyword Sentences As Well As Keywords Within Those Sentences. This Framework As An Attractive, Fully Data-Driven Alternative For Many NLP Tasks (Collobert Et Al., 2011), Researchers Have Started Considering This Mar. Approaches (Woodsend Et Al., 2010).

With The Emergence Of Deep Learning As A Viable Alternatives For Many NLP Tasks (Collobert Et Al., 2011), Researchers Have Started Considering This Mar. Approaches (Woodsend Et Al., 2010).
Pointer-Generator Networks

[See et al., 2017]
Pointer-Generator Networks


Best of both worlds: extraction + abstraction

[See et al., 2017]
Pointer-Generator Networks

[See et al., 2017]
Coverage for Redundancy Reduction

**Coverage** = cumulative attention = what has been covered so far

**Source Text:** Germany emerge victorious in 2-0 win against Argentina on Saturday

**Summary:** Germany beat _____

1. Use coverage as extra input to attention mechanism.
2. Penalize attending to things that have already been covered.

**Result:** repetition rate reduced to level similar to human summaries


[See et al., 2017]
Guest Talk by Ramakanth Pasunuru:

“Soft, Layer-Specific Multi-Task Summarization with Entailment and Question Generation” (ACL 2018)

“Multi-Reward Reinforced Summarization with Saliency and Entailment” (NAACL 2018)

(20 mins)
Topic:
Abstractive Summarization with Multi-Task Learning and Reinforcement Learning

(presented by Ramakanth Pasunuru)
Multi-Task Learning
Multi-Task Learning

- Multi-task Learning (MTL) is an inductive transfer mechanism which leverages information from related tasks to improve the primary model’s generalization performance.

- It achieves this goal by training multiple tasks in parallel while sharing representations, where the training signals from the auxiliary tasks can help improve the performance of the primary task.

[Caruana, 1998; Argyriou et al., 2007; Kumar and Daume, 2012; Luong et al., 2016]
Previous Work

German (translation)

Image (captioning)

English (unsupervised)

[Luong et al., 2016]

[Hashimoto et al., 2016]
MTL for Summarization
MTL for Summarization

- An accurate abstractive summary of a document should contain all its salient information and should be logically entailed by the input document.
MTL for Summarization

• An accurate abstractive summary of a document should contain all its salient information and should be logically entailed by the input document.

• We improve these via multi-task learning with auxiliary tasks of question generation and entailment generation.

• Question Generation teaches the summarization model how to look for salient questioning-worthy details.

• Entailment Generation teaches the model how to rewrite a summary which is a directed-logical subset of the input document.
In this paper we present an architecture that makes abstractive summarization viable (RNNs) both read and freely generate text, has been remarkably effective for eliminating repetition. Though these systems that were applied to short-text summarization. We propose a novel variant of the Forced-Attention Sentence Compression, which we use to track and control coverage of the source document. We show that coverage is re-

Our baseline model is similar to that of Vinyals et al. (2015); Rush et al. (2015); Hermann et al. (2015); Nallapati et al. (2016); Chopra et al. (2016). The attention distribution can be viewed as a balance between extrac-
tion and handling of OOV words, while retaining the ability to copy words from the source text via pointing.

Our hybrid pointer-generator network facilitates copying words from the source text to generate novel words, e.g., to produce the novel word ‘beat’ in the abstractive summary. The to-

2.1 Sequence-to-sequence attentional model

The code for our models is available online.

In this section we describe (1) our baseline sequence-to-sequence model, (2) our pointer-
generator model, and (3) our coverage mechanism that can be added to either of the first two models.

We apply our model to the recently-introduced CNN/Daily Mail dataset, which contains news articles (39%...). Though these systems...
Auxiliary Task: Question Generation

- The task of question generation is to generate a question from a given input sentence, which in turn is related to the skill of being able to find the important salient information to ask questions about the sentence.

- A good summary should also be able to find and extract all the salient information in the given source document, and hence we incorporate such capabilities into our abstractive text summarization model by multi-task learning it with a question generation task, sharing some common parameters/representations.

Sentence:
Oxygen is used in cellular respiration and released by photosynthesis, which uses the energy of sunlight to produce oxygen from water.

Questions:
- What life process produces oxygen in the presence of light? photosynthesis
- Photosynthesis uses which energy to form oxygen from water? sunlight
- From what does photosynthesis get oxygen? water
Auxiliary Task: Entailment Generation

- Directional, logical-implication relation between two sentences:
  - **Premise:** A girl is jumping on skateboard in the middle of a red bridge.
  - **Entailment:** The girl does a skateboarding trick.
  - **Contradiction:** The girl skates down the sidewalk.
  - **Neutral:** The girl is wearing safety equipment.

- **Premise:** A blond woman is drinking from a public fountain.
  - **Entailment:** The woman is drinking water.
  - **Contradiction:** The woman is drinking coffee.
  - **Neutral:** The woman is very thirsty.
Auxiliary Task: Entailment Generation

- Directional, logical-implication relation between two sentences:
  - **Premise:** A girl is jumping on skateboard in the middle of a red bridge.
  - **Entailment:** The girl does a skateboarding trick.
  - **Contradiction:** The girl skates down the sidewalk.
  - **Neutral:** The girl is wearing safety equipment.

  - **Premise:** A blond woman is drinking from a public fountain.
  - **Entailment:** The woman is drinking water.
  - **Contradiction:** The woman is drinking coffee.
  - **Neutral:** The woman is very thirsty.

- The task of entailment generation is to generate a hypothesis which is entailed by (or logically follows from) the given premise as input.

- In summarization, the generation decoder also needs to generate a summary that is entailed by the source document, i.e., does not contain any contradictory or unrelated/extraneous information as compared to the input document.
MTL Architecture

• QG stands for Question Generation
• SG stands for Summary Generation
• EG stands for Entailment Generation

(slides by Ramakanth Pasunuru)

(Pasunuru, Guo, & Bansal, ACL 2018)
Belinkov et al. (2017) observed that lower layers of RNN cells in a seq2seq machine translation model learn to represent word structure, while higher layers are more focused on high-level semantic meanings.

We believe that these tasks have different training data distributions and low-level representations, they can still benefit from sharing their models’ high-level components.

Thus, we keep the lower-level layer of the 2-layer encoder/decoder of all three tasks unshared, while we share the higher layer across the three tasks.
Soft vs. Hard Parameter Sharing

- Hard-sharing: In the most common multi-task learning hard-sharing approach, the parameters to be shared are forced to be the same. As a result, gradient information from multiple tasks will directly pass through shared parameters, hence forcing a common space representation for all the related tasks.

- Soft-sharing: We encourage shared parameters to be close in representation space by penalizing their L2 distances. Unlike hard sharing, this approach gives more flexibility for the tasks by only loosely coupling the shared space representations.
Results

<table>
<thead>
<tr>
<th>Models</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PREVIOUS WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq (50k vocab) (See et al., 2017)</td>
<td>31.33</td>
<td>11.81</td>
<td>28.83</td>
<td>12.03</td>
</tr>
<tr>
<td>Pointer (See et al., 2017)</td>
<td>36.44</td>
<td>15.66</td>
<td>33.42</td>
<td>15.35</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) *</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) †</td>
<td>38.82</td>
<td>16.81</td>
<td>35.71</td>
<td>18.14</td>
</tr>
<tr>
<td><strong>OUR MODELS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Layer Baseline (Pointer+Coverage) ⊙</td>
<td>39.56</td>
<td>17.52</td>
<td>36.36</td>
<td>18.17</td>
</tr>
<tr>
<td>⊙ + Entailment Generation</td>
<td>39.84</td>
<td>17.63</td>
<td>36.54</td>
<td>18.61</td>
</tr>
<tr>
<td>⊙ + Question Generation</td>
<td>39.73</td>
<td>17.59</td>
<td>36.48</td>
<td>18.33</td>
</tr>
<tr>
<td>⊙ + Entailment Gen. + Question Gen.</td>
<td>39.81</td>
<td>17.64</td>
<td>36.54</td>
<td>18.54</td>
</tr>
</tbody>
</table>

*Our multi-task model is stat. signif. better than baseline (based on bootstrap test with 100K samples: Efron and Tibshirani, 1994).

**Table:** Performance of our multi-task models on CNN/DailyMail dataset (~300K examples).
Results

<table>
<thead>
<tr>
<th>Models</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PREVIOUS WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq(50k vocab) (See et al., 2017)</td>
<td>31.33</td>
<td>11.81</td>
<td>28.83</td>
<td>12.03</td>
</tr>
<tr>
<td>Pointer (See et al., 2017)</td>
<td>36.44</td>
<td>15.66</td>
<td>33.42</td>
<td>15.35</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) *</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) †</td>
<td>38.82</td>
<td>16.81</td>
<td>35.71</td>
<td>18.14</td>
</tr>
<tr>
<td><strong>OUR MODELS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Layer Baseline (Pointer+Coverage) ⊘</td>
<td>39.56</td>
<td>17.52</td>
<td>36.36</td>
<td>18.17</td>
</tr>
<tr>
<td>⊘ + Entailment Generation</td>
<td>39.84</td>
<td>17.63</td>
<td>36.54</td>
<td>18.61</td>
</tr>
<tr>
<td>⊘ + Question Generation</td>
<td>39.73</td>
<td>17.59</td>
<td>36.48</td>
<td>18.33</td>
</tr>
<tr>
<td>⊘ + Entailment Gen. + Question Gen.</td>
<td>39.81</td>
<td>17.64</td>
<td>36.54</td>
<td>18.54</td>
</tr>
</tbody>
</table>

**Table:** Performance of our multi-task models on CNN/DailyMail dataset (~300K examples).

**Human evaluation:** Multi-task model is better than baseline

* Our multi-task model is stat. signif. better than baseline (based on bootstrap test with 100K samples: Efron and Tibshirani, 1994).
Results

<table>
<thead>
<tr>
<th>Models</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PREVIOUS WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq(50k vocab) (See et al., 2017)</td>
<td>31.33</td>
<td>11.81</td>
<td>28.83</td>
<td>12.03</td>
</tr>
<tr>
<td>Pointer (See et al., 2017)</td>
<td>36.44</td>
<td>15.66</td>
<td>33.42</td>
<td>15.35</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) *</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Pointer+Coverage (See et al., 2017) †</td>
<td>38.82</td>
<td>16.81</td>
<td>35.71</td>
<td>18.14</td>
</tr>
<tr>
<td><strong>OUR MODELS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Layer Baseline (Pointer+Coverage) ⊗</td>
<td>39.56</td>
<td>17.52</td>
<td>36.36</td>
<td>18.17</td>
</tr>
<tr>
<td>⊗ + Entailment Generation</td>
<td>39.84</td>
<td>17.63</td>
<td>36.54</td>
<td>18.61</td>
</tr>
<tr>
<td>⊗ + Question Generation</td>
<td>39.73</td>
<td>17.59</td>
<td>36.48</td>
<td>18.33</td>
</tr>
<tr>
<td>⊗ + Entailment Gen. + Question Gen.</td>
<td>39.81</td>
<td>17.64</td>
<td>36.54</td>
<td>18.54</td>
</tr>
</tbody>
</table>

**Table:** Performance of our multi-task models on CNN/DailyMail dataset (~300K examples).

**Human evaluation:** Multi-task model is better than baseline

<table>
<thead>
<tr>
<th>Models</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>See et al. (2017)</td>
<td>34.30</td>
<td>14.25</td>
<td>30.82</td>
</tr>
<tr>
<td>Baseline</td>
<td>35.96</td>
<td>15.91</td>
<td>32.92</td>
</tr>
<tr>
<td>Multi-Task (EG + QG)</td>
<td>36.73</td>
<td>16.15</td>
<td>33.58</td>
</tr>
</tbody>
</table>

**Table:** Performance of various models on DUC 2002 test only setup (567 examples).

* Our multi-task model is stat. signif. better than baseline (based on bootstrap test with 100K samples: Efron and Tibshirani, 1994).
Ground-truth: Celtic were defeated 3-2 after extra-time in the Scottish Cup semi-final. Leigh Griffiths had a goal-bound shot blocked by a clear handball. However, no action was taken against offender Josh Meekings. The Hoops have written the SFA for an ‘understanding’ of the decision.

See et al. (2017): John Hartson was once on the end of a major Hampden injustice while playing for Celtic. But he cannot see any point in his old club writing to the Scottish Football Association over the latest controversy at the national stadium. Hartson had a goal wrongly disallowed for offside while Celtic were leading 1-0 at the time but went on to lose 3-2.

Our Baseline: John Hartson scored the late winner in 3-2 win against Celtic. Celtic were leading 1-0 at the time but went on to lose 3-2. Some fans have questioned how referee Steven McLean and additional assistant Alan Muir could have missed the infringement.

Multi-task: Celtic have written to the Scottish Football Association in order to gain an ‘understanding’ of the refereeing decisions. The Hoops were left outraged by referee Steven McLean’s failure to award a penalty or red card for a clear handball in the box by Josh Meekings. Celtic striker Leigh Griffiths has a goal-bound shot blocked by the outstretched arm of Josh Meekings.
Reinforcement Learning
Reinforcement Learning

- Reinforcement Learning (RL) is a training mechanism in which an agent or a policy is allowed to interact with a given environment in order to maximize a reward.

- RL has successful application to many research areas such as continuous control, dialogue systems, and games.

- Recently, a special case of RL called policy gradients based reinforcement learning, has been widely applied to text generation problems in NLP through REINFORCE algorithm.

[Williams, 1992; Tesauro, 1995; White & Sofge, 1992; Singh et al., 2002; Ren et al., 2017; Rennie et al., 2017; Paulus et al., 2018; Chen & Bansal, 2018; Celikyilmaz et al., 2018]
REINFORCE

Figure: Overview of an LSTM decoder with sampling of words in a sequential fashion to generate a sentence. We measure a reward for the generated sentence w.r.t. the ground-truth and use this reward to update RL policy (model).
REINFORCE

Loss function:
\[ L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [r(w^s)] \]

Gradient estimation:
\[ \nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [r(w^s) \nabla_\theta \log p_\theta(w^s)] . \]

Gradient approximation:
\[ \nabla_\theta L(\theta) \approx -r(w^s) \nabla_\theta \log p_\theta(w^s) . \]

Reducing variance
\[ \nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [(r(w^s) - b) \nabla_\theta \log p_\theta(w^s)] . \]

Mixed loss:
\[ L_{\text{mixed}} = (1 - \gamma) L_{\text{XE}} + \gamma L_{\text{RL}} \]

Figure: Overview of an LSTM decoder with sampling of words in a sequential fashion to generate a sentence. We measure a reward for the generated sentence w.r.t. the ground-truth and use this reward to update RL policy (model).
We address *three* important aspects (saliency, directed logical entailment/correctness, and non-redundancy) of a good abstractive text summary via reinforcement learning approach with *two novel reward functions*. 
We address three important aspects (saliency, directed logical entailment/correctness, and non-redundancy) of a good abstractive text summary via reinforcement learning approach with two novel reward functions.

We also introduce a novel and effective multi-reward approach of optimizing multiple rewards simultaneously in alternate multi-task mini-batches.
Reward Functions

Based on the primary summarization metric of ROUGE package (Lin, 2004).

Rouge Reward
Reward Functions

- **Rouge Reward**: Based on the primary summarization metric of ROUGE package (Lin, 2004).
- **Saliency Reward**: Gives *higher* weight to the important, salient words/phrases when calculating the ROUGE score.
- **Entailment Reward**: Based on whether each sentence of the generated summary is entailed by the ground-truth summary.
ROUGESal reward gives higher weight to the important, salient words/phrases when calculating the ROUGE score (which by default assumes all words are equally weighted):

- To learn these saliency weights, we train our saliency predictor on \{sentence, answer spans\} pairs from the popular SQuAD reading comprehension dataset (Rajpurkar et al., 2016) (Wiki domain).

- We treat the human-annotated answer spans for important questions as representative salient information in the document.

- This saliency predictor is run on the ground-truth summary to get an importance weight for each word (used in ROUGE matching).

**Figure**: Overview of our saliency prediction model.
Entailment Reward: Entail

- A good summary should be logically entailed by source document, i.e., have no contradictory/unrelated information. We use an entailment scorer and its multi-sentence, length-normalized extension (to avoid very short sentences achieving misleadingly high entailment scores) as our “Entail” reward.

- We train the entailment classifier (Parikh et al., 2016) on the SNLI and Multi-NLI datasets and calculate the entailment probability score between the ground-truth (GT) summary (as premise) and each sentence of the generated summary (as hypothesis), and use average score as our Entail reward.

\[
\text{Entail} = \text{Entail} \times \frac{\text{#tokens in generated summary}}{\text{#tokens in reference summary}}
\]
Multi-Reward Optimization

- One approach for multi-reward optimization is to use a weighted combination of the rewards, but this has the issue of finding the complex scaling and weight balance among these diverse reward combinations.
Multi-Reward Optimization

• One approach for multi-reward optimization is to use a weighted combination of the rewards, but this has the issue of finding the complex scaling and weight balance among these diverse reward combinations.

• To address this issue, we instead introduce a simple multi-reward optimization approach inspired from multi-task learning, where we have different tasks, and they share all model parameters while having their own optimization function (different reward functions in this case), with alternate mini-batches:

$$L_{RL_1} = -(r_1(w^s) - r_1(w^a)) \nabla_\theta \log p_\theta(w^s)$$

$$L_{RL_2} = -(r_2(w^s) - r_2(w^a)) \nabla_\theta \log p_\theta(w^s)$$


## Results (CNN/Daily Mail)

<table>
<thead>
<tr>
<th>Models</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PREVIOUS WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nallapati (2016)*</td>
<td>35.46</td>
<td>13.30</td>
<td>32.65</td>
<td>-</td>
</tr>
<tr>
<td>See et al. (2017)</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Paulus (2017) (XE)*</td>
<td>38.30</td>
<td>14.81</td>
<td>35.49</td>
<td>-</td>
</tr>
<tr>
<td>Paulus (2017) (RL)*</td>
<td>39.87</td>
<td>15.82</td>
<td>36.90</td>
<td>-</td>
</tr>
<tr>
<td><strong>OUR MODELS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline (XE)</td>
<td>39.41</td>
<td>17.33</td>
<td>36.07</td>
<td>18.27</td>
</tr>
<tr>
<td>ROUGE (RL)</td>
<td>39.99</td>
<td>17.72</td>
<td>36.66</td>
<td>18.93</td>
</tr>
<tr>
<td>Entail (RL)</td>
<td>39.53</td>
<td>17.51</td>
<td>36.44</td>
<td>20.15</td>
</tr>
<tr>
<td>ROUGESal (RL)</td>
<td>40.36</td>
<td>17.97</td>
<td>37.00</td>
<td>19.84</td>
</tr>
<tr>
<td>ROUGE+Ent (RL)</td>
<td>40.37</td>
<td>17.89</td>
<td>37.13</td>
<td>19.94</td>
</tr>
<tr>
<td>ROUGESal+Ent (RL)</td>
<td>40.43</td>
<td>18.00</td>
<td>37.10</td>
<td>20.02</td>
</tr>
</tbody>
</table>

**Table:** Results on CNN/Daily Mail (nonanonymou). * represents previous work on anonymous version. ‘XE’: cross-entropy loss, ‘RL’: reinforce mixed loss (XE+RL). Columns ‘R’: ROUGE, ‘M’: METEOR. Final multi-reward RL model improvements are statistically significant over baseline, ROUGE-RL, Entail-RL.
Results (CNN/Daily Mail)

<table>
<thead>
<tr>
<th>Models</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Previous Work</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nallapati (2016)*</td>
<td>35.46</td>
<td>13.30</td>
<td>32.65</td>
<td>-</td>
</tr>
<tr>
<td>See et al. (2017)</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Paulus (2017) (XE)*</td>
<td>38.30</td>
<td>14.81</td>
<td>35.49</td>
<td>-</td>
</tr>
<tr>
<td>Paulus (2017) (RL)*</td>
<td>39.87</td>
<td>15.82</td>
<td>36.90</td>
<td>-</td>
</tr>
<tr>
<td><strong>Our Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline (XE)</td>
<td>39.41</td>
<td>17.33</td>
<td>36.07</td>
<td>18.27</td>
</tr>
<tr>
<td>ROUGE (RL)</td>
<td>39.99</td>
<td>17.72</td>
<td>36.66</td>
<td>18.93</td>
</tr>
<tr>
<td>Entail (RL)</td>
<td>39.53</td>
<td>17.51</td>
<td>36.44</td>
<td>20.15</td>
</tr>
<tr>
<td>ROUGESal (RL)</td>
<td>40.36</td>
<td>17.97</td>
<td>37.00</td>
<td>19.84</td>
</tr>
<tr>
<td>ROUGE+Ent (RL)</td>
<td>40.37</td>
<td>17.89</td>
<td>37.13</td>
<td>19.94</td>
</tr>
<tr>
<td>ROUGESal+Ent (RL)</td>
<td>40.43</td>
<td>18.00</td>
<td>37.10</td>
<td>20.02</td>
</tr>
</tbody>
</table>

Table: Results on CNN/Daily Mail (nonanonymous). * represents previous work on anonymous version. ‘XE’: cross-entropy loss, ‘RL’: reinforce mixed loss (XE+RL). Columns ‘R’: ROUGE, ‘M’: METEOR. Final multi-reward RL model improvements are statistically significant over baseline, ROUGE-RL, Entail-RL.
Results (CNN/Daily Mail)

<table>
<thead>
<tr>
<th>Models</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PREVIOUS WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nallapati (2016)*</td>
<td>35.46</td>
<td>13.30</td>
<td>32.65</td>
<td>-</td>
</tr>
<tr>
<td>See et al. (2017)</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
<td>18.72</td>
</tr>
<tr>
<td>Paulus (2017) (XE)*</td>
<td>38.30</td>
<td>14.81</td>
<td>35.49</td>
<td>-</td>
</tr>
<tr>
<td>Paulus (2017) (RL)*</td>
<td>39.87</td>
<td>15.82</td>
<td>36.90</td>
<td>-</td>
</tr>
<tr>
<td><strong>OUR MODELS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline (XE)</td>
<td>39.41</td>
<td>17.33</td>
<td>36.07</td>
<td>18.27</td>
</tr>
<tr>
<td>ROUGE (RL)</td>
<td>39.99</td>
<td>17.72</td>
<td>36.66</td>
<td>18.93</td>
</tr>
<tr>
<td>Entail (RL)</td>
<td>39.53</td>
<td>17.51</td>
<td>36.44</td>
<td>20.15</td>
</tr>
<tr>
<td>ROUGESal (RL)</td>
<td>40.36</td>
<td>17.97</td>
<td>37.00</td>
<td>19.84</td>
</tr>
<tr>
<td>ROUGE+Ent (RL)</td>
<td>40.37</td>
<td>17.89</td>
<td>37.13</td>
<td>19.94</td>
</tr>
<tr>
<td>ROUGESal+Ent (RL)</td>
<td>40.43</td>
<td>18.00</td>
<td>37.10</td>
<td>20.02</td>
</tr>
</tbody>
</table>

**Table**: Results on CNN/Daily Mail (nonanonymous). * represents previous work on anonymous version. ‘XE’: cross-entropy loss, ‘RL’: reinforce mixed loss (XE+RL). Columns ‘R’: ROUGE, ‘M’: METEOR. Final multi-reward RL model improvements are statistically significant over baseline, ROUGE-RL, Entail-RL.

Human evaluation: Our Multi-reward model is better than baseline.
Thank You
Machine Translation
Machine Translation

- Useful for tons of companies, online traffic, and our international communication!

Translate

This is an example of machine translation

यह मशीन अनुवाद का एक उदाहरण है

Yaha maśīna anuvāda kā ēka udāharaṇa hai
Statistical Machine Translation

- Source language f (e.g., French)
- Target language e (e.g., English)

We want the best target (English) translation given the source (French) input sentence, hence the probabilistic formulation is:

\[
\hat{e} = \arg\max_e p(e|f)
\]

Using Bayes rule, we get the following (since \(p(f)\) in the denominator is independent of the \(\arg\max\) over \(e\)):

\[
\hat{e} = \arg\max_e p(e|f) = \arg\max_e p(f|e)p(e)
\]
The first part is known as the ‘Translation Model’ \( p(f|e) \) and is trained on parallel corpora of \{f,e\} sentence pairs, e.g., from EuroParl or Canadian parliament proceedings in multiple languages.

The second part \( p(e) \) is the ‘Language Model’ and can be trained on tons more monolingual data, which is much easier to find!

\[
\text{French} \xrightarrow{\text{Translation Model}} p(f|e) \xrightarrow{\text{Pieces of English}} \xrightarrow{\text{Language Model}} p(e) \xrightarrow{\text{Decoder}} \text{argmax } p(f|e)p(e) \xrightarrow{\text{Proper English}}
\]
Statistical Machine Translation

- First step in traditional machine translation is to find alignments or translational matchings between the two sentences, i.e., predict which words/phrases in French align to which words/phrases in English.

- Challenging problem: e.g., some words may not have any alignments:

  ![Alignment Example](alignment_example.png)
Statistical Machine Translation

One word in the source sentence might align to several words in the target sentence:

“zero fertility” word not translated

one-to-many alignment

And

the

program

has

been

implemented

Le

programme

a

été

mis

en

application

And

the

program

has

been

implemented

Le

program

a

été

mis

en

application
Many words in the source sentence might align to a single word in the target sentence:

```
The balance was the territory of the aboriginal people
```

Many-to-one alignments:

```
The balance  Le  reste  appartent
was  the  appartenait
the  aux
of  aux
the
aboriginal  autochtones
people
```

```
The
balance
was
the
territory
of
the
aboriginal
people
```
And finally, many words in the source sentence might align to many words in the target sentence:

The poor don’t have any money

Les pauvres sont démunis

The poor don’t have any money

many-to-many alignment

phrase alignment
After learning the word and phrase alignments, the model also needs to figure out the reordering, esp. important in language pairs with very different orders!

```
er  geht  ja  nicht
  er  geht  ja nicht
   he  does not  go
       nach hause
         nach  hause
```
After many steps, you get the large ‘phrase table’. Each phrase in the source language can have many possible translations in the target language, and hence the search space can be combinatorially large!
Finally, you decode this hard search problem to find the best translation, e.g., using beam search on the several combinatorial paths through this phrase table (and also include the language model $p(e)$ to rerank).
Alignment Model Details

1. Align words with a probabilistic model
2. Infer presence of larger structures from this alignment
3. Translate with the larger structures

Yo lo haré mañana
I will do it tomorrow
Alignments: a hidden vector called an alignment specifies which English source is responsible for each French target word.

The first, simplest IBM model treated alignment probabilities as roughly uniform:

\[ a = a_1 \ldots a_J \]

\[
P(f, a | e) = \prod_j P(a_j = i)P(f_j | e_i)
\]

\[
= \prod_j \frac{1}{I + 1}P(f_j | e_i)
\]

\[
P(f | e) = \sum_a P(f, a | e)
\]

[Brown et al., 1993]
IBM Model 2 (Distortion)

The next more advanced model captures the notion of ‘distortion’, i.e., how far from the diagonal is the alignment

\[
P(f, a | e) = \prod_j P(a_j = i | j, I, J) P(f_j | e_i)
\]

\[
P(dist = i - j \frac{I}{J}) = \frac{1}{Z} e^{-\alpha(i-j \frac{I}{J})}
\]

Other approaches for biasing alignment towards diagonal include relative vs absolute alignment, asymmetric distances, and learning a full multinomial over distances

[Brown et al., 1993]
IBM Models 1/2 EM Training

- Model Parameters:
  - Translational Probabilities: $P(f_j|e_i)$
  - Distortion Probabilities: $P(a_j = i|j, I, J)$

- Start with uniform $P(f_j|e_i)$ parameters, including $P(f_j|\text{null})$
- For each sentence in training corpus:
  - For each French position $j$:
    - Calculate posterior over English positions using:
      $$P(a_j = i|f, e) = \frac{P(a_j = i|j, I, J)P(f_j|e_i)}{\sum_{i'} P(a_j = i'|j, I, J)P(f_j|e'_i)}$$
    - Increment count of word $f_j$ with word $e_i$ by these amounts
    - Similarly re-estimate distortion probabilities for Model 2
- Iterate until convergence

[Brown et al., 1993]
The HMM Model

**E:** Thank you, I shall do so gladly.

**A:**

1 → 3 → 7 → 6 → 8 → 8 → 8 → 8 → 9

**F:** Gracias, lo haré de muy buen grado.

**Model Parameters**

*Emissions:* \( P( F_1 = \text{Gracias} \mid E_{A1} = \text{Thank} ) \)

*Transitions:* \( P( A_2 = 3 \mid A_1 = 1 ) \)

[Vogel et al., 1996]
IBM Models 3/4/5 (Fertility)

Mary did not slap the green witch

Mary not slap slap slap the green witch

Mary not slap slap slap NULL the green witch

Mary no daba una botefada a la verde bruja

Mary no daba una botefada a la bruja verde

[Vogel et al., 1996]
IBM Models 3/4/5 (Fertility)

| f   | t(f | e) | φ  | n(φ | e) |
|-----|-------|-----|-------|
| le  | 0.497 | 1   | 0.746 |
| la  | 0.207 | 0   | 0.254 |
| les | 0.155 |     |       |
| l'  | 0.086 |     |       |
| ce  | 0.018 |     |       |
| cette | 0.011 |     |       |

| f   | t(f | e) | φ  | n(φ | e) |
|-----|-------|-----|-------|
| ne  | 0.497 | 2   | 0.735 |
| pas | 0.442 | 0   | 0.154 |
| non | 0.029 | 1   | 0.107 |
| rien| 0.011 |     |       |

**farmers**

| f              | t(f | e) | φ | n(φ | e) |
|----------------|-------|---|-------|
| agriculteurs   | 0.442 | 2 | 0.731 |
| les            | 0.418 | 1 | 0.228 |
| cultivateurs   | 0.046 | 0 | 0.039 |
| producteurs    | 0.021 |   |       |

[Vogel et al., 1996]
# Syntactic Machine Translation

<table>
<thead>
<tr>
<th>Type</th>
<th>System 1</th>
<th>System 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-string</td>
<td>ITG (Wu 1997)</td>
<td>Hiero (Chiang 2005)</td>
</tr>
<tr>
<td>tree-to-string</td>
<td></td>
<td>Huang et al 2006</td>
</tr>
</tbody>
</table>
Hiero

\[ S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \]
\[ S \rightarrow \langle X_1, X_1 \rangle \]
\[ X \rightarrow \langle yu X_1 you X_2, have X_2 with X_1 \rangle \]
\[ X \rightarrow \langle X_1 de X_2, the X_2 that X_1 \rangle \]
\[ X \rightarrow \langle X_1 zhiyi, one of X_1 \rangle \]
\[ X \rightarrow \langle Aozhou, Australia \rangle \]
\[ X \rightarrow \langle shi, is \rangle \]
\[ X \rightarrow \langle shaoshu guojia, few countries \rangle \]
\[ X \rightarrow \langle bangjiao, diplomatic relations \rangle \]
\[ X \rightarrow \langle Bei Han, North Korea \rangle \]

From [Chiang et al, 2005]
Synchronous Tree-Substitution Grammars

**STSG**

**extraction**

1. Phrases
   - respect word alignments
   - are syntactic constituents on both sides
2. Phrase pairs form rules
3. Subtract phrases to form rules

[Shieber, 2004; Graehl et al., 2008]
Joint Parsing and Alignment

Sample Synchronization Features

\[ \phi_{cd}(NP, b_8, NP) = \text{CoarseSourceTarget}(\text{phrasal}, \text{phrasal}) : 1 \]
\[ \phi_{d}(NN, b_7) = \text{CoarseSourceAlign}(\text{pos}) : 1 \]
\[ \phi_{d}(NP, NP) : 1 \]
\[ \phi_{d}(NP, NN) : 1 \]

[Burkett et al., 2011]
Neural Machine Translation (next week)