# Just News It: Abstractive Text Summarization with a Pointer-Generator Transformer

Alexandre Bucquet (bucqueta@stanford.edu) Vrinda Vasavada (vrindav@stanford.edu)

## Motivation

- **Abstractive text summarization** task
  - Create a summary which is **shorter** than the source document but contains its **main ideas**
  - More powerful than extractive text summarization which constrains summary to only contain words from input
  - Produces **more complex, human-like** summaries
- Evaluate See et al.'s **pointer-generator** model on complex pieces (*New York Times* op-eds) to understand areas of improvement for current models
- Experiment with combining this model with a **transformer**
  - Attractive because can be trained **in parallel** and proved in other tasks to **better model long-term dependencies**
  - Strictly **avoid source article shortening** unlike most other recent work

## Data

- *CNN/Daily Mail* dataset: multi-line summaries (56 tokens on average) for news articles (781 tokens on average)
- Manually collected 20 **op-eds** from the *New York Times*

## Baseline Model

- Used See et al.'s **pointer-generator** network for abstractive text summarization
- LSTM encoder/decoder model with **additive attention**
  - Attention computed from encoder hidden state h, decoder state s, and coverage vector c

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{att})$$

$$a^t = \text{softmax}(e^t)$$

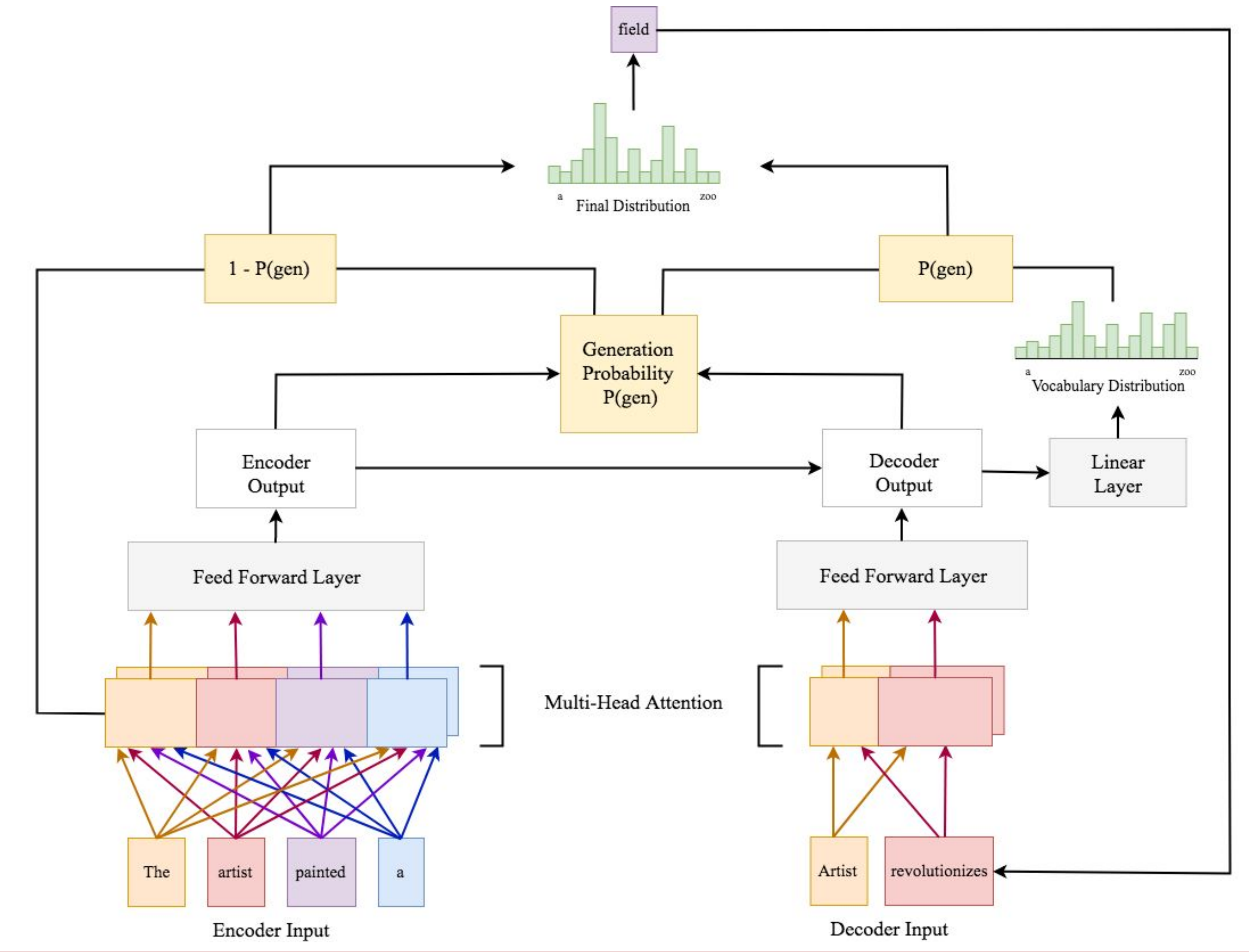
- Uses **generation probability** to decide whether to copy words from the source or generate a new word from vocabulary, **enabling preservation of factual details**

$$p_{gen} = \sigma(w_x x + w_y y_t + w_d d_t + b_{gen})$$

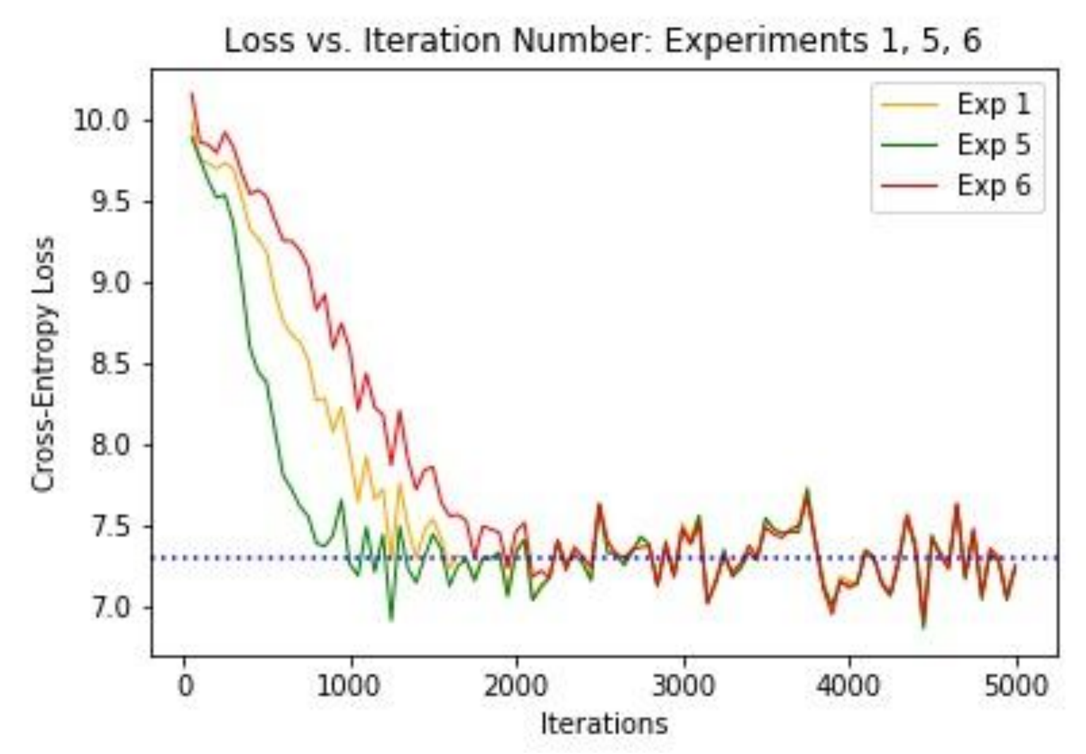
$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{w_i = w} a_i^t$$

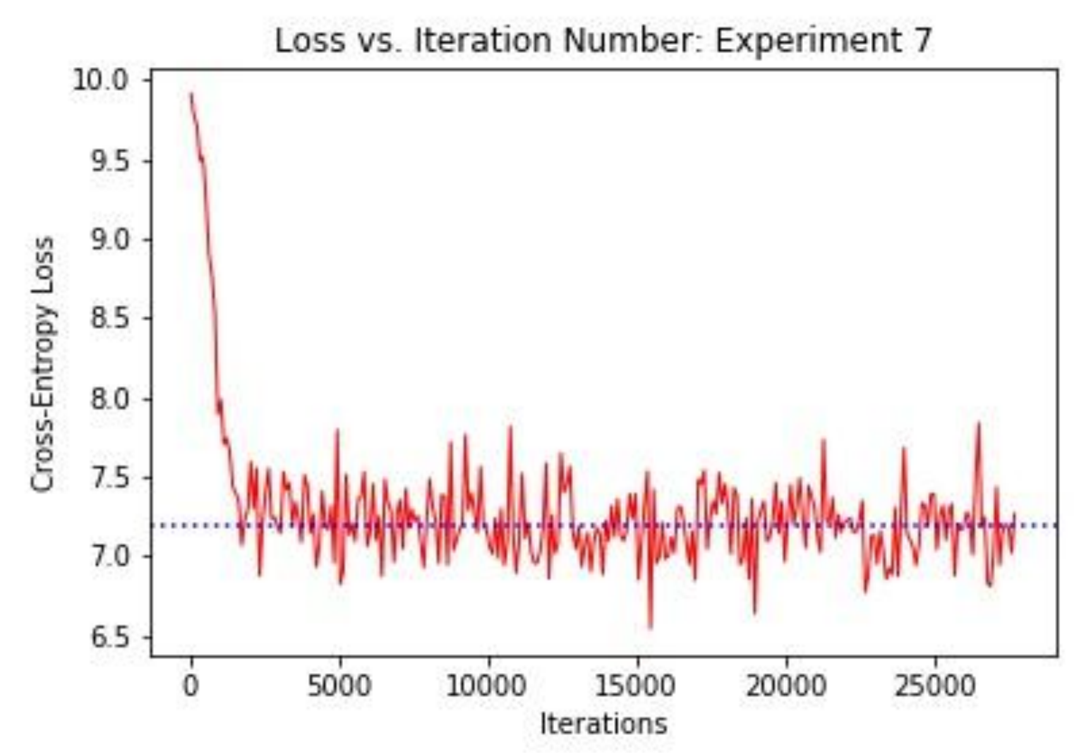
- Uses **coverage** (from Neural Machine Translation) to represent how much each word has contributed to the output summary thus far, **disincentivizing repetition of phrases**

## Model Architecture



## Quantitative Analysis





Transformer Model Performance

- Failed to match pointer-generator performance
- Learns average word distribution of the whole dataset, but **never commits to a prediction**
  - All experiments struggle to learn past this local optimum
- **Underfitting** might be a result of the **long input sequences**
  - All past applications of transformers for this task have **artificially shortened input articles**
  - No existing solution allows transformers to take in full input sequences

| Experiment | Heads | Layers | $d_k, d_q$ size | Warmup steps | Average loss past 2000 batches |
|---|---|---|---|---|---|
| 1 | 8 | 6 | 64 | 4000 | 7.301 |
| 2 | 16 | 6 | 64 | 4000 | 7.285 |
| 3 | 8 | 12 | 64 | 4000 | 7.292 |
| 4 | 8 | 6 | 128 | 4000 | 7.301 |
| 5 | 8 | 6 | 64 | 2000 | 7.286 |
| 6 | 8 | 6 | 64 | 6000 | 7.299 |
| 7 | 16 | 12 | 128 | 4000 | 7.248 |

Table 4: Hyperparameter Search on the Transpointer Model

$$lr = d_{model}^{-0.5} \cdot min(stepnum^{-0.5}, stepnum \cdot 4000^{-1.5})$$

## Qualitative Analysis

Pointer-Generator Performance on Test Set

*Reference*: "furious 7" pays tribute to star paul walker, who died during filming. vin diesel: "this movie is more than a movie." "furious 7" opens friday .

*Model Output*: paul walker's death in november 2013 at the age of 40 after a car crash. the actor was on a run at the age of 40 after a car crash in november 2013. the actor was on broken from filming `furious 7'' on friday.

*Reference*: hardy was convicted of domestic abuse against ex-girlfriend nicki holder and was suspended from the dallas cowboys for 10 days by the nfl. charges were eventually dropped after holder could not be located when hardy's lawyers appealed the decision and asked for a jury trial. this week he got stuck in his bentley in deep flash flood waters in dallas. hardy was forced to abandon his car and it was towed away hours later.

*Model Output*: greg hardy was suspended from 10 nfl games. he was fined $ [UNK] a week. He was dropped by his previous team, the sheriff says.

- Summaries often **shorter** than "gold" summaries
- Struggles to **attribute entities** and identify sources
- Often produces a somewhat **coherent but irrelevant** summary

Pointer-Generator Performance on Op-Eds

*First Paragraph Snippet:* I was 15 when I started smoking, and so were most of my friends... Because cigarettes were both forbidden and easy to get: ten quarters in a cigarette vending machine, which you could still find in most pizza joints and doughnut shops in suburban New Jersey in the early 1990s...

*Model Output:* [UNK] [UNK], who was 15 when i started smoking , has been found in the past 1990s. he says the idea of the most people who [UNK] start smoking by the end of the [UNK].

*Article Snippet:* ...Wabtec is willing to keep the existing workers at the current average of $35 per hour...

*Model Output:* [UNK] is willing to keep the existing workers at the current average of [UNK] per hour.

- Tends to collect only **information from first paragraph**
- Struggles with **unknown tokens**
- Often extracts a **full sentence** from the article

## Future Work

- Further **hyperparameter exploration**
- Apply transformers to the **easier task of extractive summarization**, in order to understand how transformers behave on long inputs
- Experiment with **adaptive attention windows** to reduce the number of neighbor words that are considered in attention computations

## References

- A. See, P. Liu, C. Manning. *Get To The Point: Summarization with Pointer-Generator Networks*. https://dblp.org/rec/bib/journals/corr/SeeLM17
- A. Vaswani et. al, *Attention is All You Need*. https://arxiv.org/abs/1706.03762
- A. Kumar, Pointer Summarizer (GitHub repository). https://github.com/atulkum/pointer_summarizer