Attention is all I need

José Ángel González¹
jogonba2@dsic.upv.es

¹Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València

January 26, 2020
Who I am

- Degree in Informatics Engineering (UPV).
- Master in Artificial Intelligence, Pattern Recognition and Digital Imaging (DSIC-UPV).
- PhD candidate (3rd year).
- Sentiment/Emotional Analysis and Irony Detection in Social Media & Text Summarization.
Outline

Transformers

Transformers for Text Classification

Attentional Extractive Summarization
Deep Learning for Sequence Modeling

- Dominance of convolutions and recurrences († 2017).
- Tendency to reduce the complexity of recurrent models:
  - GRU, Attention Mechanisms, Sequential computation ...
- Reducing the sequential computation to learn dependencies \textit{independently of the positions} with Attention Mechanisms.
- Transformers!

\[\text{Pictures from [34, 2]}\]
Transformers

Transformers for Text Classification

Attentional Extractive Summarization
Transformers

- Originally proposed as encoder-decoder model for NMT [28].
- Completely based on scaled dot-product attentions.
- More parallelizable (better suited for large and small datasets)
- The encoder is able to extract good text representations.
Self-Attention

- Learn representations from the all-vs-all interactions of the words e.g. $Z = XX^\top$
- $Q$, $K$, $V$ projections for computing the self-attentions.
- Output $Z$ is computed as a weighted sum of $V$.
- These weights are computed as a compatibility function of $Q$ and $K$.
- Advantage: path length between $X_i$ and $X_j$ of $\mathcal{O}(1)$
- What happens with the word order?

---

0 Pictures from [1]
Multi-head Attention

- Self-Attention applied $h$ times on the same input ($Z_{i \leq h}$ outputs are projected to $Z$)
- Allowing the model to jointly attend to information of different representation subspaces (e.g., heads detecting word coreferences)
- More parallelizable (even $d_q$, $d_k$ and $d_v$ can be smaller).

\[ \text{Picture from [1]} \]
Multi-head Attention
Insights

▶ Several strategies for adding positional information:
  ▶ Absolute/Relative positions [26]
  ▶ Heuristic rules / Positional embeddings [28]

▶ Optimization tricks for Deep Transformers:
  ▶ Noam learning rate schedule / LAMB [28] [33]
  ▶ Cross-Layer parameter sharing [17]
  ▶ Factorized embeddings [17]
  ▶ Gradient accumulation.

▶ Product Key Memory [16].
Product Key Memory

- To increase the network capacity without computational overhead [16].
- A 12-layered Transformer with one Product Key Memory can outperform a 24-layered Transformer.
Transformers

Transformers for Text Classification

Attentional Extractive Summarization
Transformers for Text Classification

- Moving from uncontextual pre-trained embeddings [21] to contextualized finetuning models [19, 5, 24].
- What if we work in social network texts and non-english languages?
- But we want to profit the capacity of the Transformers:
  1. Contextualize pre-trained embeddings [9, 10].
  2. Adapt finetuning models to our task [8]
Transformers for Text Classification

- Evaluation of the Transformer Encoders for Spanish Twitter text classification tasks:
  - Sentiment Analysis (TASS 2019) [6, 10]
  - Irony Detection (IroSVA 2019) [3, 9]
- Without an extensive search of the hyper-parameters.
- Same model and resources for both tasks.
- Are they more powerful than other Deep Learning approaches?
Tasks

- **TASS**: Assigning a global polarity to each tweet on four classes $\mathbb{C} = \{\text{N, NEU, NONE, P}\}$
- **IroSVA**: Determine the ironic content of each tweet in two classes $\mathbb{C} = \{\text{No-I, I}\}$
- **Spanish variants** (Peru, Costa Rica, Cuba, Mexico and Uruguay)

![TASS Distribution (ES)](chart1)

![IroSVA Distribution (ES)](chart2)
Experimental Details

- Skip-gram word embeddings ($d_e = 300 \& 87M$ tweets)
- Fixed most of the hyper-parameters:
  - $L \in \{1, 2\}$, $h = 8$, $d_q = d_k = d_v = 64$ and $d_{ff} = d_e$
- Sine-Cosine Positional Encoding
- Weighted cross-entropy using $w(c) = \frac{\max_{c' \in \mathcal{C}} N(c')}{N(c)}$
- Adam + Noam Learning Rate Annealing
- Macro-$F_1$ for evaluating TASS and $F_1$ for IroSVA.
- Comparison with DAN [15] and Att-LSTM [30].

$$
\mathcal{L}(\theta) = \mathbb{E}_D[\mathcal{L}(f(x; \theta), y)] = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{|\mathcal{C}|} y_{ij} \log f(x_i; \theta)_j w_j
$$
Comparison

- TE outperforms DAN & Att-LSTM for all metrics.
- Same behavior for all the other Spanish variants.
- Positional relationships are not useful in these corpora.

<table>
<thead>
<tr>
<th></th>
<th>MP</th>
<th>MR</th>
<th>MF&lt;sub&gt;1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>47.66</td>
<td>48.46</td>
<td>47.94</td>
</tr>
<tr>
<td>Att-LSTM</td>
<td>50.00</td>
<td>48.14</td>
<td>48.83</td>
</tr>
<tr>
<td>1-TE-NoPos</td>
<td>52.80</td>
<td><strong>54.38</strong></td>
<td><strong>53.34</strong></td>
</tr>
<tr>
<td>1-TE-Pos</td>
<td>46.26</td>
<td>46.56</td>
<td>46.25</td>
</tr>
<tr>
<td>2-TE-NoPos</td>
<td><strong>52.85</strong></td>
<td>53.03</td>
<td>51.47</td>
</tr>
<tr>
<td>2-TE-Pos</td>
<td>47.31</td>
<td>48.79</td>
<td>47.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>F&lt;sub&gt;10&lt;/sub&gt;</th>
<th>F&lt;sub&gt;11&lt;/sub&gt;</th>
<th>MF&lt;sub&gt;1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>85.78</td>
<td>69.79</td>
<td>77.78</td>
</tr>
<tr>
<td>Att-LSTM</td>
<td>81.05</td>
<td>66.05</td>
<td>73.54</td>
</tr>
<tr>
<td>1-TE-NoPos</td>
<td><strong>85.79</strong></td>
<td><strong>74.18</strong></td>
<td><strong>79.98</strong></td>
</tr>
<tr>
<td>1-TE-Pos</td>
<td>81.63</td>
<td>65.38</td>
<td>73.51</td>
</tr>
<tr>
<td>2-TE-NoPos</td>
<td>84.05</td>
<td>69.27</td>
<td>76.65</td>
</tr>
<tr>
<td>2-TE-Pos</td>
<td>82.64</td>
<td>62.83</td>
<td>72.74</td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>$MF_1$</th>
<th>MP</th>
<th>MR</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES</td>
<td>50.70</td>
<td>50.50</td>
<td>50.80</td>
<td>1/9</td>
</tr>
<tr>
<td>CR</td>
<td>49.60</td>
<td>49.80</td>
<td>49.30</td>
<td>2/9</td>
</tr>
<tr>
<td>PE</td>
<td>44.70</td>
<td>45.60</td>
<td>43.90</td>
<td>2/9</td>
</tr>
<tr>
<td>UY</td>
<td>51.50</td>
<td>49.70</td>
<td>53.60</td>
<td>2/7</td>
</tr>
<tr>
<td>MX</td>
<td>50.10</td>
<td>49.00</td>
<td>51.20</td>
<td>1/9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CU</th>
<th>ES</th>
<th>MX</th>
<th>Avg</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELiRF-UPV</td>
<td>65.27</td>
<td>71.67</td>
<td>68.03</td>
<td>68.32</td>
<td>1/18</td>
</tr>
<tr>
<td>CIMAT</td>
<td>65.96</td>
<td>64.49</td>
<td>67.09</td>
<td>65.85</td>
<td>2/18</td>
</tr>
<tr>
<td>LDSE</td>
<td>63.35</td>
<td>67.95</td>
<td>66.08</td>
<td>65.79</td>
<td>3/18</td>
</tr>
<tr>
<td>JZaragoza</td>
<td>61.63</td>
<td>66.05</td>
<td>67.03</td>
<td>64.90</td>
<td>4/18</td>
</tr>
<tr>
<td>W2V Baseline</td>
<td>60.33</td>
<td>68.23</td>
<td>62.71</td>
<td>63.76</td>
<td>5/18</td>
</tr>
</tbody>
</table>
The compatibility function between $Q$ and $K$ of the heads allows us to explain some properties captured by the system.

Be $A_{ijk}$ the attention that the word $i$ puts in the word $j$ in the attention head $k$: 

![Attention Matrix 3rd Head](image)

![Attention Matrix 4th Head](image)
Attention Analysis

- We study the relationships captured by the multi-head self-attention mechanism.
- These relationships are task dependent:
  - **Sentiment Analysis:**
    - Word polarities
    - Polarity modifiers (shifters and intensifiers)
  - **Irony Detection**
    - Ironic attention heads
    - Impact of polarity words
    - Relevance of individual words
    - Word pair relationships
Attention Analysis

How can we analyze if our system takes them into account?

Computing the average attention that each word receives from all the other words for each head, averaged for all the occurrences of the word in a dataset.

\[
\alpha["it"][1] = [... \, + \, \alpha["it"]][1] \\
\vdots \\
\alpha["it"][1]
\]
Sentiment: Detecting Word Polarity

- **mejor (best)**
- **maravilloso (wonderful)**
- **genial (cool)**
- **peor (worst)**
- **horrible (horrible)**
- **mierda (shit)**
Sentiment: Detecting Word Polarity

$C(w) = \begin{cases} \mathcal{P} & \alpha_{w4} \leq \alpha_{w5} \\ \mathcal{N} & \alpha_{w4} > \alpha_{w5} \end{cases}$ classifies correctly the $74.75 \pm 3.17\%$ of the samples from ElHuyar lexicon.

- Only considering the distribution of two independent heads!
Heads 4 and 5 do not react to non-polarity words.
Head 1 seems to react to all polarity modifiers to a greater or lesser extent.
The attention distributions for polarity modifiers are very similar.
If we switch-off an attention head and the $F_1(1)$ decreases, that head is related with the Irony.

Iterative process for masking attention heads.

We explore incrementally the $2^h - 2$ combinations of maskings.

The heads that appear in more combinations that worsen the $F_1(1)$ are the Ironic Heads.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>$H_0$</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>$H_4$</th>
<th>$H_5$</th>
<th>$H_6$</th>
<th>$H_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>IroSVA</td>
<td>16/18</td>
<td>11/18</td>
<td>13/18</td>
<td>10/18</td>
<td>8/18</td>
<td>9/18</td>
<td>4/18</td>
<td>5/18</td>
</tr>
</tbody>
</table>
Irony: Impact of polarity words

- Which words are the most attended by $H_{ironic}$ heads?
- We compute the average attention given by each head $k \in H_{ironic}$ to the word $w$, $\alpha[w][k]$
- If $\alpha[w][k] > \epsilon$, $w$ is highly attended by $k$
- Polarity lexicons to analyze the polarity of these highly attended words.

| Head Set | Heads | $|\alpha_w > \epsilon|$ | Negative | Positive | Ratio  |
|----------|-------|--------------------------|----------|----------|--------|
|          | $H_0$ | 240                      | 102      | 24       | 52.50% |
|          | $H_1$ | 221                      | 12       | 18       | 13.57% |
| $H_{ironic}$ | $H_2$ | 73                       | 22       | 8        | 41.09% |
|          | $H_3$ | 603                      | 140      | 47       | 31.01% |
|          | $\Sigma$ | 1137                  | 276      | 97       | 32.80% |
|          | $H_4$ | 276                      | 14       | 28       | 15.21% |
|          | $H_5$ | 116                      | 6        | 9        | 12.60% |
| $H_{no,ironic}$ | $H_6$ | 281                      | 41       | 11       | 18.50% |
|          | $H_7$ | 237                      | 14       | 18       | 13.50% |
|          | $\Sigma$ | 910                    | 75       | 66       | 15.50% |
Irony: Impact of individual words

- Are there words that determine the irony?
- Two approaches:
  - Average attention given by $H_{ironic}$
    \[ B \leftarrow \sum_{k \in H_{ironic}} \text{softmax}\left(\frac{f(X)q_k f(X)_k^T}{\sqrt{d_k}}\right) ; B' j \leftarrow \frac{1}{|X|} \sum_{i=1}^{|X|} B_{ij} \]
  - Euclidean norm of the input gradients:
    \[ B' j \leftarrow \| \nabla_X \mathcal{L}(f(X; \theta), y = 1)_j \| \]
## Irony: Word relationships

<table>
<thead>
<tr>
<th>Language</th>
<th>Example</th>
<th>Top-5 Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>1</td>
<td>(sleep, fun), (christmas, fun)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(going, fun), (2hrs, fun), (shopping, fun)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(look, storm), (sydney, unusual),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(’, oh), ( , storm), (unusual, oh)</td>
</tr>
<tr>
<td>Spanish</td>
<td>1</td>
<td>(fallen, butter), (down, butter), (butter, side),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(side, flat), (earth, side)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(book, April Fool’s joke), (pedro, book),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>([clap emoji], book), (year, book), (seems, book)</td>
</tr>
</tbody>
</table>
Transformers

Transformers for Text Classification

Attentional Extractive Summarization
Text Summarization

- Need to condense big amounts of unstructured information available in media platforms.
- Approaches to automatic summarization can be divided in:
  - Extractive
  - Abstractive
  - Mixed
- The most common human strategy to summarize documents consists in applying an ordered sequence of these approaches.
- The first step consists in focusing on the most relevant sentences (Extractive approach)
- Extractive Neural Summarization systems to the rescue!
Attentional Extractive Summarization

- Typical neural approaches states the problem as a sequential binary sentence classification problem.
- No corpora with this kind of labeling:
  - Suboptimal extractive oracles [4, 22, 18]
  - Reinforcement Learning [23, 35, 7, 32]
- Attentional approaches do not require a sentence labeling.
- They simplify the sequential classification problem.
- Based on the interpretation of Attentional Networks after being trained to solve a proxy binary classification task: distinguishing correct (document, summary) pairs.
All systems under this framework are based on two main ideas:

**First idea**

If we can say if a summary $y$ is correct for a document $x$ and we can look at the relevant sentences in $x$ that led us to that decision, then we can build a summary $\hat{y}$, composed by the relevant sentences in $x$, that is similar to the reference $y$.

**Second Idea**

If $y$ is a correct summary for a document $x$, then $y$ and $x$ have similar semantics (similar representations) while if $w$ is an incorrect summary for a document $x$, then $w$ and $x$ have less similar semantics (less similar representations).
From these two ideas we proposed Siamese Hierarchical Attention Networks based on Attentional LSTM encoders [31, 12]
Siamese Hierarchical Transformer Encoders

- The attention mechanism can learn word-level relationships such as coreference [27], coherence [27], anaphora [29], etc.
- But also sentence-level relationships!
- We propose to use Transformer Encoders in a hierarchical way, to process sequences of sentences by replacing the Attentional LSTM of SHA-NN.
- The sentence relevances are implicitly computed by the multi-head self-attention mechanisms.
Siamese Hierarchical Transformer Encoders

\[
\mathcal{L}(\Theta) = \sum_{k=1}^{|\mathcal{P}|} L(f(X_k, X'_k; \Theta), y = 1) + \mathbb{E}_{p(X_j \neq k | X_k)}[L(f(X_k, X'_j; \Theta), y = 0)]
\]
Once the model is trained for the proxy task, the compatibility function at sentence level can be used to detect relevant sentences [11]

Differently from SHA-NN [12], the compatibility function of these attention mechanisms do not assign a real value to each sentence:

**Sentence Relevance Hypothesis**

If a sentence $s$ is greatly attended on average by all the other sentences $s'$ for all the attention heads $h$, this sentence condenses a big part of the information of the sentences $s'$, being thus, more relevant.
before you go, we thought you’d like these … if you want a face lift, and you have the time and money to make that happen, go for it.

but if you want a non-invasive alternative to surgery to help you get younger-looking skin, you need a good device.

the nuface trinity is a skin care device designed with interchangeable treatment attachments to help with facial stimulation and the reduction of fine lines and wrinkles.

in as little as five minutes a day, you can improve your facial contour and skin tone.

watch beauty expert jenny patinkin show you just how it easy it is to use this device.

looking for something else? check out the video below to keep shopping!
Automatically collected corpora from newspaper domains:
  - CNN/DailyMail [14].
  - NewsRoom (BBC, Time, Bloomberg, Telegraph, ...) [13].

The summaries of these corpora are the highlights written manually by the editors.

Biased towards the first article sentences.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Set</th>
<th>Sentences</th>
<th></th>
<th>Words</th>
<th></th>
<th>Words/Sentence</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Articles</td>
<td></td>
<td>Articles</td>
<td></td>
<td>Articles</td>
<td></td>
</tr>
<tr>
<td>CNN/DailyMail</td>
<td>Train</td>
<td>31.87</td>
<td>3.79</td>
<td>750.10</td>
<td>51.58</td>
<td>23.53</td>
<td>13.61</td>
</tr>
<tr>
<td></td>
<td>Dev</td>
<td>26.77</td>
<td>4.11</td>
<td>737.06</td>
<td>57.57</td>
<td>27.53</td>
<td>14.00</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>27.11</td>
<td>3.88</td>
<td>745.59</td>
<td>54.65</td>
<td>27.51</td>
<td>14.07</td>
</tr>
<tr>
<td>NewsRoom</td>
<td>Train</td>
<td>29.91</td>
<td>1.40</td>
<td>773.57</td>
<td>30.37</td>
<td>25.86</td>
<td>21.65</td>
</tr>
<tr>
<td></td>
<td>Dev</td>
<td>29.69</td>
<td>1.41</td>
<td>767.34</td>
<td>30.72</td>
<td>25.84</td>
<td>21.73</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>29.62</td>
<td>1.41</td>
<td>765.56</td>
<td>30.63</td>
<td>25.84</td>
<td>21.68</td>
</tr>
</tbody>
</table>
Results

- Similar results to PGen+Cov [25], SummaRunner [22], DQN [32] and Refresh [23].
- Best models take profit of pre-trained language models [18], Reinforcement Learning [7, 35] or word-length strategies [20].

<table>
<thead>
<tr>
<th>System</th>
<th>Strategy</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead-3</td>
<td>Ext</td>
<td>40.24</td>
<td>17.70</td>
<td>36.45</td>
</tr>
<tr>
<td>SHA-NN</td>
<td>Ext</td>
<td>39.99</td>
<td>17.75</td>
<td>36.27</td>
</tr>
<tr>
<td>SHTE</td>
<td>Ext</td>
<td>39.96</td>
<td>17.60</td>
<td>36.19</td>
</tr>
<tr>
<td>SummaRunner</td>
<td>Ext/OC</td>
<td>39.60</td>
<td>16.20</td>
<td>35.30</td>
</tr>
<tr>
<td>ECS-Ext</td>
<td>Ext/OC</td>
<td>41.70</td>
<td>18.60</td>
<td>37.80</td>
</tr>
<tr>
<td>BertSumEXT</td>
<td>Ext/OC</td>
<td>43.25</td>
<td>20.24</td>
<td>39.63</td>
</tr>
<tr>
<td>Refresh</td>
<td>Ext/RL</td>
<td>40.00</td>
<td>18.20</td>
<td>36.60</td>
</tr>
<tr>
<td>DQN</td>
<td>Ext/RL</td>
<td>39.40</td>
<td>16.10</td>
<td>35.60</td>
</tr>
<tr>
<td>Latent</td>
<td>Ext/RL</td>
<td>41.10</td>
<td>18.80</td>
<td>37.40</td>
</tr>
<tr>
<td>BanditSum</td>
<td>Ext/RL</td>
<td>41.50</td>
<td>18.70</td>
<td>37.60</td>
</tr>
<tr>
<td>ECS-Comp</td>
<td>Mix</td>
<td>40.90</td>
<td>18.00</td>
<td>37.40</td>
</tr>
<tr>
<td>Latent-Comp</td>
<td>Mix</td>
<td>36.70</td>
<td>15.40</td>
<td>34.30</td>
</tr>
<tr>
<td>PGen+Cov</td>
<td>Mix</td>
<td>39.53</td>
<td>17.28</td>
<td>36.38</td>
</tr>
<tr>
<td>CopyCat</td>
<td>Mix</td>
<td>39.15</td>
<td>17.60</td>
<td>36.17</td>
</tr>
</tbody>
</table>
Results

- Corpus divided in 3 subsets relating to the extractive degree (density)
- Extracting $k = 2$ better than $k = 3$ (not in the abstractive subset).
- Except ECS, our proposal outperforms all the neural models.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead-3</td>
<td>Ext</td>
<td>51.98</td>
<td>47.85</td>
<td>51.20</td>
<td>25.62</td>
<td>13.00</td>
<td>22.30</td>
<td>14.57</td>
<td>2.62</td>
<td>11.73</td>
<td>30.66</td>
<td>21.09</td>
<td>28.35</td>
</tr>
<tr>
<td>Lead-2</td>
<td>Ext</td>
<td>57.87</td>
<td>53.03</td>
<td>56.83</td>
<td>28.60</td>
<td>14.33</td>
<td>24.46</td>
<td>15.68</td>
<td>2.77</td>
<td>12.35</td>
<td>33.98</td>
<td>23.30</td>
<td>31.14</td>
</tr>
<tr>
<td>SHAN-2</td>
<td>Ext</td>
<td>54.83</td>
<td>49.25</td>
<td>53.72</td>
<td>28.03</td>
<td>13.79</td>
<td>23.85</td>
<td>15.67</td>
<td>2.74</td>
<td>12.29</td>
<td>32.78</td>
<td>21.86</td>
<td>29.85</td>
</tr>
<tr>
<td>SHTE-2</td>
<td>Ext</td>
<td>53.97</td>
<td>47.87</td>
<td>52.59</td>
<td>27.78</td>
<td>13.41</td>
<td>23.56</td>
<td>15.57</td>
<td>2.67</td>
<td>12.22</td>
<td>32.38</td>
<td>21.25</td>
<td>29.40</td>
</tr>
<tr>
<td>ECS-Ext</td>
<td>Ext</td>
<td>69.40</td>
<td>64.30</td>
<td>68.30</td>
<td>31.90</td>
<td>16.30</td>
<td>26.90</td>
<td>17.20</td>
<td>3.10</td>
<td>13.60</td>
<td>39.50</td>
<td>27.90</td>
<td>36.26</td>
</tr>
<tr>
<td>PGen+Cov</td>
<td>Mix</td>
<td>39.10</td>
<td>28.00</td>
<td>36.20</td>
<td>25.50</td>
<td>11.00</td>
<td>21.10</td>
<td>14.70</td>
<td>2.30</td>
<td>11.40</td>
<td>26.43</td>
<td>13.76</td>
<td>22.90</td>
</tr>
<tr>
<td>TLM</td>
<td>Mix</td>
<td>53.30</td>
<td>44.20</td>
<td>50.10</td>
<td>28.10</td>
<td>12.10</td>
<td>23.00</td>
<td>18.50</td>
<td>3.90</td>
<td>14.70</td>
<td>33.30</td>
<td>20.06</td>
<td>29.26</td>
</tr>
<tr>
<td>FastRL</td>
<td>Mix</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ECS-Comp</td>
<td>Mix</td>
<td>68.40</td>
<td>62.90</td>
<td>67.30</td>
<td>31.70</td>
<td>16.10</td>
<td>27.00</td>
<td>17.10</td>
<td>3.10</td>
<td>14.10</td>
<td>39.06</td>
<td>27.36</td>
<td>36.13</td>
</tr>
</tbody>
</table>
Convergence

▶ SHTE requires visiting more samples than SHA-NN until convergence, but few training hours (4 – 5× for NewsRoom).
▶ It obtains lower results in terms of Acc on (document, summary) pairs, but similar results for ROUGE (CE mismatch [23]).
▶ Faster than other approaches for CNN/DailyMail:
  ▶ BanditSum: 76 hours (single GPU Nvidia Titan Rx)
  ▶ DQN: 10 days (single GPU Nvidia 1080 Ti)
  ▶ Refresh: 12 hours (single GPU)

<table>
<thead>
<tr>
<th>Corpora</th>
<th>System</th>
<th>Epoch</th>
<th>Samples</th>
<th>Loss</th>
<th>Acc</th>
<th>Acc Error</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN/DailyMail</td>
<td>SHA-NN</td>
<td>82</td>
<td>2,624,000</td>
<td>0.007</td>
<td>99.62 ± 0.10</td>
<td>3.51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHTE</td>
<td>13</td>
<td>4,160,000</td>
<td>0.209</td>
<td>91.92 ± 0.46</td>
<td>2.38</td>
<td></td>
</tr>
<tr>
<td>NewsRoom</td>
<td>SHA-NN</td>
<td>159</td>
<td>5,088,000</td>
<td>0.083</td>
<td>96.16 ± 0.11</td>
<td>6.45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHTE</td>
<td>9</td>
<td>5,760,000</td>
<td>0.230</td>
<td>90.61 ± 0.17</td>
<td>1.65</td>
<td></td>
</tr>
</tbody>
</table>
Two interesting observations on SHTE:

- How affects the positional information?
  - Is it required positional information?
  - Is it required positional information on both levels?
  - What if only using sentence positional information?

- What heads capture better the sentence relevance?
  - Are there individual heads related to condense information?
  - What about averaging heads?

- Is the word-length distribution obtained by SHA-NN & SHTE similar to the distribution on the reference summaries?
## Analysis

<table>
<thead>
<tr>
<th>Head</th>
<th>Precision R-1</th>
<th>Precision R-2</th>
<th>Precision R-L</th>
<th>Recall R-1</th>
<th>Recall R-2</th>
<th>Recall R-L</th>
<th>$F_1$ R-1</th>
<th>$F_1$ R-2</th>
<th>$F_1$ R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24.28</td>
<td>7.92</td>
<td>21.79</td>
<td>45.06</td>
<td>15.15</td>
<td>40.38</td>
<td>29.75</td>
<td>9.80</td>
<td>26.68</td>
</tr>
<tr>
<td>2</td>
<td>24.58</td>
<td>8.11</td>
<td>22.13</td>
<td>44.15</td>
<td>14.90</td>
<td>39.64</td>
<td>29.89</td>
<td>9.92</td>
<td>26.88</td>
</tr>
<tr>
<td>3</td>
<td>24.79</td>
<td>7.97</td>
<td><strong>22.29</strong></td>
<td>43.48</td>
<td>14.42</td>
<td>38.98</td>
<td>29.64</td>
<td>9.62</td>
<td>26.62</td>
</tr>
<tr>
<td>5</td>
<td>24.49</td>
<td>7.94</td>
<td>22.02</td>
<td>43.40</td>
<td>14.39</td>
<td>38.90</td>
<td>29.61</td>
<td>9.66</td>
<td>26.58</td>
</tr>
<tr>
<td>6</td>
<td>24.42</td>
<td>7.60</td>
<td>21.89</td>
<td>41.90</td>
<td>13.33</td>
<td>37.41</td>
<td>29.00</td>
<td>9.09</td>
<td>25.95</td>
</tr>
<tr>
<td>Avg Heads</td>
<td>24.67</td>
<td><strong>8.23</strong></td>
<td>22.16</td>
<td><strong>45.45</strong></td>
<td><strong>15.53</strong></td>
<td><strong>40.73</strong></td>
<td><strong>30.20</strong></td>
<td><strong>10.15</strong></td>
<td><strong>27.10</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Head</th>
<th>Precision R-1</th>
<th>Precision R-2</th>
<th>Precision R-L</th>
<th>Recall R-1</th>
<th>Recall R-2</th>
<th>Recall R-L</th>
<th>$F_1$ R-1</th>
<th>$F_1$ R-2</th>
<th>$F_1$ R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27.79</td>
<td>11.07</td>
<td>25.21</td>
<td>51.31</td>
<td>20.78</td>
<td>47.34</td>
<td>34.76</td>
<td>13.82</td>
<td>31.51</td>
</tr>
<tr>
<td>2</td>
<td>27.17</td>
<td>10.66</td>
<td>24.62</td>
<td>52.36</td>
<td>20.67</td>
<td>47.38</td>
<td>34.29</td>
<td>13.47</td>
<td>31.06</td>
</tr>
<tr>
<td>3</td>
<td>29.19</td>
<td>11.71</td>
<td>26.53</td>
<td>51.74</td>
<td>20.86</td>
<td>46.98</td>
<td>35.83</td>
<td>14.39</td>
<td>32.55</td>
</tr>
<tr>
<td>4</td>
<td><strong>29.84</strong></td>
<td><strong>12.09</strong></td>
<td><strong>27.15</strong></td>
<td>52.17</td>
<td>21.24</td>
<td>47.41</td>
<td>36.15</td>
<td>14.58</td>
<td>33.16</td>
</tr>
<tr>
<td>5</td>
<td>29.12</td>
<td>11.87</td>
<td>26.48</td>
<td><strong>53.09</strong></td>
<td><strong>21.66</strong></td>
<td><strong>48.19</strong></td>
<td>36.03</td>
<td>14.68</td>
<td>32.74</td>
</tr>
<tr>
<td>6</td>
<td>29.60</td>
<td>12.01</td>
<td>26.91</td>
<td>52.30</td>
<td>21.30</td>
<td>47.45</td>
<td>36.21</td>
<td>14.73</td>
<td>32.99</td>
</tr>
<tr>
<td>Avg Heads</td>
<td>29.64</td>
<td>12.03</td>
<td>26.97</td>
<td>52.46</td>
<td>21.36</td>
<td>47.67</td>
<td><strong>36.36</strong></td>
<td><strong>14.76</strong></td>
<td><strong>33.37</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Head</th>
<th>Precision R-1</th>
<th>Precision R-2</th>
<th>Precision R-L</th>
<th>Recall R-1</th>
<th>Recall R-2</th>
<th>Recall R-L</th>
<th>$F_1$ R-1</th>
<th>$F_1$ R-2</th>
<th>$F_1$ R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24.68</td>
<td>8.12</td>
<td>22.13</td>
<td>44.20</td>
<td>14.70</td>
<td>39.59</td>
<td>30.11</td>
<td>9.94</td>
<td>27.03</td>
</tr>
<tr>
<td>2</td>
<td>23.91</td>
<td>7.84</td>
<td>21.51</td>
<td>44.34</td>
<td>14.87</td>
<td>39.79</td>
<td>29.45</td>
<td>9.74</td>
<td>26.47</td>
</tr>
<tr>
<td>3</td>
<td><strong>25.83</strong></td>
<td><strong>9.69</strong></td>
<td><strong>23.32</strong></td>
<td>50.38</td>
<td>18.98</td>
<td>45.37</td>
<td>32.16</td>
<td>11.74</td>
<td>28.95</td>
</tr>
<tr>
<td>5</td>
<td>25.23</td>
<td>8.86</td>
<td>22.72</td>
<td>47.47</td>
<td>17.02</td>
<td>42.68</td>
<td>31.38</td>
<td>11.10</td>
<td>28.24</td>
</tr>
<tr>
<td>6</td>
<td>23.94</td>
<td>7.49</td>
<td>21.56</td>
<td>39.29</td>
<td>12.76</td>
<td>35.82</td>
<td>28.35</td>
<td>8.94</td>
<td>25.49</td>
</tr>
<tr>
<td>Avg Heads</td>
<td>25.33</td>
<td>9.42</td>
<td>22.84</td>
<td><strong>50.92</strong></td>
<td><strong>19.02</strong></td>
<td><strong>45.85</strong></td>
<td><strong>32.40</strong></td>
<td><strong>12.04</strong></td>
<td><strong>29.18</strong></td>
</tr>
</tbody>
</table>
Analysis

Density

Reference
\( \mu = 57.544, \sigma = 25.033 \)

Lead 3
\( \mu = 93.200, \sigma = 23.670 \)

SHANN-3
\( \mu = 94.375, \sigma = 23.505 \)

SHTE-3
\( \mu = 95.411, \sigma = 23.886 \)

Density

Reference
\( \mu = 46.700, \sigma = 47.450 \)

Lead 2
\( \mu = 61.390, \sigma = 26.754 \)

SHANN-2
\( \mu = 66.138, \sigma = 24.734 \)

SHTE-2
\( \mu = 63.516, \sigma = 26.383 \)

Density

Reference
\( \mu = 46.700, \sigma = 47.450 \)

Lead 3
\( \mu = 89.061, \sigma = 33.150 \)

SHANN-3
\( \mu = 96.448, \sigma = 30.626 \)

SHTE-3
\( \mu = 92.800, \sigma = 32.382 \)
References I


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural machine translation by jointly learning to align and translate.

Reynier Ortega Bueno, Francisco M. Rangel Pardo, Delia Irazú Hernández Farías, Paolo Rosso, Manuel Montes-y-Gómez, and José Medina-Pagola.
Overview of the task on irony detection in spanish variants.

Jianpeng Cheng and Mirella Lapata.
Neural summarization by extracting sentences and words.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
BERT: pre-training of deep bidirectional transformers for language understanding.

Overview of TASS 2019: One more further for the global spanish sentiment analysis corpus.
Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung.
BanditSum: Extractive summarization as a contextual bandit.

José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla.
Twilbert: Pre-trained deep bidirectional transformers for spanish twitter.
To be finished.

José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla.
Elirf-upv at irosva: Transformer encoders for spanish irony detection.

José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla.
Elirf-upv at TASS 2019: Transformer encoders for twitter sentiment analysis in spanish.

José-Ángel González, Encarna Segarra, Fernando García-Granada, Emilio Sanchis, and Lluís-F Hurtado.
Extractive summarization using siamese hierarchical transformer encoders.

José-Ángel González, Encarna Segarra, Fernando García-Granada, Emilio Sanchis, and Lluís-F Hurtado.
Siamese hierarchical attention networks for extractive summarization.
Max Grusky, Mor Naaman, and Yoav Artzi.
Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom.
Teaching machines to read and comprehend.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III.

Guillaume Lample, Alexandre Sablayrolles, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou.
Large memory layers with product keys.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut.

Yang Liu and Mirella Lapata.
Text summarization with pretrained encoders.
References IV

Yijia Liu, Wanxiang Che, Yuxuan Wang, Bo Zheng, Bing Qin, and Ting Liu.  
Deep contextualized word embeddings for universal dependency parsing.  

Jointly extracting and compressing documents with summary state representations.  

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean.  
Distributed representations of words and phrases and their compositionality.  

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou.  
Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.  

Shashi Narayan, Shay B. Cohen, and Mirella Lapata.  
Ranking sentences for extractive summarization with reinforcement learning.  
Language models are unsupervised multitask learners.
2019.

Abigail See, Peter J. Liu, and Christopher D. Manning.
Get to the point: Summarization with pointer-generator networks.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani.
Self-attention with relative position representations.

Dario Stojanovski and Alexander Fraser.
Coreference and coherence in neural machine translation: A study using oracle experiments.

Attention is all you need.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov.
Context-aware neural machine translation learns anaphora resolution.
References VI

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao.
Attention-based LSTM for aspect-level sentiment classification.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy.
Hierarchical attention networks for document classification.

Kaichun Yao, Libo Zhang, Tiejian Luo, and Yanjun Wu.
Deep reinforcement learning for extractive document summarization.
Neurocomputing, 284, 02 2018.

Reducing BERT pre-training time from 3 days to 76 minutes.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena.
Self-attention generative adversarial networks.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou.
Neural latent extractive document summarization.
Attention is all I need

José Ángel González\textsuperscript{1}
jogonba2@dsic.upv.es

\textsuperscript{1}Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València

January 26, 2020