# Hierarchical Attention Networks for Document Classification

Emilio Cecchini

*emilio.cecchini@stud.unifi.it*

June 15, 2020

Università degli Studi di Firenze

# Introduction

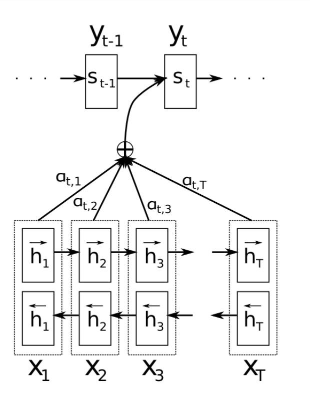
- The goal of this project is to replicate the experiments of the 2016 paper of Yang et al. [2].
- The main topic of the paper is document classification.
- The thesis of the authors is that a better representation can be obtained by incorporating knowledge of document structure in the model architecture.
- They introduced a new architecture: the Hierarchical Attention Network (HAN)
- All the code used for the experiments are available at `https://github.com/ceccoemi/han`

# Attention mechanism

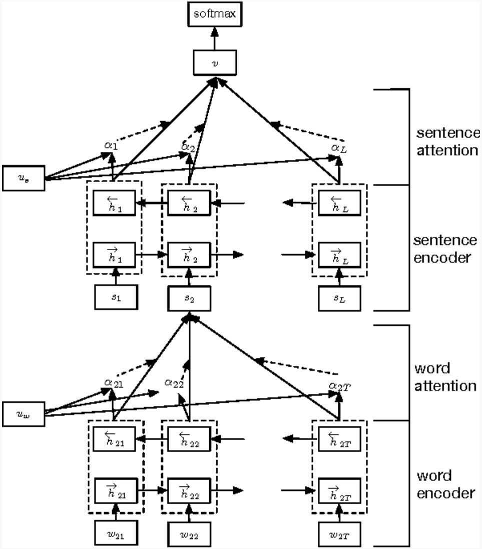
- Attention mechanism was first introduced by Bahdanau et al. in 2014 [1] as a new RNN encoder-decoder architecture in the topic of machine translation.
- The previously proposed RNN encoder-decoder models encode a source sentence into a fixed-length vector from which a decoder generates a translation.
- Bahdanau et al. conjectured that the use of a fixed-length vector is a bottleneck in improving the performance of these models, so they proposed a new method that automatically (soft-)search for parts of a source sentence that are relevant to predict a target word.

# Attention mechanism

- The probability $\alpha_{i,j}$ reflects the importance of the annotation $h_j$ with respect to the previous hidden state $s_{i-1}$ in deciding the next state $s_i$, and generating $y_i$.
- The decoder decides parts of the source sentence to pay attention to.



# Hierarchical Attention architecture



# Word encoder

- Given the $i$th sentence of length $T$ with words $w_{it}, t \in [0, T]$, we first embed the words to vectors through a pretrained embedding matrix $W_e$.
- A bidirectional GRU is used to get words annotations $\overrightarrow{h_{it}}$ and $\overleftarrow{h_{it}}$, which are finally concatenated into a unique word annotation $h_{it}$, which summarizes the information of the whole sentence centered around $w_{it}$.

$$
\begin{aligned}
x_{it} &= W_e w_{it} \\
\overrightarrow{h_{it}} &= \overrightarrow{\text{GRU}}(x_{it}) \\
\overleftarrow{h_{it}} &= \overleftarrow{\text{GRU}}(x_{it}) \\
h_{it} &= [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]
\end{aligned}
$$

# Word attention

- The word annotation $h_{it}$ is fed through a one-layer MLP to get $u_{it}$.
- Then the importance of the word is measured with a word level context vector $u_w$. More precisely, a normalized importance $\alpha_{it}$ is obtained through a softmax function.
- Finally, a sentence vector $s_i$ is computed as a weighted sum of the word annotations and their importance.

$$u_{it} = \tanh\left(W_w h_{it} + b_w\right)$$

$$\alpha_{it} = \frac{\exp\left(u_{it}^T u_w\right)}{\sum_t \exp\left(u_{it}^T u_w\right)}$$

$$s_i = \sum_t \alpha_{it} h_{it}$$

# Sentence encoder

- Given the sentence vector $s_i$, a sentence annotation $h_i$ is obtained in the same way as the word encoder.
- A bidirectional GRU is used and a sentence annotation $h_i$ is obtained, which summarizes the neighbor sentences around sentence $i$ but still focus on sentence $i$.

$$\overrightarrow{h_i} = \overrightarrow{\text{GRU}}(s_i)$$
$$\overleftarrow{h_i} = \overleftarrow{\text{GRU}}(s_i)$$
$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$$

# Sentence attention

- The same attention mechanism used for the words is introduced to get a word vector $v$ that summarizes all the information of the sentences in a document.

$$u_i = \tanh\left(W_s h_i + b_s\right)$$

$$\alpha_i = \frac{\exp\left(u_i^T u_s\right)}{\sum_i \exp\left(u_i^T u_s\right)}$$

$$v = \sum_t \alpha_i h_i$$

# Document classification

- Since the goal is document classification, a last step is required
- Given the document vector $v$, a probability vector $p$ is obtained with a MLP with a softmax activation:

$$p = \mathrm{softmax}\left(W_c v + b_c\right)$$

- As training loss, a negative log likelihood is used:

$$L = -\sum_{d} \log p_{dj}$$

# Data sets

- To evaluate the new proposed model, the authors used six different data sets: Yelp 2013, Yelp 2014, Yelp 2015, IMDB review, Yahoo answer and Amazon review.
- I choose three of the six dataset: Yelp, Yahoo and Amazon. Unfortunately, I wasn't able to find the exactly Yelp dataset used in the paper (while regarding Yahoo and Amazon I used the same data sets).
- Each data set is perfectly balanced and 80% of the data is used for training, 10% for validation and 10% for test.

| **Data set** | **classes** | **documents** |
|---|---|---|
| Yelp | 5 | 700,000 |
| Yahoo answer | 10 | 1,450,000 |
| Amazon review | 5 | 3,650,000 |

# Models

- The proposed model is compared with three other models:
  - BoW (Bag-of-Words)
  - Flat Attention Network (FAN)

Note that the results of the FAN model is not reported in the paper.

# BoW

- The 50,000 most frequent words from the training set are selected and the count of each word is used as features.
- A Stochastic Gradient Descent classifier is used together with a logistic regression loss.
- A grid search cross-validation is used to find the best value for the regularization term $\alpha$.

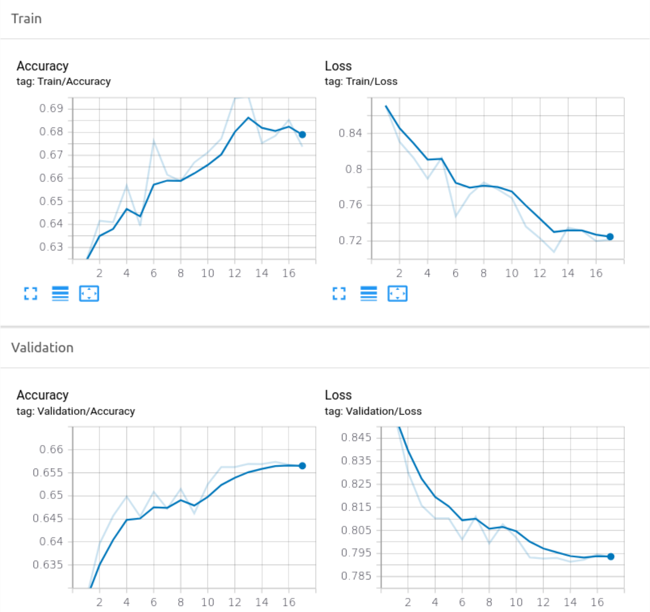
| | **BoW** | |
|---|---|---|
| **Data set** | **Yang et al. [2]** | **Observed** |
| Yelp | 58.0 | 61.3 |
| Yahoo | 68.9 | 66.9 |
| Amazon | 54.4 | 52.2 |

**Table 1:** Document classification, in percentage

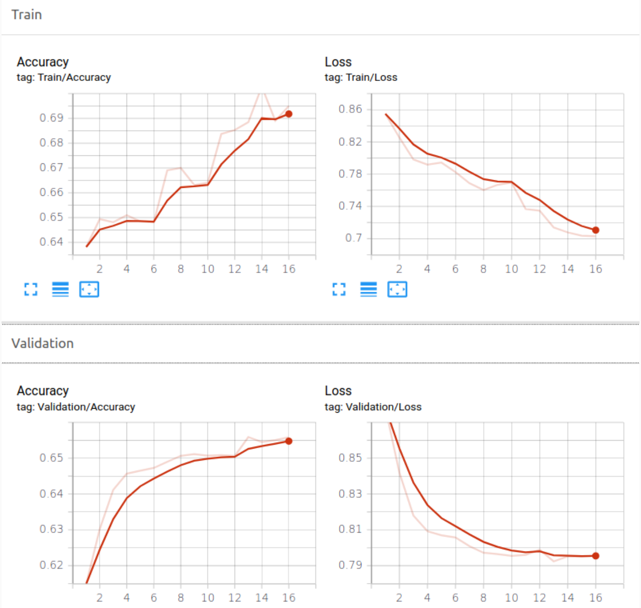
# FAN and HAN

- Regarding model configuration, hyperparameters and training I tried to follow the setup reported in the paper. The only difference is that Yang et al. [2] used grid search to find the best learning rate. Due to the high computational cost of the training I wasn't able to do that. I used a decresing learning rate instead.
- The embedding matrix was trained on the training and validation set. The word embedding dimension was set to 200.
- The GRU dimension was set to 50, so a bidirectional GRU gives 100 dimensions for word/sentence annotation.
- For training, a mini-batch size of 64 was used.
- The number of epochs are not specified in the paper. I choose to use early stopping with patience equals to 3.
- Stochastic gradient descent with momentum of 0.9 was used.

# Training (FAN model - Yelp)

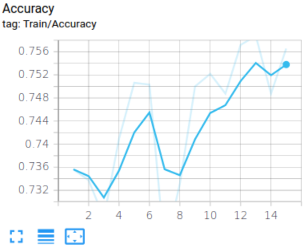


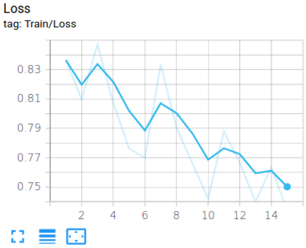
# Training (HAN model - Yelp)



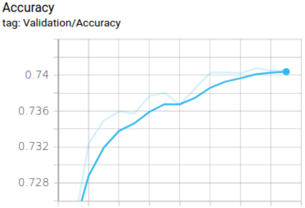
# Training (FAN model - Yahoo)

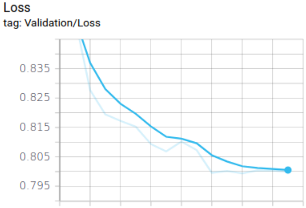
Train




Validation

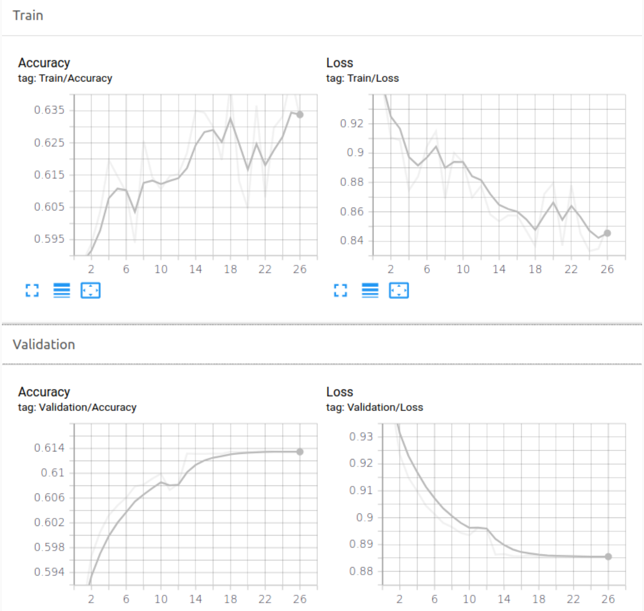


# Training (HAN model - Yahoo)

# Training (FAN model - Amazon)

# Training (HAN model - Amazon)



# Experimental results

| | Flat attention (FAN) | Hierarchical attention (HAN) | |
|---|---|---|---|
| **Data set** | **Observed** | **Yang et al. [2]** | **Observed** |
| Yelp | 65.6 | 68.2 | 65.6 |
| Yahoo | 74.1 | 75.8 | 74.4 |
| Amazon | 61.7 | 63.6 | 62.5 |

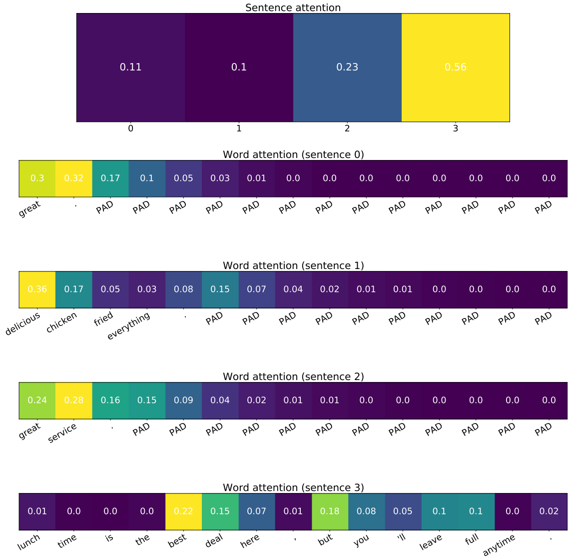
**Table 2:** Document classification, in percentage

# Attention visualization

- An interesting feature of the attention mechanism is that it's easy to debug. That's because, as reported in the paper, you can easly visualize the how much importance each word and sentence has for the classification task.
- I've tried to reproduce this attention visualization.

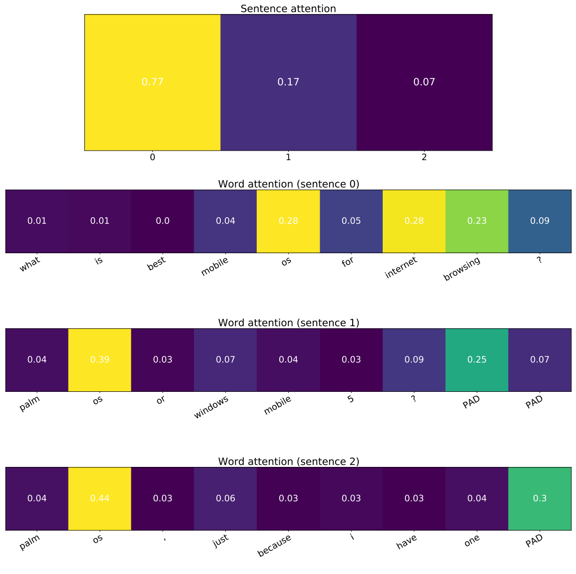
# Yelp - Attention visualization

label = 5; predicted = 5



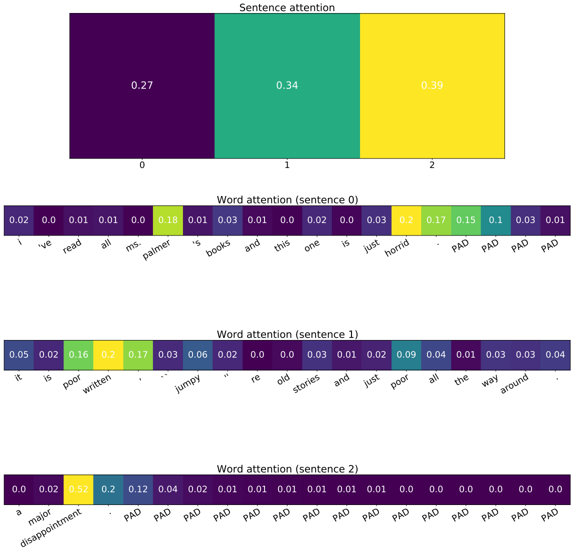
# Yahoo - Attention visualization

label = *Computers & Internet*; predicted = *Computers & Internet*



# Amazon - Attention visualization

label = 1; predicted = 1



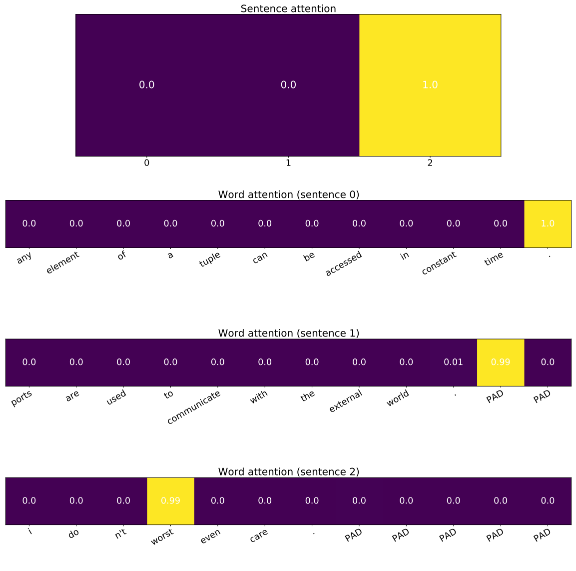
# Synthetic data set

- To verify that my model works properly, I've created a synthetic data set composed of random text. For each document, I added a keyword which is associated to a specific label.
- After the training, the model is always able to identify the keyword and it always predict the correct label (100% on test set)

| **Keywords** | **Label** |
|---|---|
| *bad, worst, dirty, irritating, disgusting* | 0 |
| *vague, vain, untouchable, selfish, rude* | 1 |
| *perverse, possessive, arrogant, cruel, calm* | 2 |
| *clever, comfortable, creative, clean, gentle* | 3 |
| *nice, fantastic, good, modern, quite* | 4 |

# Synthetic data set - Attention visualization

label = 0; predicted = 0



# Conclusions

- I was able to verify that the HAN model performs better than BoW and than the flat attention.
- However, I wasn't able to obtain the scores reported in the paper with none of the three data sets.
- Maybe, it was due to the padding/cropping procedure that I had to implement to limit the computational cost.

| | **BoW** | | **FAN** | **HAN** | |
|---|---|---|---|---|---|
| **Data set** | **Yang et al. [2]** | **Observed** | **Observed** | **Yang et al. [2]** | **Observed** |
| Yelp | 58.0 | 61.3 | 65.6 | 68.2 | 65.6 |
| Yahoo | 68.9 | 66.9 | 74.1 | 75.8 | 74.4 |
| Amazon | 54.4 | 52.2 | 61.7 | 63.6 | 62.5 |

**Table 3:** Document classification, in percentage

Thanks for your attention

# References i


D. Bahdanau, K. Cho, and Y. Bengio.
**Neural machine translation by jointly learning to align and translate.**
*arXiv preprint arXiv:1409.0473*, 2014.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy.
**Hierarchical attention networks for document classification.**
In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.