3^{rd} year of Economist-Statistician Magistere's degree
M2 Case study number 3
Session 2021-2022

Attention is all you need[1]

Reviews classification (positive/negative)

Charlotte Bredy-Maux & Kayané Robach

Abstract

Attention mechanism is one of the leading process in sentiment analysis to capture contextual information. While this mechanism is born from graph theory it has been shown efficient in miscellaneous fields such as visual attention to describe images content Xu et al. [2015], sentence summarization in Rush et al. [2015] or document classification Yang et al. [2016]. The present study will focus on the latter and investigate hierarchical attention mechanism on textual data for movies reviews classification. Results obtained on data from the Internet Movie Database (IMDb) corroborates a previous work on document classification (Yang et al. [2016]) and argues that the model does a great sentiment analysis on the documents. The aim of this project is to understand and reproduce a hierarchical attention network (HAN) architecture and to illustrate its effectiveness for the purpose of classifying reviews rather than to compare the efficiency of our network to the literature.

Key words: Self-Attention, Hierarchical Attention Network, Sentiment Analysis

[1] From Vaswani et al. [2017]'s famous paper.

Contents

1 Introduction

The epistemology of Attention Theory in Deep Learning takes roots in Graph-based Sentiment Analysis (SA). Indeed, before the years 1990's, Artificial Neural Networks (ANN) were mainly fed with structured and unstructured data represented in Euclidian space e.g. audio, text, images, videos in 1D, 2D or even 3D grids. The digital revolution led by the democratization of Internet of the Things (IoT) and Social Networks in particular, put Graph Theory at the forefront of the Deep Learning Research agenda. New methods and optimization frameworks needed to be created in order to process the information encompassed within this new kind of data structure. Namely, and albeit the scant knowledge about the underlying theory behind Graph-based ANNs, working with non-Euclidian spaces topologies compelled the academic community to address the plateauing of ANNs scalability, processing speed and local trap i.e., vanishing gradient effect from a new perspective. To such extent, GCNs experimentations extended RNNs, CNNs (Yann leCun) and MLPs to graph-structured data i.e. Recurrent GNNs (RecGNNs), Graph Convolutional Networks (GCNs) and Graph Autoencoders (GAEs), respectively. Aforementioned network's architectures essentially differ from their original counterpart in different ways. Pertaining to the ReCNNs versus RNNs comparison for example, the latent set of necessary conditions imposed on the former framework relates to the basic data structure. The said data needs to be hierarchically organised in a top-down stream with the root node or vector point defined as the 'super-source'. If all data points are related to each other according to such a defined and uni-directional causal chain, convergence of RNNs towards a local optimum is guarantee. On the other hand, GNNs'graph input relies on Dynamic Optimization Theory and namely needs to:

- live in Banach Spaces
- be mapped by a Contraction

The stability of the network derived from the two main points stated hereinabove then translates into the Banach Fixed Point Theorem and recovers the existence of a local optimum. In other words, restraining graph topography and inter-node relationship reflects the special case of graphs driven by a similar dynamic from period to period. When the state transition function mapping each node to its nearest neighbours is not a Contraction, i.e., when its jacobian matrix is not bounded by a number smaller than 1, a penalty term is added to the objective loss function in order to force convergence towards a steady-state of the dynamic system of nodes. Then, Attention Analysis comes as a refinement layer on top of the foundational framework of Graph-based ANN, and mainly consists in quantifying inter-node connections through weights. GCNs namely re-think the structure of the first convolution layers by weighting undirected graphs prior to pooling. Such methods have first been used with images, considering that pixels are the nodes, and that their adjacencies translate into edges. Later on, work on GCNs fed with textual data led researchers to understand that feature extraction could also greatly benefit from better design of the convolution layer, namely in sentiment analysis (SA). Indeed, words encompass great heterogeneity in the polarity they convey to the broader sentence. To such extent, researchers address this heterogeneity by weighting each word throughout the network's layers. This mechanism directly refers to Attention and hereby differs from traditional CNN methods that only use min/max/average pooling. In particular, the attention layer can thus be seen as a 'soft sparsing matrix', which goal is to essentialize the meaning of a text, and drive the network's focus on the relatively most weighted subset of words in each epoch.

Following this nascent arch of research in SA on Graph-Structured data, we thereby propose to apply attention convolutioning on traditional Euclidian text data for classification. The goal of the study is to determine whether canonical CNNs can benefit from the advances made in GCNs on non-Graph data. The second main architectural feature of our network relates to the hierarchical nature of Euclidian text data. Indeed, because micro-elements of a corpus, i.e. words, make macro-structures, i.e. sentences, which in turns form paragraphs and then documents, it might be interesting to apply attention principle on it as a sentiment analysis measure to do classification.

In the present body of work, we will implement a Hierarchical Attention Network (HAN) upon a Recurrent Neural Network (RNN) with Gated Recurrent Units (GRU) based on self-attention mechanism to accomplish a movie reviews classification task. Attention in such context will allow the network to weight the attention it should pay on words in specific sentences and on sentences in specific reviews. In the end, the model built will return a probability of being a positive review in the purpose of producing classification reviews.

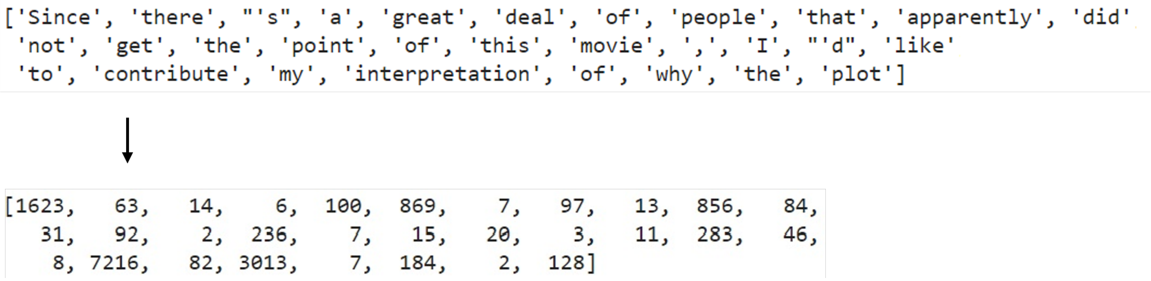
We will first describe our dataset in Section 2 before presenting some reminders on the RNN architecture and introducing the GRU cells in Section 3. We will then present self-attention mechanism in Section 4 and, in the end we will talk about our model, the HAN; we will produce results for the classification task and we will discuss about it in Section 5.

2 Material

For this project we use a dataset for binary sentiment classification i.e. positive or negative, of 25000 polar movie reviews from the Internet Movie Database (IMDb) labeled and 25000 reviews for testing. We found the dataset on kaggle and it has been produced in the context of a paper about sentiment analysis, [Maas et al., 2011].

Raw text has been preprocessed in a simple way. Since some movies collected substantially more reviews than others, the dataset is limited at 30 reviews per movie. All words across the corpus are then integrated in a dictionary of words without removing stop words, punctuation or abbreviations. Stop words and punctuation might be indicative of sentiment (e.g. negative words, '!', ':)', ...) and abbreviations has been kept to allow our future model to learn different but similar representations of words when suited.

The dictionary of vocabulary is turned into a list of indices where the '0' stands for the special padding token and '1' for Out-Of-Vocabulary elements that are encoded with the 'OOV' token. Then in the set of indices words are sorted according to their frequency—small tokens refer to frequent words—and we get a total of almost 30000 tokens.



```
['Since', 'there', "'s", 'a', 'great', 'deal', 'of', 'people', 'that', 'apparently', 'did'
 'not', 'get', 'the', 'point', 'of', 'this', 'movie', ',', 'I', "'d", 'like'
 'to', 'contribute', 'my', 'interpretation', 'of', 'why', 'the', 'plot']
```

```
[1623,   63,   14,    6,  100,  869,    7,   97,   13,  856,   84,
   31,   92,    2,  236,    7,   15,   20,    3,   11,  283,   46,
    8, 7216,   82, 3013,    7,  184,    2,  128]
```

Figure 1: Example of sentence encoding with respect to the vocabulary dictionary.

We will now use a particular architecture of the RNN, self-attention and a HAN model for movies reviews classification on this dataset from the Internet Movie Database.

3 Reminders on RNN with Gated Recurrent Units

When one wants to process a sentence or a movie to do a prediction for example, we sent the set of words or the set of frames into a network. However a network will not necessarily process the set of words or frames correctly by taking into account the temporal structure of the set. Hence the motivation for the Recurrent Neural Network (RNN) where each element of the set (e.g. a word or a frame) would be passed to the network one at a time. The RNN then process each input one by one while updating its cell as it goes along.

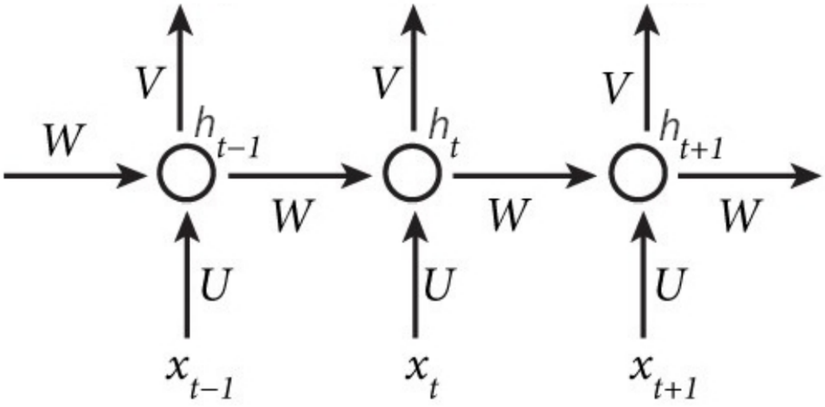


Figure 2: Unrolled representation of a RNN (3 steps).

The input x_{t-1} at time $t-1$ is passed through the U channel; in a translation purpose one might wants to return a word for each word passed in input (through the V channel) and to update the cell (through the W channel) in order to give it some memory about the sentence context while translating the following input x_t. In a prediction purpose the V channel is no longer needed. On Fig. 2 we can see an unrolled representation of the RNN in which the main cell evolves along the elements passed in inputs.

The GRU, Gated Recurrent Unit, refers to a specific architecture of the RNN cell.

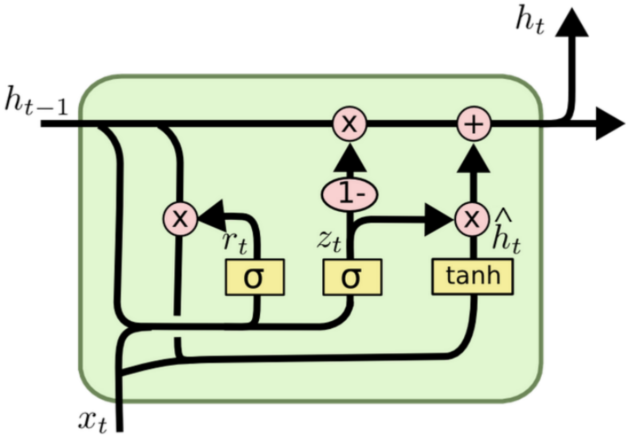


Figure 3: Typical GRU cell architecture. – Image

The GRU cell at time t has two gates, the reset one r_t and the update one z_t, as we can see on Fig. 3, Gangwar and Vadlamani [2020]. h_{t-1} illustrates the condition of the cell in the previous step $t-1$, x_t is the element passed in input to the network at time t, from both elements h_{t-1} and x_t the RNN cell should update and become h_t at time t. The reset gate r_t acts as a filter on what should be remembered from h_{t-1}, the update gate z_t choose what should be transformed from h_{t-1}. From the filtered and transformed h_{t-1} and the input x_t the cell produces an intermediary update of itself $\hat{h}_t$ to be merged with another transformation of the h_{t-1} to become h_t (linear interpolation between h_{t-1} and the intermediary update $\hat{h}_t$).

Such architecture of the cell has been shown to be an efficient recurrent unit in RNN, Chung et al. [2014]. Moreover, we will use a bi-directional GRU structure of cell in this project based on a famous work on document classification thanks to hierarchical attention, Yang et al. [2016]. Bi-directional, because it will collect information both ways, from the beginning to the end and from the end to the beginning in order to get annotations (encoding words and sentences) incorporating contextual information; we develop this architecture in Section 5.

In the literature, the sequence of update of the cell in the RNN—the sequence of hidden states $(h_1, \dots, h_T)$—is often called sequence of annotations, Bahdanau et al. [2014].

On Fig. 3 σ is the sigmoid function, the reset gate and the update gate are defined as follows: The intermediary candidate for the new hidden state $\hat{h}_t$ and the final one h_t obtained by a linear

$$\left| \begin{aligned} r_t &= \sigma(U_r x_t + W_r h_{t-1} + b_r) \\ z_t &= \sigma(U_z x_t + W_z h_{t-1} + b_z) \end{aligned} \right.$$

interpolating between the previous hidden state h_{t-1} and the candidate one $\hat{h}_t$ are computed as follows:

$$\left| \begin{aligned} \hat{h}_t &= \tanh(U_h x_t + W_h(r_t \circ h_{t-1}) + b_h) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \hat{h}_t \end{aligned} \right.$$

4 Self-attention for HAN

You probably are aware of self-attention albeit you do not know it. Indeed, it is a component of the transformer architecture, which is at the root of the BERT model Devlin et al. [2018].

The self-attention mechanism considers the sequence of annotations $(h_1, \dots, h_T)$ instead of summarizing the network process with the last annotation only. Therefore self-attention consists in producing an attentional vector s which is a weighted sum of the annotations.

$$\left| \begin{aligned} & s = \textstyle\sum_{t=1}^{T} \alpha_t h_t \\ & \alpha_t = \frac{\exp(u_t^\top c)}{\sum_{l=1}^{T} \exp(u_l^\top c)} \\ & c\text{, a trainable context vector} \\ & u_t = \tanh(W h_t) \end{aligned} \right.$$

To summarize, the annotations $(h_1, \dots, h_T)$ are passed through a dense layer, the output from this operation: $(u_1, \dots, u_T)$ is compared to a trainable context vector c and by normalizing this comparison with a softmax, the alignments coefficients $(\alpha_1, \dots, \alpha_T)$ are computed. When taking a new input (e.g. a new sentence) the model uses the trainable context vector—that might be interpreted as an average sentence optimally analyzed (in terms of attention)—to decide how much it should pay attention to each word in the sentence. In general, this context vector is initialized randomly and updated during training.

The use of self-attention in the transformer model is motivated by its capacity to take care of the dependencies between words in their sequential representation. In the transformer model recurrent operations are replaced by self-attention, Vaswani et al. [2017]. This choice make sense since the self-attention process is more efficient while faster (the number of sequential operation for self-attention layers is constant whereas it is proportional to the number of items in recurrent layers).

5 Hierarchical Attention Network

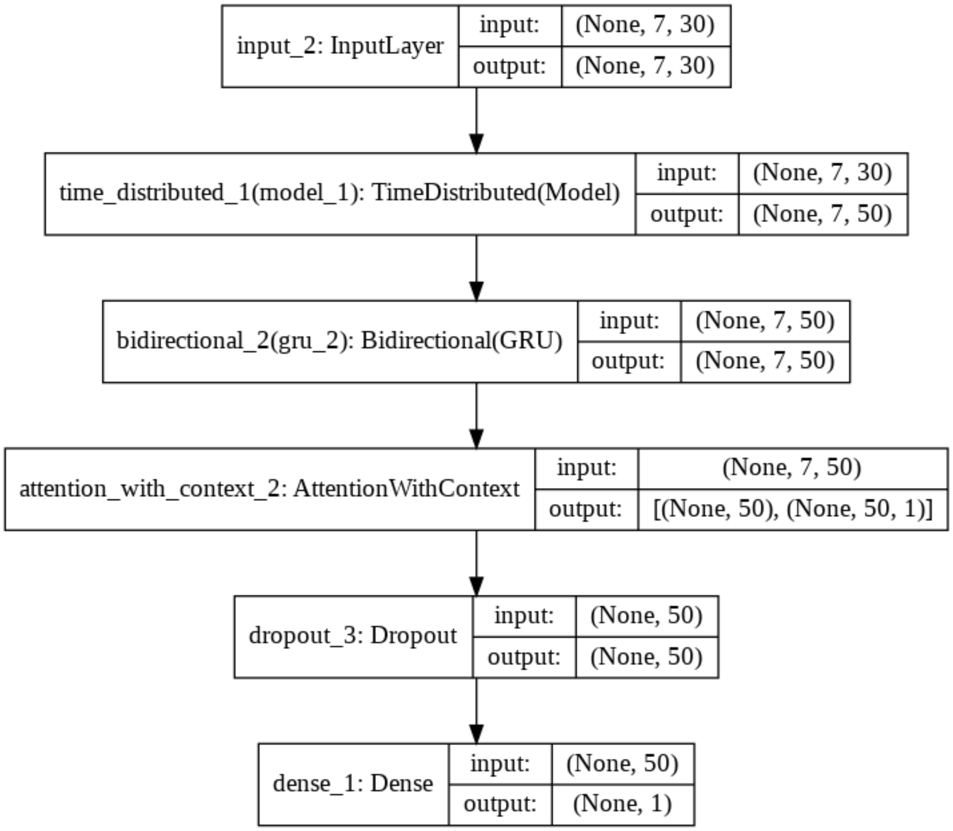


Figure 4: Model architecture built with keras on python. – Image produced by the authors based on our model built with Keras on python.

In this section we will study the HAN architecture, which aims to produce reviews embeddings based on sentence embeddings. We will classify movie reviews preprocessed and turned into arrays of shape (1, *review size*, *sentence size*), where *review size* is the maximal number of sentences per review and *sentence size* is the maximal number of words per sentence. Smaller sentences or smaller reviews were padded with a special character. Longer sentences or longer reviews were truncated. In our case, we set the maximal number of sentences per review at 7 and the maximal number of words per sentence at 30.

Each integer in the encodings of sentences and then reviews refers to a vocabulary word. The dictionary of vocabulary (built from the training set) gathered 29936 elements. Again, the index

‘0’ refers to the special token for padding and the index ‘1’refers to the Out-Of-Vocabulary token. The training needs 2 epochs to optimize the validation loss and is not so expensive—at most 200s/epoch on the google collab CPU—and ontly takes a few minutes.

The model we build in the purpose of ‘understanding’ feelings expressed in movies reviews uses a Recurrent Neural Network (RNN) with bi-directional Gated Recurrent Units (GRU), its architecture follows the work of Yang et al. [2016]. The two GRU cells that constitute the bi-directional GRU network collect information from past and future simultaneously in order to collect context information and detect pattern wherea CNN would only have detected n-grams.

5.1 HAN architecture

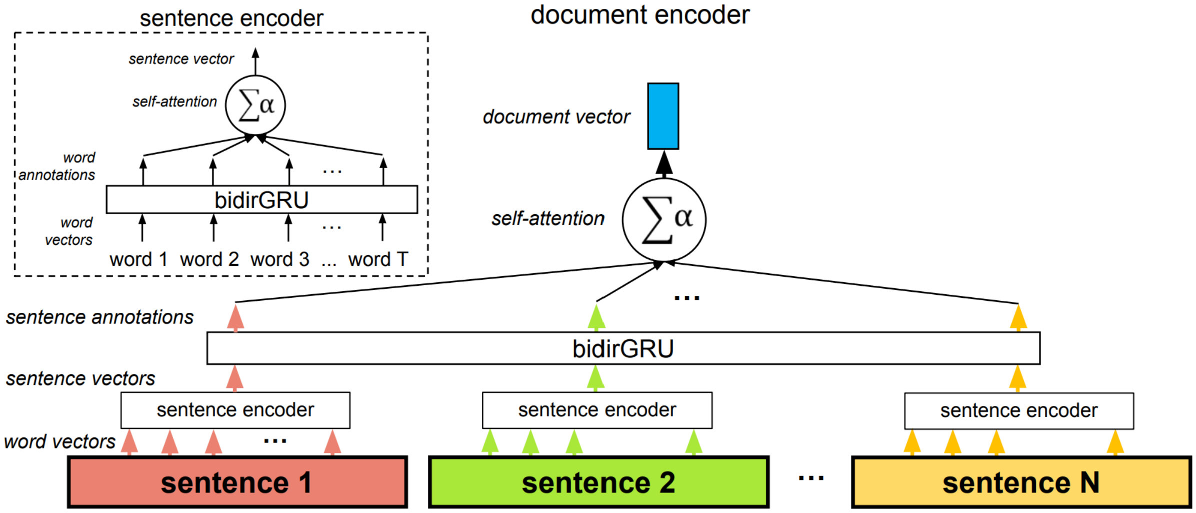
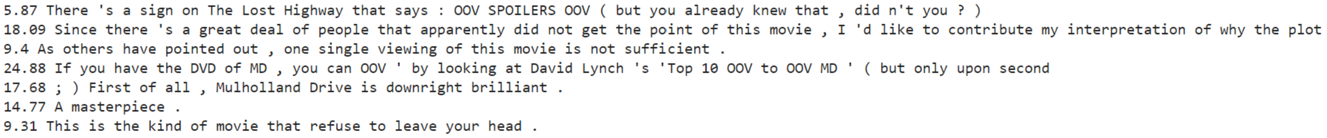


Figure 5: HAN architecture. – Image from [Tixier, 2018]

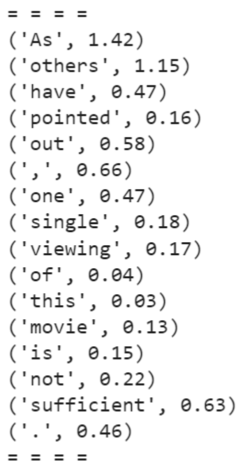
The hierarchical attention is a good illustration of how self-attention might be used in a Hierarchical Attention Network. As we can see on Fig. 5 the mechanism is used twice, at the word level to encode sentences (see on Fig. 7) and then at the sentence level to encode reviews as we observe on Fig. 6. Such architecture matches the natural structure of a review from the most granular level (the words) to the whole review via the sentences. It is a brilliant architecture since the model first determine the attention paid to the words in each sentence, and then to the sentence over the entire review. Fig. 8 presents the attention coefficient with colors for a specific review.



```
5.87 There 's a sign on The Lost Highway that says : OOV SPOILERS OOV ( but you already knew that , did n't you ? )
18.09 Since there 's a great deal of people that apparently did not get the point of this movie , I 'd like to contribute my interpretation of why the plot
9.4 As others have pointed out , one single viewing of this movie is not sufficient .
24.88 If you have the DVD of MD , you can OOV ' by looking at David Lynch 's 'Top 10 OOV to OOV MD ' ( but only upon second
17.68 ; ) First of all , Mulholland Drive is downright brilliant .
14.77 A masterpiece .
9.31 This is the kind of movie that refuse to leave your head .
```

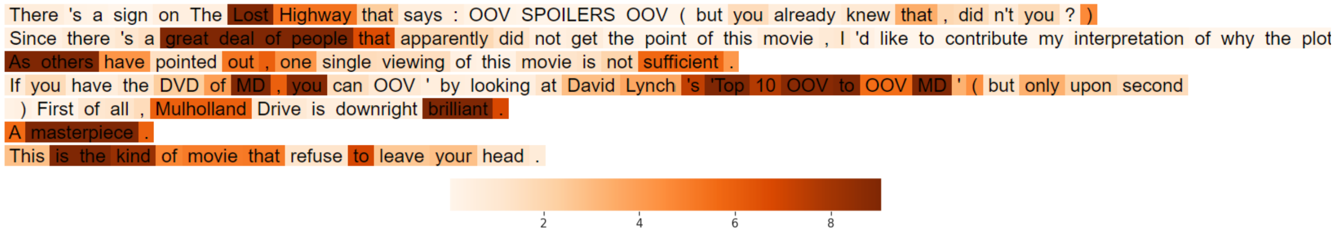
Figure 6: Attention coefficient (at the sentence level) in front of each sentence in a review.

More specifically, the broad architecture can be divided into four symbolic sub-processes that works recursively. To be precise, for each review of the training (respectively of the validation set), the said sub-processes are applied word after word, from the beginning of each sentence to the end, and from the end to the beginning (i.e., bi-directional information processing). In addition, the



```
= = = =
('As', 1.42)
('others', 1.15)
('have', 0.47)
('pointed', 0.16)
('out', 0.58)
(',', 0.66)
('one', 0.47)
('single', 0.18)
('viewing', 0.17)
('of', 0.04)
('this', 0.03)
('movie', 0.13)
('is', 0.15)
('not', 0.22)
('sufficient', 0.63)
('.', 0.46)
= = = =
```

Figure 7: Attention coefficient (at the word level) over each word in a specific sentence of the same review.



There 's a sign on The Lost Highway that says : OOV SPOILERS OOV (but you already knew that , did n't you ?)
Since there 's a great deal of people that apparently did not get the point of this movie , I 'd like to contribute my interpretation of why the plot
As others have pointed out , one single viewing of this movie is not sufficient .
If you have the DVD of MD , you can OOV ' by looking at David Lynch 's 'Top 10 OOV to OOV MD ' (but only upon second
) First of all , Mulholland Drive is downright brilliant .
A masterpiece .
This is the kind of movie that refuse to leave your head .

Figure 8: Representation of attention coefficients for a specific review. The review is classified as ‘positive’ and the model file it correctly.

flow of information is regulated via three main sets of functions related to the reset, update and interpolation gates respectively (remind the GRU design, see Fig. 3). Finally, this architecture is replicated at the sentence level, taking the weights learnt from the word-to-word processing as inputs (hence the name of Hierarchical Attention Network). Thus, adding the design engineering of the network architecture with the Attention Mechanism to this model yields the attention-based bi-directional gated recurrent unit hierarchical network. Considering the architecture through which data flows and is being processed, we now introduce the inputs and ouptut going in and out of the neural network. Namely, the concepts of information memory h_{t-1} and new information x_t respectively describes:

- the vectors of the convoluted sigmoïd-transformed sum of weighted words or sentences (in their vector forms) that have survived the previous flow of gates at time $t-1$.
- the vectors of words or sentences (in their vector forms) entering the network at time t.

Each component is assigned to a weight and a constant that are both randomly initialized, and updated by Stochastic Gradient Descent until the last epoch. Hence, the four sub-processes are fed with such linear combinations of new information and information memory, and respectively work at:

- Filtering out what should be kept from both past the information memory h_{t-1} and the new information x_t by learning the weights of both components, along with their respective constants.

- Filtering out what should be discarded from both past the information memory h_{t-1} and the new information x_t by learning the weights of both components, along with their respective constants.

- A two-fold sub-process. On the one hand, the process aims at revealing the temporal independence (if any) between x_t and h_{t-1} through a weighted sum i.e., the updated state of the system at time t, net of redundancies. This learning relates to the keeping of relevant information. On the other hand, the given sub-process ought to learns what should be discarded from x_t, as well as what has been discarded from h_{t-1} in $t-1$. This second learning relates to the discarding of irrelevant information. Together, the first and second folds of this sub-process work at extracting the temporal dependency between kept and discarded information by convoluting them at the gate. Of course, each version of x_t and h_{t-1} that are duplicated throughout the network is weighted differently at each computation step.

- A second two-fold sub-process aiming at learning the temporal dependency between the first fold of the previous sub-process, and the full memory h_{t-1} by convoluting the two at the end of the gate.

Through this process, the network retains the words that are weighted the further away from '0', and that are the most important to infer sentiment-related information, at the sentence scale. In particular, a softmax activation function maps the bi-directional GRU ouptut i.e. a tahn, with a word-context vector. Each mapped pair of elements is added to form a sentence vector. Rich of this learning, the network takes the sentence vectors with the most important words as inputs for the sentence-level part of the design. One vector after the other, each review of the training (respectively of the validation) set is processed and optimally hardly-labelled so that the error and the non-linear objective loss function are minimized via back-propagation.

5.2 Predictions and limitations

This research work aims at presenting an example of how useful self-attention can be when analyzing text to make predictions. In this case, encoding reviews with the self-attention mechanism allowed the model to capture the idea that two instances of the same words might be important in a specific sentence while not when found in another one. The hierarchical attention provided by our model (at the word level and then at the sentence level) allowed us to classify movie reviews. We trained our HAN on 25000 reviews and tested the model built on 25000 other reviews. Our code is available at `https://colab.research.google.com/drive/11G_QiKJpNGxx1KuDHttSgPJAHe45GqK9?usp=sharing`.

After the training, the model reaches $\approx 85\%$ accuracy in 2 epochs so that we added an early stopping to avoid overfitting. We obtain good results see Fig. 9. This graph visualizes the trade-off between true positive rate and false positive rate. The higher true positive and the lower false positive are, the better. Hence the more top-left-side is the curve, the better is our classifier hierarchical model.

We chose to classify a review as 'positive' when the probability of the review to be in favor of the movie was strictly superior to 0.5. We did not try to enhance our predictions based on this threshold to avoid any kind of overfitting. Hence the confusion matrix we obtained (Fig. 10), leading to a F1-score of approximately 0.82 which corresponds to the harmonic mean of precision and recall (rate of correct positive predictions and rate of correctly predicted positives respectively) focusing on the positive predictions.

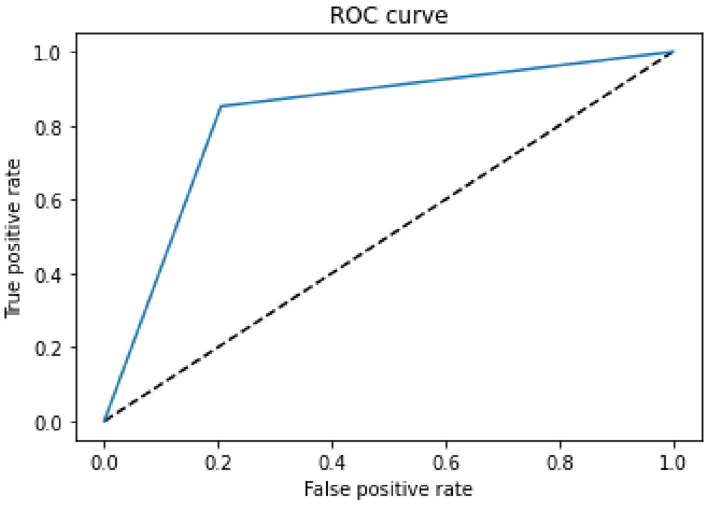


Figure 9: ROC AUC chart.

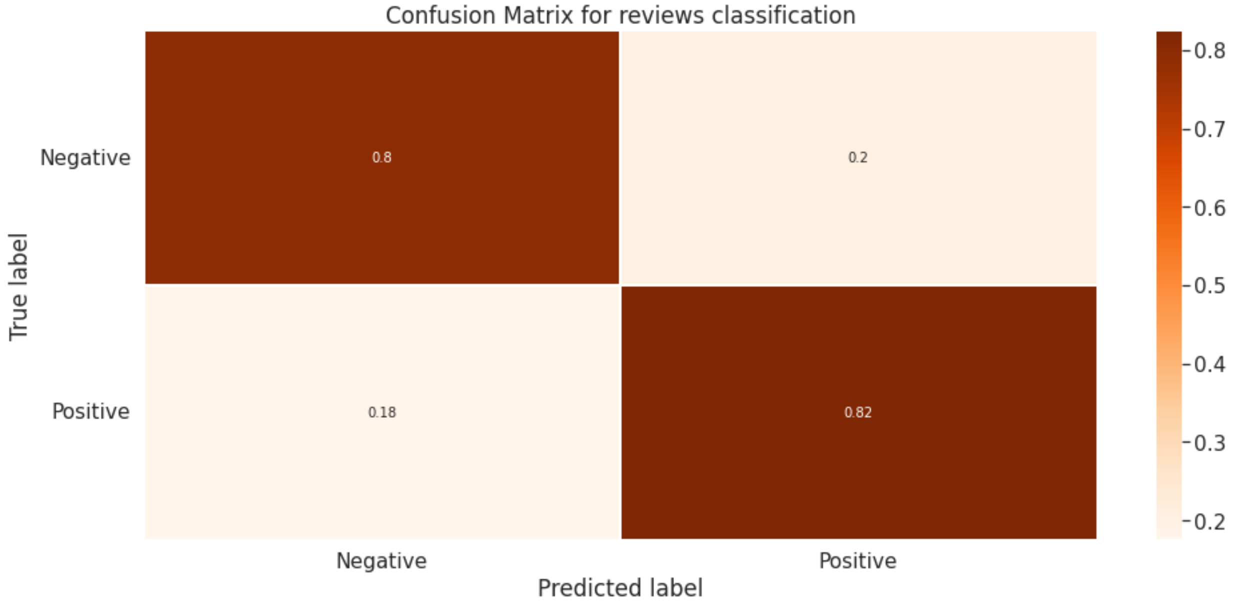


Figure 10: Confusion matrix obtained from our predictions on the test dataset of 25000 reviews.

While our model leads good reviews classification, it presents some limitations, Remy et al. [2019]. The HAN mechanism encode each sentence independently from the others, as a consequence, if an important word is repeated in different sentences and catch all the attention on it (high weight), the process will not be able to focus elsewhere. This is an important limitation because it is at the first level that weights are assigned to words in each sentence; since then, even if some context is identified at the next level, because the importance scores are already set, the process cannot prevent redundancy coming from the lack of dependencies at the first step. To encompass this problem Lin et al. [2017] forced each weight vector to focus on one specific word of any sentence by subtracting an identity matrix to the weight matrix in order to point distinct words designated by diagonal coefficient (more details in the paper). With their method they found a way of associating one specific important group to each weight vector built.

6 Conclusion

Rather than outperforming literature results we aimed to understand, reproduce and illustrate a process for sentiment analysis on movies reviews in the objective of classifying them as positive or negative.

We investigated hierarchical attention mechanism for this project in order to classify reviews on a dataset from the Internet Movies Database (IMDb). We built a Hierarchical Attentional Network (HAN) with a self-attention mechanism to process the data and make the classification thanks to a Recurrent Neural Network (RNN) with Gated Reccurent Units (GRU).

The hierarchical structure of HAN perfectly mirrors the natural and hierarchical architecture of documents (from words to sentences to paragraphs and documents). The process of such network allows efficient understanding of the sentiments. Indeed once the model built we applied it on 25000 reviews from the IMDb to train and we went through the black box to understand the extend to which attention coefficients might help predicting positiveness of negativeness of the movies reviews. We applied the trained network on 25000 other reviews and made predictions based on the probability of being a favourable or an unfavourable review returned by the model.

While we did not compare our results to the literature we visualized the attention coefficients produced over words and sentences illustrating the importance the network gave to each element from the reviews. It reveals that in most sentences the model selects qualitatively informative group of words, highlighting the way it understands reviews and then returns predictions of their ‘level of positiveness’.

7 References

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Gangwar, A. and Vadlamani, R. (2020). A novel bgcapsule network for text classification.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Remy, J.-B., Tixier, A. J.-P., and Vazirgiannis, M. (2019). Bidirectional context-aware hierarchical attention network for document understanding.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization.

Tixier, A. J. P. (2018). Notes on deep learning for nlp.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification.