González-Barba, JÁ.; Segarra Soriano, E.; García-Granada, F.; Sanchís Arnal, E.; Hurtado Oliver, LF. (2020). Extractive summarization using siamese hierarchical transformer encoders. Journal of Intelligent & Fuzzy Systems. 39(2):2409-2419. https://doi.org/10.3233/JIFS-179901

Undefined 1 (2009) 1–5
IOS Press



# Extractive Summarization using Siamese Hierarchical Transformer Encoders

José Ángel González *, Encarna Segarra, Fernando García-Granada, Emilio Sanchis and Lluís-F. Hurtado

*VRAIN: Valencian Research Institute for Artificial Intelligence*
*Universitat Politècnica de València*
*Camí de Vera sn, 46022, València, Spain*
*E-mail: {jogonba2, esegarra, fgarcia, esanchis, lhurtado}@dsic.upv.es*

**Abstract.**
In this paper, we present an extractive approach to document summarization, the Siamese Hierarchical Transformer Encoders system, that is based on the use of siamese neural networks and the transformer encoders which are extended in a hierarchical way. The system, trained for binary classification, is able to assign attention scores to each sentence in the document. These scores are used to select the most relevant sentences to build the summary. The main novelty of our proposal is the use of self-attention mechanisms at sentence level for document summarization, instead of using only attentions at word level. The experimentation carried out using the CNN/DailyMail summarization corpus shows promising results in-line with the state-of-the-art.

Keywords: Siamese Neural Networks, Self Attention, Extractive summarization.

## 1. Introduction

The automatic summarization of textual documents has had an important development in recent years due mainly to two factors: the need to provide summaries of the large amount of information available on the web, and the success of the application of methods based on Neural Networks.

Initial works on automatic summarization were based on unsupervised learning approaches by considering statistical word features [4], topic modeling such as Latent Semantic Analysis [19], graph based approaches such as LexRank [8] and TextRank [15], among others [25] [13]. There are also systems based on supervised learning techniques such as Conditional Random Fields [23] and Support Vector Machines [3]. Modern supervised approaches to single document summarization take advantage of the success of Neural Network architectures and their ability to learn continuous features without the use of preprocessing tools or linguistic annotations [5] [17] [16] [22] [20] [18] [9].

Some recent Neural Network based approaches to automatic summarization incorporate attention mechanisms. In particular, Cheng and Lapata [5] proposed an attentional encoder-decoder approach for extractive single-document summarization and Nallapati, Zhai and Zhou [16] presented an extractive summarization approach, based on a sequential sentence classification problem, by using Neural Networks.

In a previous work [9], the SHA-NN system is proposed. It is a supervised approach to text summarization which is based on Siamese Hierarchical Attention Neural Networks using distributed vector representation of words. Siamese Neural Networks are capable of learning from positive and negative samples. The network is provided with positive and negative document-summary pairs; a positive pair is a document and its summary and a negative pair is a document and a summary of other different document randomly extracted from the training set.

*Corresponding author. E-mail: jogonba2@dsic.upv.es

The siamese network is used as a classifier that decides, given a document and a summary, whether the summary is suitable or not for the document. The model consists of two networks, so that one of them processes the document and the other the summary. Furthermore, this model is enriched with an attention mechanism that provides a score associated to each word and each sentence of the input document. When the siamese network learn that a given summary is suitable for a given document, the most relevant sentences from the document (those with the highest attention scores) lead the classifier to take the right decision. Authors hypothesize that these salient sentences are good candidates to make a extractive summary of the document.

Due to the process of assigning scores to document sentences of the SHA-NN system is based on the attention mechanisms, then the capacity of these mechanisms plays a crucial role. The greater the capacity of these attention mechanisms to capture complex relationships among different sentences, the better the SHA-NN system will be extracting the most salient sentences to build the summaries. Moreover, the SHA-NN system, as most of the recent extractive systems, rely on recurrent neural networks to derive a semantic representation of the document.

Recently, the attention mechanisms have been developed in such a way that they completely replace convolutional and recurrent methods through self-attention mechanisms proposed as part of the so called Transformer models [26], improving the state of the art in several tasks such as Machine Translation [26], Question Answering [7], Automatic Summarization [12], as well as the self-attention mechanisms by itself on tasks like Sentiment Analysis [1].

These self-attention mechanisms compute word representations by relating different positions of the words in a sentence. Concretely, to compute the representation for a given word, the self-attention compares it to every other word in the sentence. The result of these comparisons is an attention score for every other word in the sentence that determines how much each of the other words should contribute to the representation of the given word, capturing complex relationships between words in sentences such as anaphora, co-reference, coherence and lexical cohesion [28] [24]. Therefore, it seems interesting to incorporate these attention mechanisms in the SHA-NN framework (both at word and sentence level), in order to extract better representations and scores for each sentence in a given document.

Until now, only the ability of transformers to capture word level relationships has been explored. However, these models have not been previously experimented to integrate sentence level relationships in a hierarchical way from the relationships captured at word level. In this paper we propose to extend the transformers in a hierarchical way to also work at sentence level. This way, the model could explain relationships among document sentences such as co-reference and paraphrasing.

Deep learning models require to adjust millions of parameters, therefore, large size corpora are needed in order to train them. An important resource for data-driven models is the CNN/DailyMail summarization corpus, originally constructed by [10] for the passage-based question answering task, and adapted for the single document summarization task [5] [17]. It consists of news articles from CNN and DailyMail and contains 312,085 document-summary pairs. This corpus has been widely used by recent works on automatic summarization and we used it in this work in order to make a fair comparison.

In this work, we propose a new extractive summarization system, the Siamese Hierarchical Transformer Encoders (SHTE) system, that is based on two main contributions. First, the integration of the transformer encoders in the classifier based on siamese networks for automatic summarization, then, allowing for learning the sentence representations and assigning sentence scores. Second, the extension of the transformer encoders in order to apply them in a hierarchical way on documents. Some experiments on the CNN/DailyMail corpus were performed that show that the proposed approach is adequate for the single-document extractive summarization problem.

## 2. System Description

Our system addresses an intermediate binary classification problem, which consists in determining positive and negative pairs of documents and summaries ($X \in \mathbb{X},\ Y \in \mathbb{Y}$), in order to learn rich semantic representations and attention distributions over sentences that are useful to extract relevant sentences to compose extractive summaries.

The proposed system is based on siamese networks to distinguish correct summaries for documents. However, differently from the SHA-NN system, the encoders used both at word and sentence level are replaced by self-attention mechanisms. Additionally to

extract representations, these attentions also can be used to assign scores to each word and sentence, based on the different relationships learned by them. More concretelly, we used as self-attention mechanism the encoder proposed in [26] for the Transformer, applied in a hierarchical way, both at word and sentence level, similarly to [27]. In Figure 1 the architecture of our system is shown.

### 2.1. Word Level

Let $X = \{\overbrace{\{w_{11}, ..., w_{1W}\}}^{Y_1}, ..., \overbrace{\{w_{T1}, ..., w_{TW}\}}^{Y_R}\}^{X_1 \quad X_T}$ and $Y = \{\overbrace{\{v_{11}, ..., v_{1V}\}}^{Y_1}, ..., \overbrace{\{v_{R1}, ..., v_{RV}\}}^{Y_R}\}$ be the input document and the input summary respectively, where $w_{ij}$ is the word $j$ in the sentence $i$ of the document $X$ and $v_{ij}$ is the word $j$ in the sentence $i$ of the summary $Y$. $W$ and $V$ are the maximum number of words per sentence in document and summary, and $T$ and $R$ are the maximum number of sentences in document and summary.

First of all, the words from document and summary are embedded by a $d_e$ dimensional embedding matrix $E$, shared among the two branches of the siamese network, due to the words in both sequences come from the same source of information (domain, language, etc.). We added these representations with positional encodings matrices, both for document sentences $P^x \in \mathbb{R}^{W \times d_e}$ and summary sentences $P^y \in \mathbb{R}^{V \times d_e}$, based on sine and cosine functions as in [26]. This is with the aim of allowing the model to explain temporal relationships among the words inside each sentence. Concretely, we sum the corresponding $P$ matrix with the sequence of embeddings that represents each sentence, both for document $X^0 = \{\overbrace{P^x + E(X_1)}^{Y_1^0}, ..., \overbrace{P^x + E(X_T)}^{Y_R^0}\}^{X_1^0 \quad X_T^0}$ and summary $Y^0 = \{\overbrace{P^y + E(Y_1)}^{Y_1^0}, ..., \overbrace{P^y + E(Y_R)}^{Y_R^0}\}$, where $E(X_i) \in \mathbb{R}^{W \times d_e}$ is the word embedding sequence of the sentence $i$ in the document and $E(Y_i) \in \mathbb{R}^{V \times d_e}$ is the word embedding sequence of the sentence $i$ in the summary.

With $X^0$ and $Y^0$ as input, each network computes the sentence representations in the same way for the document and the summary (the left branch for processing $X^0$ and the right branch for processing $Y^0$). These representations are computed by means of an encoder of $N$ layers, relying on multi-head scaled dot-product attention as defined in Equations 1, 2 and 3.

$$MultiHead(A, B, C) = [head_1; ...; head_h]W^O \quad (1)$$

$$head_i = Attention(AW_i^Q, BW_i^K, CW_i^V) \quad (2)$$

$$Attention(Q, K, V) = softmax(\frac{QK^\intercal}{\sqrt{d_k}})V \quad (3)$$

Where $h$ is the number of heads, $W_i^Q \in \mathbb{R}^{d_e \times d_k}$, $W_i^K \in \mathbb{R}^{d_e \times d_k}$, $W_i^V \in \mathbb{R}^{d_e \times d_k}$, and $W^O \in \mathbb{R}^{h \times d_k \times d_e}$ are the projection matrices for Query ($Q$), Key ($K$), Value ($V$) of the head $i$, and output of the multi-head attention, all at word level; and $A$, $B$, $C$ are the inputs of the multi-head attention.

Once the multi-head attention is computed, a layer-normalized [2] residual connection is applied. After that, a position wise feed-forward network is applied to each position independently and its outputs are connected with its inputs by means of another layer-normalized residual connection. Finally, with the aim of obtaining a single vector representation for each sentence, pooling is applied on top of the last encoder. Equations from 4 to 11, show the full process to compute the representation of the sentence $i$ for document, $s_i$, and summary, $q_i$, with a $N = 1$ encoder and $X^0$, $Y^0$ as input. We use the superscript $1w$ to refer all the weights and intermediate outputs from the first encoder at word level.

$$M^{1w} = MultiHead(X_i^0, X_i^0, X_i^0) \quad (4)$$

$$\hat{M}^{1w} = MultiHead(Y_i^0, Y_i^0, Y_i^0) \quad (5)$$

$$L^{1w} = LayerNorm(X_i^0 + M^{1w}) \quad (6)$$

$$\hat{L}^{1w} = LayerNorm(Y_i^0 + \hat{M}^{1w}) \quad (7)$$

$$F^{1w} = max(0, L^{1w}W_1^{1w} + b_1^{1w})W_2^{1w} + b_2^{1w} \quad (8)$$

$$\hat{F}^{1w} = max(0, \hat{L}^{1w}W_1^{1w} + b_1^{1w})W_2^{1w} + b_2^{1w} \quad (9)$$

$$s_i = Pooling(LayerNorm(L^{1w} + F^{1w})) \quad (10)$$

$$q_i = Pooling(LayerNorm(\hat{L}^{1w} + \hat{F}^{1w})) \quad (11)$$

Where $M^{1w}, L^{1w}, F^{1w} \in \mathbb{R}^{W \times d_e}$ are the intermediate outputs from the document branch and $\hat{M}^{1w}$, $\hat{L}^{1w}$, $\hat{F}^{1w} \in \mathbb{R}^{V \times d_e}$ are the same for the summary branch, $W_1^{1w} \in \mathbb{R}^{d_e \times d_{ffw}}$, $W_2^{1w} \in \mathbb{R}^{d_{ffw} \times d_e}$ are the weights of the position wise feed-forward network, and $s_i \in \mathbb{R}^{d_e}$, $q_i \in \mathbb{R}^{d_e}$ are the representations of the sentence $i$ for document and summary respectively. As it can be noted, all the weights are shared between the two branches of our siamese network.

Then, this process is applied independently to each matrix that represents the word embeddings sequence of each sentence for document and summary. Its outputs $\{s_i : 1 \leq i \leq T\}$ for document and $\{q_i : 1 \leq i \leq R\}$ for summary, are the inputs to the sentence level that computes a representation of documents and summaries based on their sentences.

### 2.2. Sentence Level

From the representations obtained after $N$ word-level encoders, both for document $S = \{s_1, ..., s_T\} \in \mathbb{R}^{T \times d_e}$ and summary $Q = \{q_1, ..., q_T\} \in \mathbb{R}^{R \times d_e}$, positional encodings, in the same way that at word level, are added to them, in order to take into account temporal relationships among the sentences of documents. Let $P^s \in \mathbb{R}^{T \times d_e}$ and $P^q \in \mathbb{R}^{R \times d_e}$ be the positional encoding matrices for document and summary, the input to the first encoder of the sentence level are $S^0 = P^s + S$ and summary $Q^0 = P^q + Q$.

The representations of document $r$ and summary $p$, by using a $\hat{N} = 1$ encoder and $S^0$, $Q^0$ as input, are obtained as shown from Equations 12 to 19. We use the superscript $1s$ to refer all the weights and intermediate outputs from the first encoder at sentence level.

$$M^{1s} = MultiHead(S^0, S^0, S^0) \quad (12)$$

$$\hat{M}^{1s} = MultiHead(Q^0, Q^0, Q^0) \quad (13)$$

$$L^{1s} = LayerNorm(S^0 + M^{1s}) \quad (14)$$

$$\hat{L}^{1s} = LayerNorm(Q^0 + \hat{M}^{1s}) \quad (15)$$

$$F^{1s} = max(0, L^{1s}W_1^{1s} + b_1^{1s})W_2^{1s} + b_2^{1s} \quad (16)$$

$$\hat{F}^{1s} = max(0, \hat{L}^{1s}W_1^{1s} + b_1^{1s})W_2^{1s} + b_2^{1s} \quad (17)$$

$$r = Pooling(LayerNorm(L^{1s} + F^{1s})) \quad (18)$$

$$p = Pooling(LayerNorm(\hat{L}^{1s} + \hat{F}^{1s})) \quad (19)$$

Where $M^{1s}, L^{1s}, F^{1s} \in \mathbb{R}^{T \times d_e}$ are the intermediate outputs from the document branch and $\hat{M}^{1s}$, $\hat{L}^{1s}$, $\hat{F}^{1s} \in \mathbb{R}^{R \times d_e}$ are the same for the summary branch, $W_1^{1s} \in \mathbb{R}^{d_e \times d_{ffs}}$, $W_2^{1s} \in \mathbb{R}^{d_{ffs} \times d_e}$ are the weights of the position wise feed-forward network, and $r \in \mathbb{R}^{d_e}$, $p \in \mathbb{R}^{d_e}$ are the representations of document and summary respectively. In this level, all the weights are also shared between the two branches of the network.

### 2.3. Classification

From the representations $r$ and $p$, the interaction between them is computed as their concatenation with their absolute difference, following [6]. This interaction is used as input for a single-layer feed-forward network whose output is a probability distribution over

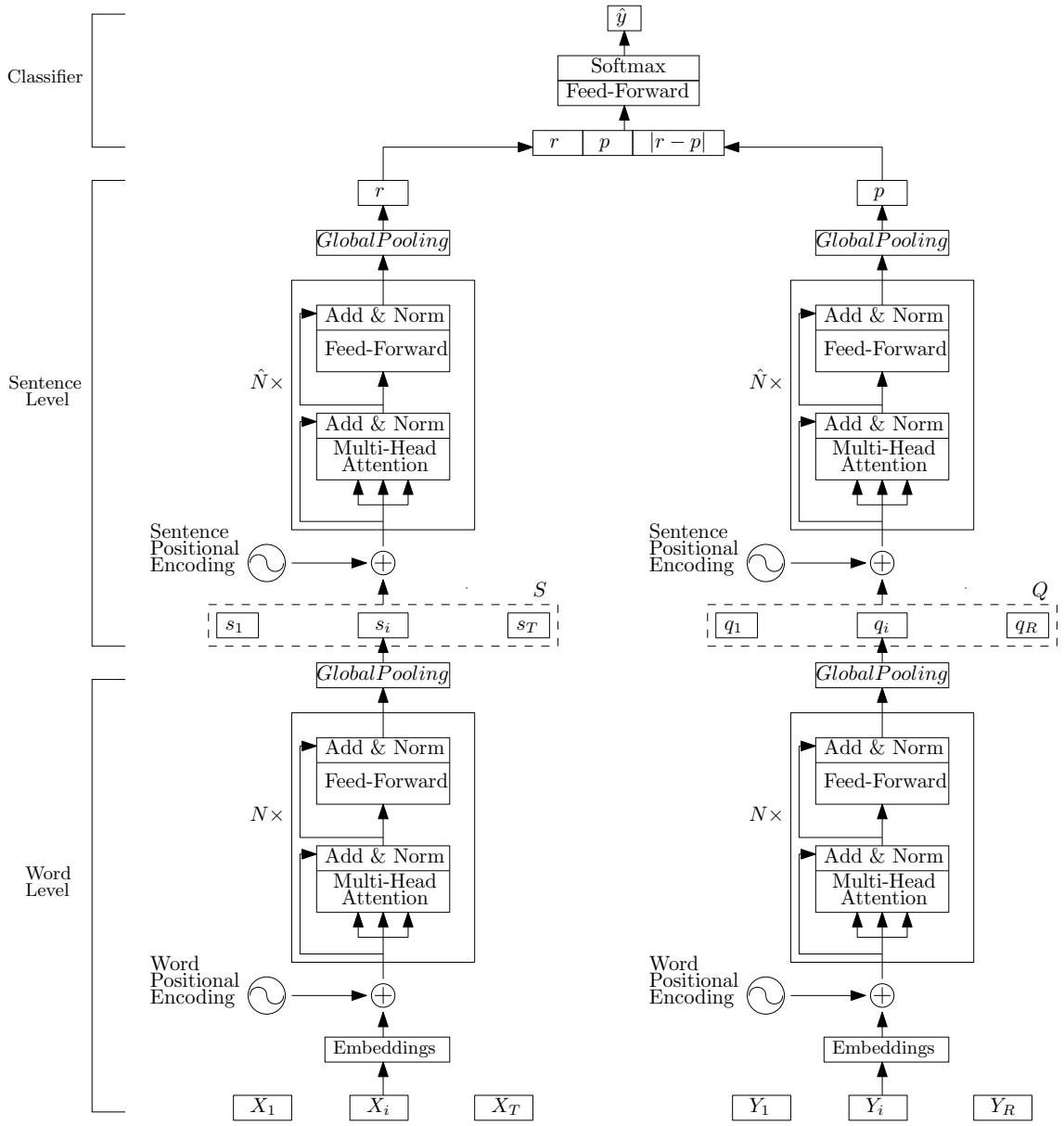


Fig. 1. Architecture of HTE system where the left branch processes documents and the right branch processes summaries.

$\mathbb{C} = \{0, 1\}$ where $y = 1$ is used as ground truth for positive pairs and $y = 0$ for negative pairs.

$$\hat{y} = softmax(W_2^{\hat{y}}(max(0, W_1^{\hat{y}}[r; p; |r-p|]+b_1^{\hat{y}}))+b_2^{\hat{y}}) \tag{20}$$

Where $W_1^{\hat{y}} \in \mathbb{R}^{3\times d_e \times d_h}$ are the weights to project the interaction between document and summary, and $W_2^{\hat{y}} \in \mathbb{R}^{2\times d_h}$ are the weights of the output layer whose outputs are transformed into a probability distribution over $\mathbb{C} = \{0, 1\}$.

In order to train the model, for each document we build one positive pair $(X, Y)$, provided by the corpus, and one negative pair $(X, Y') : Y' \neq Y$ where $Y'$ is randomly chosen from the summaries of the remaining documents.

### 2.4. Sentence Scoring

In order to select the most relevant sentences of a document to build a summary, a score have to be assigned to each sentence based on some criterion. In this work, we consider that a sentence is more relevant the more semantics from a document it captures. In our proposal, it can be explained as those sentences which are more attended by the other sentences of the document. Moreover, we assume that the higher the level of representation, the better this knowledge will be captured, for that reason, in this work we only use the attentions of the last encoder at sentence level.

Note that this proposal is different from SHA-NN [9], where the attention outputs are directly these scores and are used during training to control the contribution of each sentence in the computation of a semantic representation of a document, which is useful to distinguish correct summaries for it. In this work, the attentions are a mechanism of the model to learn sentence representations, but also, it can be used to compute sentence scores.

Then, with the aim of building a ranking over document sentences by means of our hierarchical self-attention model, we use the attention matrices at sentence level, obtained after a forward pass on the left branch of the network with an input document, following Equations from 21 to 23.

$$G_i = softmax(\frac{Q_i K_i^\intercal}{\sqrt{d_k}}) \tag{21}$$

$$H = \frac{1}{h}\sum_{i=0}^{h} G_i \tag{22}$$

$$\alpha = \frac{1}{T}\sum_{i=0}^{T} H_i \tag{23}$$

Where $Q_i, K_i \in \mathbb{R}^{T\times d_k}$ are the Queries and Keys in head $i$, $G_i \in \mathbb{R}^{T\times T}$ is the attention matrix of head $i$, $H \in \mathbb{R}^{T\times T}$ is the averaged attention of all the headers, and $\alpha \in \mathbb{R}^{T}$ is the vector that contains the final score assigned to each sentence.

The system disposes of $h$ different attentions that explain different relationships among the sentences, which are unknown a priori. As it is shown in Equation 22, we consider that all the relationships captured by the self-attention mechanism are equally relevant to obtain a score. For this reason, the most attended sentences, in average among the different relationships (attentions), must be more relevant.

After computing the average attention of all the heads, $H$, the component $H_{ij}$ represents the average attention that the model assigns to the sentence $j$ when it is processing the sentence $i$. Then, it could be used to compute the relevance of a sentence $j$ in the document based on the average attention that $j$ have in all the other sentences of the document. This value is the assigned score to each sentence, see Equation 23, and it is used to rank them and to select the $k$ most relevant sentences to compose the summary.

## 3. Corpus

The CNN/DailyMail[1] corpus was used in this work. This corpus, which is a combination from articles of the news websites CNN and DailyMail, was originally constructed by [10] for Question Answering and modified by [5] and [17] for both abstractive and extractive summarization. The CNN/DailyMail corpus had been partitioned into 287,227 training document-summary pairs, 13,368 validation document-summary pairs and 11,490 test document-summary pairs. Some corpus characteristics are presented in Table 1.

Table 1
Average number of sentences and words (including words per sentence) of the corpus.

| | Sents | Words | Words/Sent |
|---|---|---|---|
| Train Documents | 28.2 | 765.4 | 27.1 |
| Train Summaries | 3.8 | 53.4 | 14.1 |
| Dev Documents | 26.6 | 749.9 | 28.2 |
| Dev Summaries | 4.2 | 59.1 | 14.3 |
| Test Documents | 26.9 | 758.9 | 28.2 |
| Test Summaries | 3.9 | 56.0 | 14.3 |

In order to compare this work to other works [16] and [9], we used in the experiments the entity-anonymized version of this corpus, where entity occurrences are replaced with document-specific integer-ids, thereby reducing the vocabulary size. It should be noted that the ground truth summaries provided by this

[1] https://cs.nyu.edu/~kcho/DMQA/

corpus are abstractive, and they were constructed by concatenation of the highlights associated to the documents.

## 4. Related Work

In this Section we describe different approaches to summarization, in particular, those systems used in the experimental comparison and systems which are similar to our SHTE system. One of the most known of the first approaches to automatic extractive summarization is TextRank [15], which is based on the PageRank algorithm in order to extract the most relevant sentences. In TexRank, sentences which are vertices of the graph, are interconnected among them by following some criterion based on their similarity.

Moreover, there are some systems designed specifically for specific domains. This is the case of the Lead system, which is based on extracting the first $k$ sentences of the documents to make a summary. Although it seems naive, it is specially robust when it is applied on articles of newspapers, generally due to in this domain, the first sentences are dedicated to condense the information of all the document and are used to call the attention from the reader.

Recently, due to the increasing popularity of the Neural Networks, a large number of extractive approaches based on Deep Learning have been proposed. The first systems based on these techniques were [5] and [16]. Concretely, Cheng and Lapata in [5] proposed an attentional encoder-decoder approach for extractive single-document summarization. In [16], Nallapati, Zhai and Zhou presented two versions of Hierarchical Attention Networks to choose sentences from the document as a binary sequence classification problem. One of these versions, SummaRunner-Abs, is trained using directly the samples provided by the corpus. The other version, SummaRunner-Ext, requires a greedy algorithm to prepare the corpus in an usable way for training the system, choosing as reference the set of sentences from the document that maximize the similarity with respect to the abstractive summary.

Additionally, in [9], the SHA-NN system, based on addressing a binary classification problem in order to select the most relevant sentences by means of the attention mechanisms, was proposed. This system, differently from the previous mentioned works, does not require the preparation of the corpus, being the system which learns that alignment, moreover, it addresses the problem as a binary classification task, instead of performing sequence classification.

All these works are experimented on the anonymized version of the CNN/DailyMail, however, there are also many other works which use the non-anonymized version of the corpus, such as [22], where a hybrid abstractive-extractive system was proposed, or [18] which explores the use of Reinforcement Learning on extractive summarization.

Our SHTE system is similar to SHA-NN enriching the attention mechanisms with new self-attention mechanisms proposed in the encoder part of the Transformer architecture [26]. This kind of attention mechanisms have been recently used for abstractive multi-document summarization [12]. In this work, a modification of the Transformer Decoder was used to abstractively generate the first section of Wikipedia articles, based on salient information extracted from non-Wikipedia documents, by means of extractive summarization systems. However, to our knowledge, the extractive summarization by using the self-attention of the Transformer encoders have not been explored and it is interesting to integrate them in the binary classification framework for extractive summarization.

## 5. Experiments

In order to carry out the experimentation, we used randomly initialized word embeddings with $d_e = 128$ which are trained along with the model. Also, some hyperparameters of the model were fixed, such as $N = 2$ word encoders and $\hat{N} = 2$ sentences encoders, $h = 6$ heads, $d_k = d_v = d_q = 64$, $d_{ffw} = d_{ffs} = 128$ and $d_h = 512$.

In order to train our model, we used batches of 128 document-summary pairs, 64 positives and 64 negatives. Adam was used as update rule with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize the cross entropy, and Noam was used as learning rate scheduler with $warmup\_steps$ = 4000. The model was trained during 20 epochs of 5000 batches (640.000 examples, 2 times the size of the training set), and after finish, the weights of the best model until that epoch (model that minimized the cross-entropy on the development set) were used to summarize the test samples. In order to compose these summaries, the $k = 3$ most relevant sentences were selected.

The evaluation of the performance of the systems was done by using three variants of the ROUGE measure [11]. Concretely, Rouge-N with unigrams and

Table 2

Experimentation modifying the addition of positional information and the selected attention head to rank the sentences.

| | | Precision | | | Recall | | | $F_1$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Head | R-1 | R-2 | R-L | R-1 | R-2 | R-L | R-1 | R-2 | R-L |
| No Positional | 1 | 24.28 | 7.92 | 21.79 | 45.06 | 15.15 | 40.38 | 29.75 | 9.80 | 26.68 |
| | 2 | 24.58 | 8.11 | 22.13 | 44.15 | 14.90 | 39.64 | 29.89 | 9.92 | 26.88 |
| | 3 | **24.79** | 7.97 | **22.29** | 43.48 | 14.42 | 38.98 | 29.64 | 9.62 | 26.62 |
| | 4 | 24.14 | 7.81 | 21.67 | 44.14 | 14.71 | 39.25 | 29.51 | 9.63 | 26.46 |
| | 5 | 24.49 | 7.94 | 22.02 | 43.40 | 14.39 | 38.90 | 29.61 | 9.66 | 26.58 |
| | 6 | 24.42 | 7.60 | 21.89 | 41.90 | 13.33 | 37.41 | 29.00 | 9.09 | 25.95 |
| | Avg Heads | 24.67 | **8.23** | 22.16 | **45.45** | **15.53** | **40.73** | **30.20** | **10.15** | **27.10** |
| Sent Positional | 1 | 27.79 | 11.07 | 25.21 | 51.31 | 20.78 | 47.34 | 34.76 | 13.82 | 31.51 |
| | 2 | 27.17 | 10.66 | 24.62 | 52.36 | 20.67 | 47.38 | 34.29 | 13.47 | 31.06 |
| | 3 | 29.19 | 11.71 | 26.53 | 51.74 | 20.86 | 46.98 | 35.83 | 14.39 | 32.55 |
| | 4 | **29.84** | **12.09** | **27.15** | 52.17 | 21.24 | 47.41 | 36.15 | 14.58 | 33.16 |
| | 5 | 29.12 | 11.87 | 26.48 | **53.09** | **21.66** | **48.19** | 36.03 | 14.68 | 32.74 |
| | 6 | 29.60 | 12.01 | 26.91 | 52.30 | 21.30 | 47.45 | 36.21 | 14.73 | 32.99 |
| | Avg Heads | 29.64 | 12.03 | 26.97 | 52.46 | 21.36 | 47.67 | **36.36** | **14.76** | **33.37** |
| Sent-Word Positional | 1 | 24.68 | 8.12 | 22.13 | 44.20 | 14.70 | 39.59 | 30.11 | 9.94 | 27.03 |
| | 2 | 23.91 | 7.84 | 21.51 | 44.34 | 14.87 | 39.79 | 29.45 | 9.74 | 26.47 |
| | 3 | **25.83** | **9.69** | **23.32** | 50.38 | 18.98 | 45.37 | 32.16 | 11.74 | 28.95 |
| | 4 | 23.59 | 7.66 | 21.18 | 43.99 | 14.61 | 39.39 | 28.98 | 9.48 | 25.98 |
| | 5 | 25.23 | 8.86 | 22.72 | 47.47 | 17.02 | 42.68 | 31.38 | 11.10 | 28.24 |
| | 6 | 23.94 | 7.49 | 21.56 | 39.29 | 12.76 | 35.82 | 28.35 | 8.94 | 25.49 |
| | Avg Heads | 25.33 | 9.42 | 22.84 | **50.92** | **19.02** | **45.85** | **32.40** | **12.04** | **29.18** |

bigrams (Rouge-1 and Rouge-2) and Rouge-L were used. Although in the literature there are some proposals to evaluate automatic summarizations without using the gold standard [21] [14], in order to compare our system to other approaches in the same conditions, we evaluated it with ROUGE statistics using the gold standard provided by the CNN/DailyMail corpora.

We consider two interesting aspects to be analyzed. The first one consists in the impact of the positional information on the selection of the most relevant sentences. Concretely, we explore three ways for the incorporation of positional information: i) just at the sentence level, ii) both at word and sentence level; and iii) without positional information. For the CNN/DailyMail corpus, the first sentences of the documents tend to be the most representative sentences to compose the summary. This is due to the journalistic style, that tries to capture the attention of the reader in the first paragraphs of the articles. For this reason, we expect that the sentence positional information must be specially relevant.

The second aspect to analyze is the strategy of averaging attentions from all the heads of our model in order to rank the sentences. For selecting the relevant sentences, we hypothesize that the combination of all the relationships captured by the different heads is more adequate than individual attention captured by only one head. This is due to it is impossible to distinguish which heads capture relevant relationships for generate good summaries. Then, this lack of knowledge could be countered by averaging all the different relationships. Additionally, it is important to highlight that we only used the attentions of the last encoder at sentence level because the relationships captured at this level are semantically richer than the relationships captured in the first encoder.

The results of the experimentation are shown in Table 2, where three blocks of experiments were done varying the positional information (no positional, at sentence level, and both at word and sentence level). The column labelled as "Head" represents what head was used to assign the scores (only one head or all heads). From this table, on the one hand, it is interesting to observe that the addition of positional information only at sentence level is more informative than its combination with positional information both at word

and sentence level. The improvements obtained by adding positional information on the sentences seem to support the assumption of the importance of the sentence order in the generation of the summaries. Moreover, both of them provides better results than not using positional information.

On the other hand, the strategy of averaging the attention heads is the best mechanism for sentence scoring in almost all the cases. Concretely, it obtains always the best results in terms of $F_1$ and it seems to have worse results in terms of Precision. Although the improvements are not statistically significant, it is possible to see that there are heads which capture less relevant relationships than others and the averaging of them with the remaining heads counters these low results.

In Table 3, the results of SHTE system as well as the results of Lead-3, TextRank and Random-3 systems are shown (†). The Table also contains the results of the SHA-NN [9], SummaRunner-Abs and SummaRunner-Ext [16] systems (‡). All these results have been obtained on the anonymized version of CNN/DailyMail corpus. We also included the results of the Lead-3 system presented in [16] (‡) since they are different from ours and the authors do not provided enough information to reproduce the experimentation.

It is possible to see that the Lead-3 system outperforms the SHTE system in our experimentation. The same happens with SHA-NN and SummaRunner-Abs systems (the most similar to our system). This supports the assumption that the first sentences of the documents are more relevant than the remaining sentences in this news articles domain. Moreover, the SHTE system obtains better relative results with respect to Lead-3 (SHTE† vs Lead-3†) than SummarRunner-Abs (SummaRunner-Abs‡ vs Lead-3‡).

Table 3
Results in terms of full length Rouge $F_1$

| System | R-1 | R-2 | R-L |
|---|---|---|---|
| SHTE† | 36.4 | 14.8 | 33.4 |
| Lead-3† | **37.3** | **15.1** | **34.0** |
| TextRank† | 29.4 | 10.1 | 26.3 |
| Random-3† | 26.7 | 7.3 | 23.9 |
| SHA-NN‡ | 35.4 | 14.7 | 33.2 |
| Lead-3‡ | 39.2 | 15.7 | **35.5** |
| SummaRunner-Abs‡ | 37.5 | 14.5 | 33.4 |
| SummaRunner-Ext‡ | **39.6** | **16.2** | 35.3 |

*Document:* **police are desperately searching for a teenage schoolgirl who went missing near @entity2 on friday afternoon . @entity3 , 14 , was last seen leaving her school in @entity5 around 3pm , according to @entity0 .** she was described by police as caucasian , 150cm tall , with blue eyes and brown shoulder length hair . **@entity3 , 14 , was last seen leaving her school in @entity5 around 3pm , according to @entity0 police confirmed she was still missing early on saturday morning , and were concerned for ms @entity10 because of her age .** her family and police have appealed for anyone who has seen ms @entity10 to immediately contact she was last seen wearing a blue jeans , black jumper and black shoes . police confirmed she was still missing early on saturday morning , and were concerned for ms @entity10 because of her age .

*Ground Truth:* police worried about teen girl who has been missing since friday afternoon . @entity3 , 14 , was last seen leaving her school near @entity2 . police confirmed she was still missing early on saturday morning . an image of ms @entity10 released in the hopes of finding her quickly .

Fig. 2. Extractive summarization with a test sample of CNN/DailyMail corpus (CNN subset).

Comparing the siamese neural network based approaches, the SHTE system outperforms SHA-NN system in terms of Rouge-1 and obtains slightly better results in terms of R-2 and R-L. Additionally, the transformer encoders of the SHTE system present less training and inference time than the LSTM encoders of the SHA-NN system since the first one can be parallelized.

Figure 2 shows an example of summarization using the proposed SHTE system. We provide the Document, its Ground Truth summary, and the three sentences extracted by our system (bold font). The figure shows how the system is capable of focusing on the first sentences of the document, but it have the ability to skip the third sentence in order to extract the summary.

Figure 3 shows the averaged self-attention matrix $H$ for all sentences in the example of Figure 2. Sentences are arranged in the matrix following the order of occurrence in document. $H_{ij}$ represents the average attention that the model assigns to the sentence in column $j$ when it is processing the sentence in row $i$. It can be observed that the first sentences have assigned higher relevance due to their columns show darker colors. This example also illustrates the fact that the higher relevance are placed in the lower diagonal of the matrix, showing that the left context of sentences is more important than the right one, that is, when writing a given sentence the writer focus on the past sentences instead of the future ones.

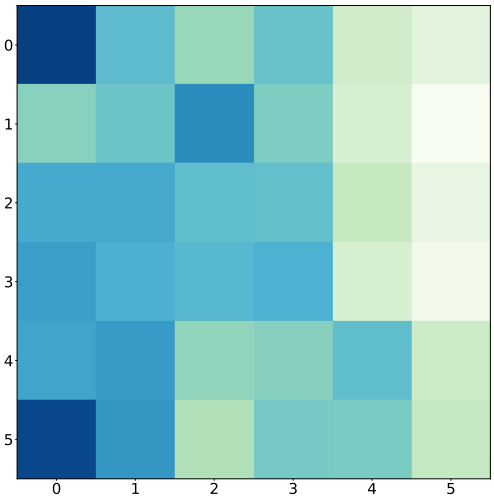


Fig. 3. Averaged self-attention from all heads, for all sentences in the example 2. Darker colors indicate a higher attention value.

## 6. Conclusions

In this work, we propose a new extractive summarization system, the Siamese Hierarchical Transformer Encoders system, that is based on the use of siamese neural networks and the transformer encoders which are extended in a hierarchical way.

Given a document, the SHTE system assigns scores to the document sentences in order to select the most salient sentences to build the summary. It can be considered that a sentence is more relevant when more semantics from the document it captures. In our proposal, these more relevant sentences are those which are more attended by the other sentences of the document. We studied the influence of the positional information into the attentions and the improvements obtained seem to support the assumption of the importance of the sentence order in the generation of the summaries. We also present experiments considering both the use of only one attention head and the average of all the heads. The results show that the strategy of averaging the attention heads is the best mechanism. Finally, we compare our SHTE system to other extractive summarization approaches. The obtained results are in-line with the state-of-the-art on the CNN/DailyMail corpus.

As future work, we will study other strategies in order to compute the scores of the sentences from the attentions differently from the averaging used in this work.

## Acknowledgements

This work has been partially supported by the Spanish MINECO and FEDER founds under project AMIC (TIN2017-85854-C4-2-R). Work of José-Ángel González is also financed by Universitat Politècnica de València under grant PAID-01-17.

## References

[1] A. Ambartsoumian and F. Popowich. Self-attention: A better building block for sentiment analysis neural network classifiers. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 130–139, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics.

[2] J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[3] N. Begum, M. Fattah, and F. Ren. Automatic text summarization using support vector machine. 5:1987–1996, 07 2009.

[4] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[5] J. Cheng and M. Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680, 2017.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[8] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[9] J.-Á. González, E. Segarra, F. García-Granada, E. Sanchis, and L.-F. Hurtado. Siamese hierarchical attention networksfor extractive summarization. *Journal of Intelligent & Fuzzy Systems*, 36(5):4599–4607, 2019.

[10] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[11] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[12] P. J. Liu, M. A. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

[13] E. Lloret and M. Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

[14] A. Louis and A. Nenkova. Automatically assessing machine summary content without a gold standard. *Comput. Linguist.*, 39(2):267–300, June 2013.

[15] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[16] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3075–3081, 2017.

[17] R. Nallapati, B. Zhou, C. N. dos Santos, a. G. Çagl[Takase, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290. ACL, 2016.

[18] S. Narayan, S. B. Cohen, and M. Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[19] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan. Text summarization of turkish texts using latent semantic analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 869–876, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[20] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304, 2017.

[21] H. Saggion, J.-M. Torres-Moreno, I. d. Cunha, and E. SanJuan. Multilingual summarization evaluation without human models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1059–1067, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[22] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics, 2017.

[23] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2862–2867, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[24] D. Stojanovski and A. Fraser. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics.

[25] G. Tur and R. De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[27] 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. Association for Computational Linguistics, 2016.

[28] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.