This version of the paper is as published in ACL's proceedings. There is an errata for the question answering results that can be found here: http://homepages.inf.ed.ac.uk/scohen/acl18external-errata.pdf.

Document Modeling with External Attention for Sentence Extraction

Shashi Narayan* University of Edinburgh shashi.narayan@ed.ac.uk

Ronald Cardenas* Charles University in Prague ronald.cardenas@matfyz.cz

Nikos Papasarantopoulos* University of Edinburgh nikos.papasa@ed.ac.uk

Shay B. Cohen **Mirella Lapata** University of Edinburgh {scohen,mlap}@inf.ed.ac.uk

Jiangsheng Yu **Yi Chang** Huawei Technologies {jiangsheng.yu,yi.chang}@huawei.com

Abstract

Document modeling is essential to a variety of natural language understanding tasks. We propose to use external information to improve document modeling for problems that can be framed as sentence extraction. We develop a framework composed of a hierarchical document encoder and an attention-based extractor with attention over external information. We evaluate our model on extractive document summarization (where the external information is image captions and the title of the document) and answer selection (where the external information is a question). We show that our model consistently outperforms strong baselines, in terms of both informativeness and fluency (for CNN document summarization) and achieves state-of-the-art results for answer selection on WikiQA and NewsQA.[1]

1 Introduction

Recurrent neural networks have become one of the most widely used models in natural language processing (NLP). A number of variants of RNNs such as Long Short-Term Memory networks (LSTM; Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit networks (GRU; Cho et al., 2014) have been designed to model text capturing long-term dependencies in problems such as language modeling. However, document modeling, a key to many natural language understanding tasks, is still an open challenge. Recently, some neural network architectures were proposed to capture large context for modeling text (Mikolov and Zweig, 2012; Ghosh et al., 2016; Ji et al., 2015; Wang and Cho, 2016). Lin et al. (2015) and Yang et al. (2016) proposed a hierarchical RNN network for document-level modeling as well as sentence-level modeling, at the cost of increased computational complexity. Tran et al. (2016) further proposed a contextual language model that considers information at inter-document level.

It is challenging to rely only on the document for its understanding, and as such it is not surprising that these models struggle on problems such as document summarization (Cheng and Lapata, 2016; Chen et al., 2016; Nallapati et al., 2017; See et al., 2017; Tan and Wan, 2017) and machine reading comprehension (Trischler et al., 2016; Miller et al., 2016; Weissenborn et al., 2017; Hu et al., 2017; Wang et al., 2017). In this paper, we formalize the use of external information to further guide document modeling for end goals.

We present a simple yet effective document modeling framework for sentence extraction that allows machine reading with "external attention." Our model includes a neural hierarchical document encoder (or a machine reader) and a hierarchical attention-based sentence extractor. Our hierarchical document encoder resembles the architectures proposed by Cheng and Lapata (2016) and Narayan et al. (2018) in that it derives the document meaning representation from its sentences and their constituent words. Our novel sentence extractor combines this document meaning representation with an attention mechanism (Bahdanau et al., 2015) over the external information to label sentences from the input document. Our model explicitly biases the extractor with external cues and

*The first three authors made equal contributions to this paper. The work was done when the second author was visiting Edinburgh.

[1]Our TensorFlow code and datasets are publicly available at https://github.com/shashiongithub/Document-Models-with-Ext-Information.

implicitly biases the encoder through training.

We demonstrate the effectiveness of our model on two problems that can be naturally framed as sentence extraction with external information. These two problems, extractive document summarization and answer selection for machine reading comprehension, both require local and global contextual reasoning about a given document. Extractive document summarization systems aim at creating a summary by identifying (and subsequently concatenating) the most important sentences in a document, whereas answer selection systems select the candidate sentence in a document most likely to contain the answer to a query. For document summarization, we exploit the title and image captions which often appear with documents (specifically newswire articles) as external information. For answer selection, we use word overlap features, such as the inverse sentence frequency (ISF, Trischler et al., 2016) and the inverse document frequency (IDF) together with the query, all formulated as external cues.

Our main contributions are three-fold: First, our model ensures that sentence extraction is done in a larger (rich) context, i.e., the full document is read first before we start labeling its sentences for extraction, and each sentence labeling is done by implicitly estimating its local and global relevance to the document and by directly attending to some external information for importance cues.

Second, while external information has been shown to be useful for summarization systems using traditional hand-crafted features (Edmundson, 1969; Kupiec et al., 1995; Mani, 2001), our model is the first to exploit such information in deep learning-based summarization. We evaluate our models automatically (in terms of ROUGE scores) on the CNN news highlights dataset (Hermann et al., 2015). Experimental results show that our summarizer, informed with title and image captions, consistently outperforms summarizers that do not use this information. We also conduct a human evaluation to judge which type of summary participants prefer. Our results overwhelmingly show that human subjects find our summaries more informative and complete.

Lastly, with the machine reading capabilities of our model, we confirm that a full document needs to be "read" to produce high quality extracts allowing a rich contextual reasoning, in contrast to previous answer selection approaches that often measure a score between each sentence in the document and the question and return the sentence with highest score in an isolated manner (Yin et al., 2016; dos Santos et al., 2016; Wang et al., 2016). Our model with ISF and IDF scores as external features achieves competitive results for answer selection. Our ensemble model combining scores from our model and word overlap scores using a logistic regression layer achieves state-of-the-art results on the popular question answering datasets WikiQA (Yang et al., 2015) and NewsQA (Trischler et al., 2016), and it obtains comparable results to the state of the art for SQuAD (Rajpurkar et al., 2016). We also evaluate our approach on the MSMarco dataset (Nguyen et al., 2016) and elaborate on the behavior of our machine reader in a scenario where each candidate answer sentence is contextually independent of each other.

2 Document Modeling For Sentence Extraction

Given a document D consisting of a sequence of n sentences $(s_1, s_2, ..., s_n)$, we aim at labeling each sentence s_i in D with a label $y_i \in \{0, 1\}$ where $y_i = 1$ indicates that s_i is extraction-worthy and 0 otherwise. Our architecture resembles those previously proposed in the literature (Cheng and Lapata, 2016; Nallapati et al., 2017). The main components include a sentence encoder, a document encoder, and a novel sentence extractor (see Figure 1) that we describe in more detail below. The novel characteristics of our model are that each sentence is labeled by implicitly estimating its (local and global) relevance to the document and by directly attending to some external information for importance cues.

Sentence Encoder A core component of our model is a convolutional sentence encoder (Kim, 2014; Kim et al., 2016) which encodes sentences into continuous representations. We use temporal narrow convolution by applying a kernel filter K of width h to a window of h words in sentence s to produce a new feature. This filter is applied to each possible window of words in s to produce a feature map $f \in R^{k-h+1}$ where k is the sentence length. We then apply max-pooling over time over the feature map f and take the maximum value as the feature corresponding to this particular filter K. We use multiple kernels of various sizes and each kernel multiple times to construct the representation of a sentence. In Figure 1, ker-

nels of size 2 (red) and 4 (blue) are applied three times each. The max-pooling over time operation yields two feature lists f^{K_2} and $f^{K_4} \in R^3$. The final sentence embeddings have six dimensions.

Document Encoder The document encoder composes a sequence of sentences to obtain a document representation. We use a recurrent neural network with LSTM cells to avoid the vanishing gradient problem when training long sequences (Hochreiter and Schmidhuber, 1997). Given a document D consisting of a sequence of sentences $(s_1, s_2, \ldots, s_n)$, we follow common practice and feed the sentences in reverse order (Sutskever et al., 2014; Li et al., 2015; Filippova et al., 2015).

Sentence Extractor Our sentence extractor sequentially labels each sentence in a document with 1 or 0 by implicitly estimating its relevance in the document and by directly attending to the external information for importance cues. It is implemented with another RNN with LSTM cells with an attention mechanism (Bahdanau et al., 2015) and a softmax layer. Our attention mechanism differs from the standard practice of attending intermediate states of the input (encoder). Instead, our extractor attends to a sequence of p pieces of external information $E : (e_1, e_2, ..., e_p)$ relevant for the task (e.g., e_i is a title or an image caption for summarization) for cues. At time t_i, it reads sentence s_i and makes a binary prediction, conditioned on the document representation (obtained from the document encoder), the previously labeled sentences and the external information. This way, our labeler is able to identify locally and globally important sentences within the document which correlate well with the external information.

Given sentence s_t at time step t, it returns a probability distribution over labels as:

$$p(y_t|s_t, D, E) = \text{softmax}(g(h_t, h'_t)) \quad (1)$$

$$g(h_t, h'_t) = U_o(V_h h_t + W'_h h'_t) \quad (2)$$

$$h_t = \text{LSTM}(s_t, h_{t-1})$$

$$h'_t = \sum_{i=1}^{p} \alpha_{(t,i)} e_i,$$

$$\text{where } \alpha_{(t,i)} = \frac{\exp(h_t e_i)}{\sum_j \exp(h_t e_j)}$$

where $g(\cdot)$ is a single-layer neural network with parameters U_o, V_h and W'_h. h_t is an intermediate RNN state at time step t. The dynamic context vector h'_t is essentially the weighted sum of the external information $(e_1, e_2, \ldots, e_p)$. Figure 1 summarizes our model.

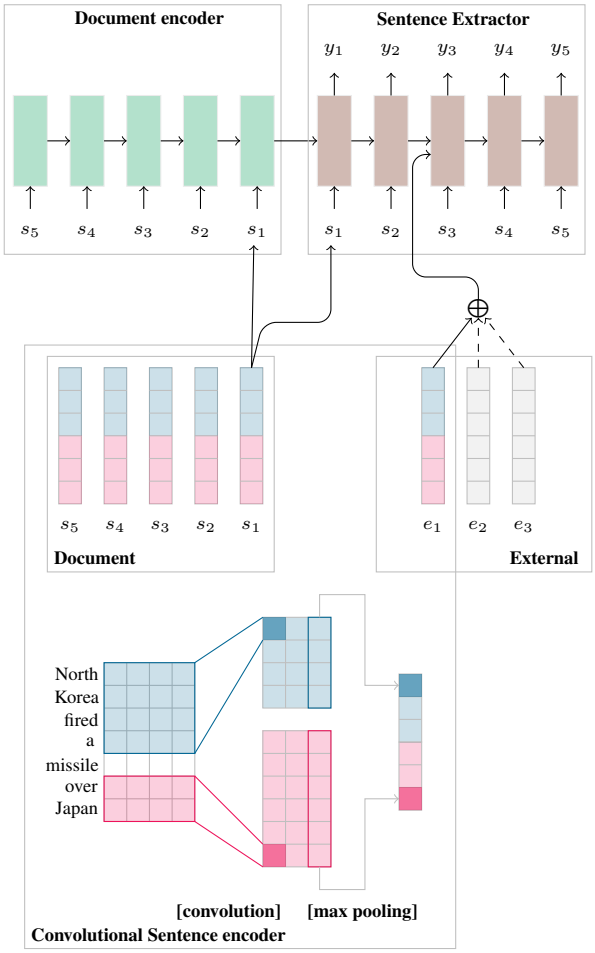

Figure 1: Hierarchical encoder-decoder model for sentence extraction with external attention. $s_1, \ldots, s_5$ are sentences in the document and, e_1, e_2 and e_3 represent external information. For the extractive summarization task, e_is are external information such as title and image captions. For the answers selection task, e_is are the query and word overlap features.

3 Sentence Extraction Applications

We validate our model on two sentence extraction problems: extractive document summarization and answer selection for machine reading comprehension. Both these tasks require local and global contextual reasoning about a given document. As such, they test the ability of our model to facilitate document modeling using external information.

Extractive Summarization An extractive summarizer aims to produce a summary $\mathcal{S}$ by selecting m sentences from D (where $m < n$). In this setting, our sentence extractor sequentially predicts label $y_i \in \{0, 1\}$ (where 1 means that s_i should be included in the summary) by assigning score $p(y_i|s_i, D, E, \theta)$ quantifying the relevance of s_i to the summary. We assemble a summary $\mathcal{S}$ by selecting m sentences with top $p(y_i = 1|s_i, D, E, \theta)$ scores.

We formulate external information E as the sequence of the title and the image captions associated with the document. We use the convolutional sentence encoder to get their sentence-level representations.

Answer Selection Given a question q and a document D , the goal of the task is to select one candidate sentence $s_i \in D$ in which the answer exists. In this setting, our sentence extractor sequentially predicts label $y_i \in \{0, 1\}$ (where 1 means that s_i contains the answer) and assign score $p(y_i|s_i, D, E, \theta)$ quantifying s_i's relevance to the query. We return as answer the sentence s_i with the highest $p(y_i = 1|s_i, D, E, \theta)$ score.

We treat the question q as external information and use the convolutional sentence encoder to get its sentence-level representation. This simplifies Eq. (1) and (2) as follow:

$$p(y_t|s_t, D, q) = \text{softmax}(g(h_t, q)) \quad (3)$$
$$g(h_t, q) = U_o(V_h h_t + W_q q),$$

where V_h and W_q are network parameters. We exploit the simplicity of our model to further assimilate external features relevant for answer selection: the inverse sentence frequency (ISF, (Trischler et al., 2016)), the inverse document frequency (IDF) and a modified version of the ISF score which we call *local ISF*. Trischler et al. (2016) have shown that a simple ISF baseline (i.e., a sentence with the highest ISF score as an answer) correlates well with the answers. The ISF score α_{s_i} for the sentence s_i is computed as $\alpha_{s_i} = \sum_{w \in s_i \cap q} \text{IDF}(w)$, where IDF is the inverse document frequency score of word w, defined as: $\text{IDF}(w) = \log \frac{N}{N_w}$, where N is the total number of sentences in the training set and N_w is the number of sentences in which w appears. Note that, $s_i \cap q$ refers to the set of words that appear both in s_i and in q. Local ISF is calculated in the same manner as the ISF score, only with setting the total number of sentences (N) to the number of sentences in the article that is being analyzed.

More formally, this modifies Eq. (3) as follows:

$$p(y_t|s_t, D, q) = \text{softmax}(g(h_t, q, \alpha_t, \beta_t, \gamma_t)) \quad (4)$$

where α_t, β_t and γ_t are the ISF, IDF and local ISF scores (real values) of sentence s_t respectively . The function g is calculated as follows:

$$\begin{aligned} g(h_t, q, \alpha_t, \beta_t, \gamma_t) = & U_o\,(V_h h_t + \\ & W_q q + W_{\text{isf}}(\alpha_t \cdot \overline{1}) + \\ & W_{\text{idf}}(\beta_t \cdot \overline{1}) + W_{\text{lisf}}(\gamma_t \cdot \overline{1})\big)\,, \end{aligned}$$

where W_{isf}, W_{idf} and W_{lisf} are new parameters added to the network and $\overline{1}$ is a vector of 1s of size equal to the sentence embedding size. In Figure 1, these external feature vectors are represented as 6-dimensional gray vectors accompanied with dashed arrows.

4 Experiments and Results

This section presents our experimental setup and results assessing our model in both the extractive summarization and answer selection setups. In the rest of the paper, we refer to our model as XNET for its ability to exploit e**X**ternal information to improve document representation.

4.1 Extractive Document Summarization

Summarization Dataset We evaluated our models on the CNN news highlights dataset (Hermann et al., 2015).[2] We used the standard splits of Hermann et al. (2015) for training, validation, and testing (90,266/1,220/1,093 documents). We followed previous studies (Cheng and Lapata, 2016; Nallapati et al., 2016, 2017; See et al., 2017; Tan and Wan, 2017) in assuming that the

[2] Hermann et al. (2015) have also released the DailyMail dataset, but we do not report our results on this dataset. We found that the script written by Hermann et al. to crawl DailyMail articles mistakenly extracts image captions as part of the main body of the document. As image captions often do not have sentence boundaries, they blend with the sentences of the document unnoticeably. This leads to the production of erroneous summaries.

"story highlights" associated with each article are gold-standard abstractive summaries. We trained our network on a named-entity-anonymized version of news articles. However, we generated deanonymized summaries and evaluated them against gold summaries to facilitate human evaluation and to make human evaluation comparable to automatic evaluation.

To train our model, we need documents annotated with sentence extraction information, i.e., each sentence in a document is labeled with 1 (summary-worthy) or 0 (not summary-worthy). We followed Nallapati et al. (2017) and automatically extracted ground truth labels such that all positively labeled sentences from an article collectively give the highest ROUGE (Lin and Hovy, 2003) score with respect to the gold summary.

We used a modified script of Hermann et al. (2015) to extract titles and image captions, and we associated them with the corresponding articles. All articles get associated with their titles. The availability of image captions varies from 0 to 414 per article, with an average of 3 image captions. There are 40% CNN articles with at least one image caption.

All sentences, including titles and image captions, were padded with zeros to a sentence length of 100. All input documents were padded with zeros to a maximum document length of 126. For each document, we consider a maximum of 10 image captions. We experimented with various numbers (1, 3, 5, 10 and 20) of image captions on the validation set and found that our model performed best with 10 image captions. We refer the reader to the supplementary material for more implementation details to replicate our results.

Comparison Systems We compared the output of our model against the standard baseline of simply selecting the first three sentences from each document as the summary. We refer to this baseline as LEAD in the rest of the paper.

We also compared our system against the sentence extraction system of Cheng and Lapata (2016). We refer to this system as POINTERNET as the neural attention architecture in Cheng and Lapata (2016) resembles the one of Pointer Networks (Vinyals et al., 2015).[3] It does not exploit any external information.[4] Cheng and Lapata (2016) report only on the DailyMail dataset. We used their code (https://github.com/cheng6076/NeuralSum) to produce results on the CNN dataset.[5]

MODELS	R1	R2	R3	R4	RL	Avg.
LEAD	49.2	18.9	9.8	6.0	43.8	25.5
POINTERNET	53.3	19.7	10.4	6.4	47.2	27.4
XNET+TITLE	55.0	21.6	11.7	7.5	48.9	28.9
XNET+CAPTION	55.3	21.3	11.4	7.2	49.0	28.8
XNET+FS	54.8	21.1	11.3	7.2	48.6	28.6
Combination Models (XNET+)						
TITLE+CAPTION	**55.4**	**21.8**	**11.8**	**7.5**	**49.2**	**29.2**
TITLE+FS	55.1	21.6	11.6	7.4	48.9	28.9
CAPTION+FS	55.3	21.5	11.5	7.3	49.0	28.9
TITLE+CAPTION+FS	55.4	21.5	11.6	7.4	49.1	29.0

Table 1: Ablation results on the validation set. We report R1, R2, R3, R4, RL and their average (Avg.). The first block of the table presents LEAD and POINTERNET which do not use any external information. LEAD is the baseline system selecting first three sentences. POINTERNET is the sentence extraction system of Cheng and Lapata. XNET is our model. The second and third blocks of the table present different variants of XNET. We experimented with three types of external information: title (TITLE), image captions (CAPTION) and the first sentence (FS) of the document. The bottom block of the table presents models with more than one type of external information. The best performing model (highlighted in boldface) is used on the test set.

Automatic Evaluation To automatically assess the quality of our summaries, we used ROUGE (Lin and Hovy, 2003), a recall-oriented metric, to compare our model-generated summaries to manually-written highlights.[6] Previous work has reported ROUGE-1 (R1) and ROUGE-2 (R2) scores to access informativeness, and ROUGE-L (RL) to access fluency. In addition to R1, R2 and RL, we also report ROUGE-3 (R3) and ROUGE-4 (R4) capturing higher order n-grams overlap to assess informativeness and fluency simultaneously.

[3]The architecture of POINTERNET is closely related to our model without external information.

[4]Adding external information to POINTERNET is an interesting direction of research but we do not pursue it here. It requires decoding with multiple types of attentions and this is not the focus of this paper.

[5]We are unable to compare our results to the extractive system of Nallapati et al. (2017) because they report their results on the DailyMail dataset and their code is not available. The abstractive systems of Chen et al. (2016) and Tan and Wan (2017) report their results on the CNN dataset, however, their results are not comparable to ours as they report on the full-length F_1 variants of ROUGE to evaluate their abstractive summaries. We report ROUGE recall scores which is more appropriate to evaluate our extractive summaries.

[6]We used pyrouge, a Python package, to compute all our ROUGE scores with parameters "-a -c 95 -m -n 4 -w 1.2."

We report our results on both full length (three sentences with the top scores as the summary) and fixed length (first 75 bytes and 275 bytes as the summary) summaries. For full length summaries, our decision of selecting three sentences is guided by the fact that there are 3.11 sentences on average in the gold highlights of the training set. We conduct our ablation study on the validation set with full length ROUGE scores, but we report both fixed and full length ROUGE scores for the test set.

We experimented with two types of external information: title (TITLE) and image captions (CAPTION). In addition, we experimented with the first sentence (FS) of the document as external information. Note that the latter is not external information, it is a sentence in the document. However, we wanted to explore the idea that the first sentence of the document plays a crucial part in generating summaries (Rush et al., 2015; Nallapati et al., 2016). XNET with FS acts as a baseline for XNET with title and image captions.

We report the performance of several variants of XNET on the validation set in Table 1. We also compare them against the LEAD baseline and POINTERNET. These two systems do not use any additional information. Interestingly, all the variants of XNET significantly outperform LEAD and POINTERNET. When the title (TITLE), image captions (CAPTION) and the first sentence (FS) are used separately as additional information, XNET performs best with TITLE as its external information. Our result demonstrates the importance of the title of the document in extractive summarization (Edmundson, 1969; Kupiec et al., 1995; Mani, 2001). The performance with TITLE and CAPTION is better than that with FS. We also tried possible combinations of TITLE, CAPTION and FS. All XNET models are superior to the ones without any external information. XNET performs best when TITLE and CAPTION are jointly used as external information (55.4%, 21.8%, 11.8%, 7.5%, and 49.2% for R1, R2, R3, R4, and RL respectively). It is better than the the LEAD baseline by 3.7 points on average and than POINTERNET by 1.8 points on average, indicating that external information is useful to identify the gist of the document. We use this model for testing purposes.

Our final results on the test set are shown in Table 2. It turns out that for smaller summaries (75 bytes) LEAD and POINTERNET are superior to XNET. This result could be because LEAD (always) and POINTERNET (often) include the first sentence in their summaries, whereas, XNET is better capable at selecting sentences from various document positions. This is not captured by smaller summaries of 75 bytes, but it becomes more evident with longer summaries (275 bytes and full length) where XNET performs best across all ROUGE scores. We note that POINTERNET outperforms LEAD for 75-byte summaries, then its performance drops behind LEAD for 275-byte summaries, but then it outperforms LEAD for full length summaries on the metrics R1, R2 and RL. It shows that POINTERNET with its attention over sentences in the document is capable of exploring more than first few sentences in the document, but it is still behind XNET which is better at identifying salient sentences in the document. XNET performs significantly better than POINTERNET by 0.8 points for 275-byte summaries and by 1.9 points for full length summaries, on average for all ROUGE scores.

MODELS	R1	R2	R3	R4	RL
Fixed length: 75b					
LEAD	20.1	7.1	**3.5**	2.1	14.6
POINTERNET	**20.3**	**7.2**	**3.5**	**2.2**	**14.8**
XNET	20.2	7.1	3.4	2.0	14.6
Fixed length: 275b					
LEAD	39.1	14.5	7.6	4.7	34.6
POINTERNET	38.6	13.9	7.3	4.4	34.3
XNET	**39.7**	**14.7**	**7.9**	**5.0**	**35.2**
Full length summaries					
LEAD	49.3	19.5	10.7	6.9	43.8
POINTERNET	51.7	19.7	10.6	6.6	45.7
XNET	**54.2**	**21.6**	**12.0**	**7.9**	**48.1**

Table 2: Final results on the test set. POINTERNET is the sentence extraction system of Cheng and Lapata. XNET is our best model from Table 1. Best ROUGE score in each block and each column is highlighted in boldface.

Models	1st	2nd	3rd	4th
LEAD	0.15	0.17	**0.47**	0.21
POINTERNET	0.16	0.05	0.31	**0.48**
XNET	0.28	**0.53**	0.15	0.04
HUMAN	**0.41**	0.25	0.07	0.27

Table 3: Human evaluations: Ranking of various systems. Rank 1st is best and rank 4th, worst. Numbers show the percentage of times a system gets ranked at a certain position.

Human Evaluation We complement our automatic evaluation results with human evaluation. We randomly selected 20 articles from the test set.

Annotators were presented with a news article and summaries from four different systems. These include the LEAD baseline, POINTERNET, XNET and the human authored highlights. We followed the guidelines in Cheng and Lapata (2016), and asked our participants to rank the summaries from best (1st) to worst (4th) in order of informativeness (does the summary capture important information in the article?) and fluency (is the summary written in well-formed English?). We did not allow any ties and we only sampled articles with non-identical summaries. We assigned this task to five annotators who were proficient English speakers. Each annotator was presented with all 20 articles. The order of summaries to rank was randomized per article. An example of summaries our subjects ranked is provided in the supplementary material.

The results of our human evaluation study are shown in Table 3. As one might imagine, HUMAN gets ranked 1st most of the time (41%). However, it is closely followed by XNET which ranked 1st 28% of the time. In comparison, POINTERNET and LEAD were mostly ranked at 3rd and 4th places. We also carried out pairwise comparisons between all models in Table 3 for their statistical significance using a one-way ANOVA with post-hoc Tukey HSD tests with ($p < 0.01$). It showed that XNET is significantly better than LEAD and POINTERNET, and it does not differ significantly from HUMAN. On the other hand, POINTERNET does not differ significantly from LEAD and it differs significantly from both XNET and HUMAN. The human evaluation results corroborates our empirical results in Table 1 and Table 2: XNET is better than LEAD and POINTERNET in producing informative and fluent summaries.

4.2 Answer Selection

Question Answering Datasets We run experiments on four datasets collected for open domain question-answering tasks: WikiQA (Yang et al., 2015), SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), and MSMarco (Nguyen et al., 2016).

NewsQA was especially designed to present lexical and syntactic divergence between questions and answers. It contains 119,633 questions posed by crowdworkers on 12,744 CNN articles previously collected by Hermann et al. (2015). In a similar manner, SQuAD associates 100,000+ question with a Wikipedia article's first paragraph, for 500+ previously chosen articles. WikiQA was collected by mining web-searching query logs and then associating them with the summary section of the Wikipedia article presumed to be related to the topic of the query. A similar collection procedure was followed to create MSMarco with the difference that each candidate answer is a whole paragraph from a different browsed website associated with the query.

We follow the widely used setup of leaving out unanswered questions (Trischler et al., 2016; Yang et al., 2015) and adapt the format of each dataset to our task of answer sentence selection by labeling a candidate sentence with 1 if any answer span is contained in that sentence. In the case of MSMarco, each candidate paragraph comes associated with a label, hence we treat each one as a single long sentence. Since SQuAD keeps the official test dataset hidden and MSMarco does not provide labels for its released test set, we report results on their official validation sets. For validation, we set apart 10% of each official training set.

Our dataset splits consist of 92,525, 5,165 and 5,124 samples for NewsQA; 79,032, 8,567, and 10,570 for SQuAD; 873, 122, and 237 for WikiQA; and 79,704, 9,706, and 9,650 for MSMarco, for training, validation, and testing respectively.

Comparison Systems We compared the output of our model against the ISF (Trischler et al., 2016) and LOCALISF baselines. Given an article, the sentence with the highest ISF score is selected as an answer for the ISF baseline and the sentence with the highest local ISF score for the LOCALISF baseline. We also compare our model against a neural network (PAIRCNN) that encodes $(\text{question}, \text{candidate})$ in an isolated manner as in previous work (Yin et al., 2016; dos Santos et al., 2016; Wang et al., 2016). The architecture uses the sentence encoder explained in earlier sections to learn the question and candidate representations. The distribution over labels is given by $p(y_t|q) = p(y_t|s_t, q) = \text{softmax}(g(s_t, q))$ where $g(s_t, q) = \text{ReLU}(W_{sq} \cdot [s_t; q] + b_{sq})$. In addition, we also compare our model against APCNN (dos Santos et al., 2016), ABCNN (Yin et al., 2016), L.D.C (Wang and Jiang, 2017), KVMemNN (Miller et al., 2016), and COMPAGGR, a state-of-the-art system by Wang et al. (2017).

We experiment with several variants of our model. XNET is the vanilla version of our sen-

	SQuAD			WikiQA			NewsQA			MSMarco		
	ACC	MAP	MRR	ACC	MAP	MRR	ACC	MAP	MRR	ACC	MAP	MRR
WRD CNT	77.84	27.50	27.77	51.05	48.91	49.24	44.67	46.48	46.91	20.16	19.37	19.51
WGT WRD CNT	78.43	28.10	28.38	49.79	50.99	51.32	45.24	48.20	48.64	20.50	20.06	20.23
AP-CNN	-	-	-	-	68.86	69.57	-	-	-	-	-	-
ABCNN	-	-	-	-	69.21	71.08	-	-	-	-	-	-
L.D.C	-	-	-	-	70.58	72.26	-	-	-	-	-	-
KV-MemNN	-	-	-	-	70.69	72.65	-	-	-	-	-	-
LOCALISF	79.50	27.78	28.05	49.79	49.57	50.11	44.69	48.40	46.48	20.21	20.22	20.39
ISF	78.85	28.09	28.36	48.52	46.53	46.72	45.61	48.57	48.99	20.52	20.07	20.23
PAIRCNN	32.53	46.34	46.35	32.49	39.87	38.71	25.67	40.16	39.89	14.92	34.62	35.14
COMPAGGR	85.52	91.05	91.05	60.76	73.12	74.06	54.54	67.63	68.21	32.05	**52.82**	**53.43**
XNET	35.50	58.46	58.84	54.43	69.12	70.22	26.18	42.28	42.43	15.45	35.42	35.97
XNETTOPK	36.09	59.70	59.32	55.00	68.66	70.24	29.41	46.69	46.97	17.04	37.60	38.16
LRXNET	**85.63**	**91.10**	**91.85**	**63.29**	**76.57**	**75.10**	**55.17**	**68.92**	**68.43**	**32.92**	31.15	30.41
XNET+	79.39	87.32	88.00	57.08	70.25	71.28	47.23	61.81	61.42	23.07	42.88	43.42

Table 4: Results (in percentage) for answer selection comparing our approaches (bottom part) to baselines (top): AP-CNN (dos Santos et al., 2016), ABCNN (Yin et al., 2016), L.D.C (Wang and Jiang, 2017), KV-MemNN (Miller et al., 2016), and COMPAGGR, a state-of-the-art system by Wang et al. (2017). (WGT) WRD CNT stands for the (weighted) word count baseline. See text for more details.

tence extractor conditioned only on the query q as external information (Eq. (3)). XNET+ is an extension of XNET which uses ISF, IDF and local ISF scores in addition to the query q as external information (Eqn. (4)). We also experimented with a baseline XNETTOPK where we choose the top k sentences with highest ISF score, and then among them choose the one with the highest probability according to XNET. In our experiments, we set $k = 5$. In the end, we experimented with an ensemble network LRXNET which combines the XNET score, the COMPAGGR score and other word-overlap-based scores (tweaked and optimized for each dataset separately) for each sentence using a logistic regression classifier. It uses ISF and LocalISF scores for NewsQA, IDF and ISF scores for SQuAD, sentence length, IDF and ISF scores for WikiQA, and word overlap and ISF score for MSMarco. We refer the reader to the supplementary material for more implementation and optimization details to replicate our results.

Evaluation Metrics We consider metrics that evaluate systems that return a ranked list of candidate answers: mean average precision (MAP), mean reciprocal rank (MRR), and accuracy (ACC).

Results Table 4 gives the results for the test sets of NewsQA and WikiQA, and the original validation sets of SQuAD and MSMarco. Our first observation is that XNET outperforms PAIRCNN, supporting our claim that it is beneficial to read the whole document in order to make decisions, instead of only observing each candidate in isolation.

Secondly, we can observe that ISF is indeed a strong baseline that outperforms XNET. This means that just "reading" the document using a vanilla version of XNET is not sufficient, and help is required through a coarse filtering. Indeed, we observe that XNET+ outperforms all baselines except for COMPAGGR. Our ensemble model LRXNET can ultimately surpass COMPAGGR on majority of the datasets.

This consistent behavior validates the machine reading capabilities and the improved document representation with external features of our model for answer selection. Specifically, the combination of document reading and word overlap features is required to be done in a soft manner, using a classification technique. Using it as a hard constraint, with XNETTOPK, does not achieve the best result. We believe that often the ISF score is a better indicator of answer presence in the vicinity of certain candidate instead of in the candidate itself. As such, XNET+ is capable of using this feature in datasets with richer context.

It is worth noting that the improvement gained by LRXNET over the state-of-the-art follows a pattern. For the SQuAD dataset, the results are comparable (less than 1%). However, the improvement for WikiQA reaches ∼3% and then the gap shrinks again for NewsQA, with an improvement of ∼1%. This could be explained by the fact that each sample of the SQuAD is a paragraph, compared to an article summary for WikiQA, and

to an entire article for NewsQA. Hence, we further strengthen our hypothesis that a richer context is needed to achieve better results, in this case expressed as document length, but as the length of the context increases the limitation of sequential models to learn from long rich sequences arises.[7]

Interestingly, our model lags behind COMPAGGR on the MSMarco dataset. It turns out this is due to contextual independence between candidates in the MSMarco dataset, i.e., each candidate is a stand-alone paragraph in this dataset, in contrast to contextually dependent candidate sentences from a document in the NewsQA, SQuAD and WikiQA datasets. As a result, our models (XNET+ and LRXNET) with document reading abilities perform poorly. This can be observed by the fact that XNET and PAIRCNN obtain comparable results. COMPAGGR performs better because comparing each candidate independently is a better strategy.

5 Conclusion

We describe an approach to model documents while incorporating external information that informs the representations learned for the sentences in the document. We implement our approach through an attention mechanism of a neural network architecture for modeling documents.

Our experiments with extractive document summarization and answer selection tasks validates our model in two ways: first, we demonstrate that external information is important to guide document modeling for natural language understanding tasks. Our model uses image captions and the title of the document for document summarization, and the query with word overlap features for answer selection and outperforms its counterparts that do not use this information. Second, our external attention mechanism successfully guides the learning of the document representation for the relevant end goal. For answer selection, we show that inserting the query with word overlap features using our external attention mechanism outperforms state-of-the-art systems that naturally also have access to this information.

Acknowledgments

We thank Jianpeng Cheng for providing us with the CNN dataset and the implementation of PointerNet. We also thank the members of the Edinburgh NLP group for participating in our human evaluation experiments. This work greatly benefitted from discussions with Jianpeng Cheng, Annie Louis, Pedro Balage, Alfonso Mendes, Sebastião Miranda, and members of the Edinburgh NLP group. We gratefully acknowledge the support of the European Research Council (Lapata; award number 681760), the European Union under the Horizon 2020 SUMMA project (Narayan, Cohen; grant agreement 688139), and Huawei Technologies (Cohen).

[7] See the supplementary material for an example supporting our hypothesis.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, California, USA.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. New York, USA, pages 2754–2760.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, pages 484–494.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*. Doha, Qatar, pages 1724–1734.

Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR* abs/1602.03609.

Harold P. Edmundson. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery* 16(2):264–285.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pages 360–368.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual LSTM (CLSTM) models for large scale NLP tasks. *CoRR* abs/1602.06291.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*. pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9(8):1735–1780.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR* abs/1705.02798.

Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein. 2015. Document context language models. *CoRR* abs/1511.03962.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Phoenix, Arizona USA, pages 2741–2749.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, USA, pages 406–407.

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China, pages 1106–1115.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Edmonton, Canada, pages 71–78.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods on Natural Language Processing*. Lisbon, Portugal, pages 899–907.

Inderjeet Mani. 2001. *Automatic Summarization*. Natural language processing. John Benjamins Publishing Company.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proceedings of the Spoken Language Technology Workshop*. IEEE, pages 234–239.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing*. Austin, Texas, pages 1400–1409.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. San Francisco, California USA, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany, pages 280–290.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. New Orleans, US.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS Marco: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches, co-located with the 30th Annual Conference on Neural Information Processing Systems*. Barcelona, Spain.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 2383–2392.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pages 379–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada, pages 1073–1083.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*. pages 3104–3112.

Jiwei Tan and Xiaojun Wan. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada, pages 1171–1181.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2016. Inter-document contextual language model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California, pages 762–766.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *CoRR* abs/1611.09830.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28*. pages 2692–2700.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France.

Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, pages 1319–1329.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada, pages 189–198.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 1340–1349.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning*. Vancouver, Canada, pages 271–280.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pages 2013–2018.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. 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*. San Diego, California, pages 1480–1489.

Wenpeng Yin, Hinrich Schtze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4:259–272.