Generating Arabic News Headlines
(Class Project)

Omar Alhadlaq
Mentor: Michael Hahn
Department of Computer Science
Stanford University
hadlaq@cs.stanford.edu

Abstract

In this project, we aim to build a model that generates accurate headlines of news articles. This task can be seen as a text summarization task, which is vital for the understanding of natural language. The Arabic language is a main focus of this effort since it is underrepresented in the deep learning literature, even though it introduces many new complexities and challenges. In this project, we also introduce a new dataset of 326K Arabic article and headline pairs. We also experiment with five predictive models, which are: a basic Seq2Seq model, a Seq2Seq model with scaled multiplicative attention, a Seq2Seq model with copying, a Seq2Seq model with subword encoding, and finally a Transformer model. The Seq2Seq subword encoding model and the Transformer achieve the best performance with ROUGE scores of 37.23 and 36.74 respectively. We also present performance analysis showing how attention mechanisms and subword encoding can help with factual accuracy and dealing with out-of-vocabulary words, which both are common issues with headline generation. Also, we conclude that our models struggle with generating creative headlines which can be a requirement in the task. Finally, we suggest new directions for dealing with the creativity problem.

1 Introduction

Text summarization is a common natural language task of producing a coherent, informative, accurate and brief summary of a document. Mainly, there are two approaches to summarization: extractive and generative. Extractive approaches select the sentences and the passages that contribute the most to the meaning of the document. On the other hand, generative approaches produce new pieces of text that are coherent and more similar to human-written summaries. This task is a cornerstone of NLP, since the ability to accurately identify the most important ideas in a document, as well as producing text that can communicate these ideas is a great step towards the understanding of natural language.

The generation of news headlines, given the news articles, is a special case of text summerization where summaries are even more brief and constrained to the one or two main ideas of the article. Therefore, progress in this realm is a progress in natural language understanding. In addition, the Arabic language is a main focus in this work, since it is underrepresented in the deep learning literature and introduce many complexities and challenges to the task. Therefore, the goal of this project is develop a model that is able to generate appropriate Arabic news headlines given the articles.

With the rising of neural networks in the recent years, and due to its ability to produce coherent text pieces, the literature has shifted from focusing on extractive models to generative models as they became more viable. Many of the recent work tackled text summarization as a conditioned language generation or a sequence-to-sequence mapping, much like what is used in neural machine translation [1, 2]. These approaches have shown great potential, however, they tend to inaccurately reproduce

factual details, and show an inability to handle out-of-vocabulary (OOV) words [3, 4]. In this paper, we try to remedy these issues by applying relatively new techniques, such as the use of attention and copying mechanisms.

In addition to known challenges in text summarization, Arabic imposes a new level of complexity to the task. In specific, two main challenges in Arabic are: the variation in writing forms and the lack of word boundaries. Every Arabic word has many proper forms that are considered as different words by the computer, although, they essentially mean the same thing. One reason is the presence and absence of Arabic diacritics (see figure 1). These are small marks that are added around letters which indicates short vowels. Since most Arabic readers can infer many of the diacritics even if they weren't written explicitly, many writers choose to omit some of the diacritics. Thus, the amount of diacritics used differs from one writer to another. Similarly, some letters, such as the letter Alif, has different forms, although only one form is considered correct based on the context, many writers make the mistake of using the wrong form. One technique to overcome this issue, known as Hamzah normalization, is to map all of the different forms of the letter to one of them. A downside of this is that it would generate words with the wrong spelling. The other challenge with Arabic, is the lack of word boundaries compared to English. For example, pronouns are usually attached to verbs, so the word "played" will have a different form in all of the sentences: "I played", "he played", "she played", "we played", and so on. As a result of these issues, we see words in the training corpus much less often than we would if the corpus was in English. So, to compensate, we introduce a new dataset that is more than ten times larger than available Arabic news articles datasets [5]. Moreover, we also use the Google SentencePiece model (SPM), to process the text on a subword-level rather than word- or character-level.

بِسْمِ ٱللهِ ٱلرَّحْمٰنِ ٱلرَّحِيمِ
بسم الله الرحمن الرحيم

Figure 1: An Arabic sentence with diacritics (top) and without (bottom).

2 Approaches

2.1 Sequence-to-Sequence (Baseline Model)

Since our task can be seen as a mapping from an input sequence of words to an output sequence of words, a Seq2Seq model is natural approach to start with. This family of models has enjoyed a tremendous success in a variety of natural language learning tasks, such as machine translation [6]. Our baseline model uses a 2-layer bidirectional Long Short-Term Memory (LSTM) encoder to map the input article to an article embedding of a fixed dimensionality. Then, it uses another 2-layer LSTM decoder to generate the title sequence from the article embedding. The decoder uses teacher forcing during training and beam search for inference.

2.2 Sequence-to-Sequence with Attention

Our second approach is very similar to the first one with the difference that we add a multiplicative attention layer between the encoder and the decoder. In specific, we use the a modified version of the Luong attention described in [7]. The general way of calculating the attention is to score a decoder hidden state and a corresponding encoder state, normalized over all encoder states to get attention scores summing to 1 as in equation (1).

$$a_d(e) = \frac{exp(\text{score}(h_d, h_e))}{\sum_e exp(\text{score}(h_d, h_e))} \tag{1}$$

The score function can have a variety of forms. The specific form that we use is a dot product between the two vectors, with the addition of a scaling factor inspired by the normalized form of Bahdanau attention as in equation (2).

$$\text{score}(h_d, h_e) = \frac{h_d^\top h_e}{\sqrt{n}} \tag{2}$$

For the two proposed approaches we used the official Tensorflow implementation [8].

2.3 Sequence-to-Sequence with Copying

Copying mechanisms, such as CopyNet, are one intuitive and effective technique to deal with rare and out-of-vocabulary (OOV) words [9]. The basic idea of CopyNet is at each decoding step T, instead of predicting a word from the vocabulary, the model may choose to output a word from the input source by copying it over directly to the output.

CopyNet is another encoder-decoder architecture, where the encoder part is similar to the one in the attention model. The decoder part is also similar to the attention model with the three major differences:

- Prediction: the decoder switches between generate-mode and copy-mode based on a mixed probabilistic model. In generate-mode, it predicts words from the vocabulary, while in copy-mode, it picks words from the input sequence.
- State update: the decoder updates the state at time t using the predicted word at time $t-1$. Also, it doesn't only use its word-embedding but also its corresponding hidden state from the encoder if it had one.
- Reading the encoder: the decoder attend to the hidden states of the encoder in order to decide the next predicted word. In CopyNet, it also performs "selective read" to the encoder hidden states which is similar to the attention step but decides the words that get copied.

At any timestep t, the decoder decides to output the word y_t with the following probability distribution:

$$p(y_t|s_t, y_{t-1}, c_t, M) = p(y_t, g|s_t, y_{t-1}, c_t, M) + p(y_t, c|s_t, y_{t-1}, c_t, M) \tag{3}$$

where g and c refer to generate-mode and copy-mode respectively. The probabilities of the two modes are given by:

$$p(y_t, g|\cdot) = \begin{cases} \frac{1}{Z} e^{\psi_g(y_t)}, & y_t \in \mathcal{V} \\ 0, & y_t \in \mathcal{X} \cap \mathcal{V} \\ \frac{1}{Z} e^{\psi_g(\text{unk})}, & y_t \notin \mathcal{X} \cup \mathcal{V} \end{cases} \tag{4}$$

$$p(y_t, c|\cdot) = \begin{cases} \frac{1}{Z} \sum_{j:x_j=y_t} e^{\psi_c(x_j)}, & y_t \in \mathcal{V} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In the previous equation, Z is the normalization term, $\mathcal{V}$ is the vocabulary set, and $\mathcal{X}$ is the set of unique words from the input. Moreover, $\psi_g(\cdot), \psi_c(\cdot)$ refer to special scoring functions for generate-mode and copy-mode respectively [9].

2.4 Sequence-to-Sequence with Byte-Pair-Encoding

In this approach, we process the input by a SetnencePiece model (SPM) [10]. This model lets us learn a vocabulary that provides a good compression rate of the text. At worst, the vocabulary can include all characters in the language, and therefore, we are guaranteed not to encounter any out-of-vocabulary (OOV) tokens in the input or the output. This helps us with OOV issue that is commonly encountered in text summarization systems, especially ones geared towards news articles where rare nouns and words are very common.

The newly learned vocabulary allows us to operate on a subword level, which is particularly useful for languages that lack word boundaries such as Arabic. It is very common in Arabic to combine two words into a single word, for example, the pronoun and the verb are usually combined. So, instead

of having a different word for each of: "he played", "she played", "we played", and "they played", in our new vocabulary, we have a single word for the verb "played", which allows us to reduce the needed vocabulary size significantly.

The segmentation algorithm used is byte-pair-encoding (BPE). Originally, BPE is a data compression algorithm that was invented in 1994. It works by iteratively replaceing the most frequent pair of bytes in a sequence with a single unused byte [11]. For word segmentation, we merge characters or character sequences instead of merging frequent bytes.

At first, the model starts with the vocabulary set that only includes all characters in the language, such that each word is represented as a sequence of characters. We also add a special end-of-word token, which allows us to restore the original word-level tokenization for the generated news headline. Then, it counts all token pairs in the corpus and replaces the most frequent pair ('A', 'B') with a new token 'AB'. The vocabulary size is equal to the size of the character vocabulary, plus the number of merge steps. We repeat the merging step and stop when we reach the desired vocabulary size pairs of bytes [12].

2.5 The Transformer

The Transformer is new model architecture to train sequence-to-sequence models [13]. It still has the encoder-decoder structure but it completely drop the use of recurrent neural networks and relies solely on attention mechanisms. The model parallelizes very well and so is much faster in training. More importantly, it has shown to out perform basic Seq2Seq models in many sequence learning tasks including machine translation and text summerization.

Encoder: The encoder is a stack of $N = 6$ encoding units. All of these are identical in structure but don't share weights. Each unit has two layers: a multi-head self-attention mechanism, and a fully connected feed-forward network. It also includes a residual connection around each of the two layers, followed by layer normalization, as in equation (6).

$$z = \text{LayerNorm}(x + \text{Layer}(x)) \tag{6}$$

All layers and all units in the encoder, as well as the embedding layers, has outputs of dimension $d = 512$.

Decoder: The decoder is also a stack of $N = 6$ decoding units. Similarly to the encoder unit, the decoding unit has the two multi-head self-attention mechanism, and a fully connected feed-forward network layers. However, it also adds a third layer of multi-head attention over the output of the encoder stack. The decoder also includes a residual connection around each of the layers, followed by layer normalization.

Multi-Head Attention: The authors propose a new attention mechanism called "Scaled Dot-Product Attention". The inputs of the attention consist of queries and keys of dimension d_k, and values of dimension d_v. The output is the dot products of the query with all keys, divide each by $\sqrt{d_k}$ and passed into a softmax layer to obtain the weights on the values. This process is captured in equation (7).

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{(d_k)}})V \tag{7}$$

However, in the Transformer, instead of using a single layer of the scaled dot-product attention with keys, values and queries of dimension d, the paper shows that a better way to conduct the attention is to use h different linear projections for the queries, keys and values into d_k, d_k and d_v dimensions respectively. On each of these projected queries, keys and values we then compute the scaled dot-product attention in parallel. The output values of this step are d_v-dimensional vectors, which are finally concatenated and projected into the desired dimension as described in equations (8, 9).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{8}$$

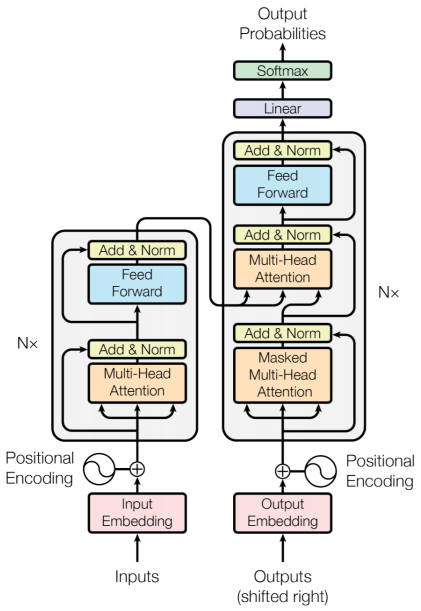

Figure 2: The Transformer model architecture as presented in the paper.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{9}$$

3 Experiments

3.1 The Dataset

3.1.1 Overview

We collected our own dataset for the purpose of this project and this class. The dataset is the first Arabic news articles dataset of this scale. It is composed of 326K article and title pairs collected from 3 different prominent Saudi news websites (see breakdown in table 1) using 3 different web scrappers that we developed.

	News Agency	**Website**	**#Articles**
1	Alarabiya	`www.alarabiya.net`	54K
2	Sabq	`www.sabq.org`	110K
3	Alriyadh	`www.alriyadh.com`	162K

Table 1: The sources of the collected dataset

3.1.2 Preprocessing

The titles and the articles are preprocessed such that punctuations are separated from the words and Arabic diacritics are removed. The dataset is divide into a train, development, and test sets of sizes 303K, 11.5K, and 11.5K respectively.

3.1.3 Analysis

The dataset has a total number of 40 million words composed of 450K unique words. The average title length is 10.46 words, while the average article length is 250.81 words. Most words are rare, which means they are seen less than 5 times in the entire corpus as can be seen in figure 3.

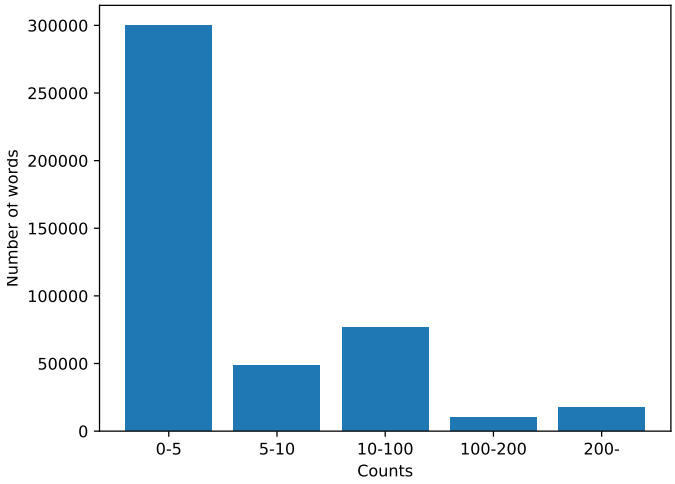

Figure 3: Word frequencies in the Arabic news dataset.

3.2 Evaluation Methods

The primary evaluation metric used for this task is ROUGE, which measures recall [14]. ROUGE indicates how much of the human produced titles appears in the model generated titles. Moreover, we also use BLEU as a secondary metric, which measures precision [15]. In specific, how much of the model generated titles appears in the human produced titles.

3.3 Experimental Details

We trained each of the five models for 35 epochs on a train dataset of size 303K. For the Seq2Seq models we used a hidden size of 256, which is also the embedding size. We used 80 as our max input length, meaning that we only look at the first 80 words of the article, which usually consists of the first 1 or 2 paragraphs. During inference, the maximum title length is set to 20 and we use a beam search with beam width of 6. For training, we used a dropout of 0.2 and an SGD optimizer with a learning rate of 1.0. Moreover, we clipped gradients to have a norm of 5.0, which we think has helped. For the transformer model, we used the "base" transformer configuration, which has a hidden size of 512, 6 hidden layers, and 8 heads. For all experiments we used a vocabulary size of 30k, except for the SPM model we used a vocabulary size of 8k.

3.4 Results

Unsurprisingly, the SPM model outperformed all other models as can be seen in table 2. The attention model shows expected significant performance improvement over the baseline model. And, both the SPM and the Transformer models shows expected significant performance improvement over the attention model. The Transformer, also, uses a subword encoding, so that may explain the spike in the performance. However, the CopyNet model performed the worst, surprisingly. We believe something went wrong during the experiment, however, we didn't have time to debug it. Normally, we wouldn't report the results since they are not interesting; however, since grading in this course takes effort into account, we decided to add them. Unfortunately, there are no other studies that attempted the same dataset or task in Arabic that we can benchmark with.

	Baseline	**Attention**	**CopyNet**	**SPM**	**Transformer**
ROUGE	26.91	33.75	23.80	**37.23**	36.74
BLEU	9.58	14.70	7.547	**17.58**	17.60

Table 2: Model performance on the test dataset

4 Analysis

We have seen that UNK tokens are very common in generated headlines for word-level encoding. Nevertheless, most generated titles are very related to article which indicates that the models are, at least partially, able to understand the articles. Models with attention (all except the baseline) seem to do a much better job at identifying the key ideas of the article as can be seen in the samples below.

Source	Headline
Human	The Custodian of the Two Holy Mosques leaves Japan.
Baseline	The Custodian of the Two Holy Mosques receives the chairman of Japan.
Attention	The Custodian of the Two Holy Mosques leaves Japan after an official visit.
CopyNet	The Custodian of the Two Holy Mosques leaves <unk><unk><unk>.
SPM	The Custodian of the Two Holy Mosques leaves Japan after an official visit.
Transformer	The Custodian of the Two Holy Mosques leaves Japan after an official visit.

Table 3: Sample (1): generated headlines (translated).

In table 3, we see a sample of an 'easy' headline. The baseline, as always, touches on the right topic (i.e. the King of Saudi Arabia and Japan did something); however, it struggles at getting the details right. The attention, the SPM, and the Transformer models seem to always get the details right for short and straightforward articles.

Source	Headline
Human	Aramco sets propane price in January to 435 dollars per ton.
Baseline	"Aramco" sets the price price <unk>- euro to 5.33 million dollars.
Attention	Aramco sets <unk>price in January contract to <unk>dollars.
CopyNet	Aramco: <unk><unk><unk><unk><unk><unk>Aramco.
SPM	Aramco sets propane price in January to 435 dollars per ton.
Transformer	Aramco sets propane price in January to 435 dollars per ton.

Table 4: Sample (2): generated headlines (translated).

In table 4, we show a sample of a fairly challenging article. The baseline this time produced a non-coherent headline, let alone inaccurate. The attention model seems to touch on the right idea, but struggles with the numbers and rare nouns (i.e. here the word "propane"). However, here where the SPM and the Transformer models shine for their subword encoding. Both models have gotten the headline exactly as the human-written one, and had no issue with out-of-vocab words or numbers.

Source	Headline
Human	More that 5515 citizen has applied to 2962 open jobs in education.
Baseline	Civil Service Ministry: jobs in education for men tonight.
Attention	Civil Service Ministry: <unk>applicatns to jobs in education.
CopyNet	Civil Service Ministry: <unk>to jobs in education <unk>.
SPM	Civil Service Ministry: more than 5515 applicant to jobs in education.
Transformer	Civil Service Ministry: we will announce new jobs in education at the 10th of Ramadan

Table 5: Sample (3): generated headlines (translated).

In table 5, we see a sample of a more challenging article. We note that all models seem to only try to do direct summarization of the main points of the first paragraph of the article and can't get any creative like the human-written headlines. Also, this sample confirms our observations that the baseline can only touch on the topic, at best. And, the attention model struggles with numbers. Here, the SPM and the Transformer models had totally different headlines; however, both seem like acceptable and relevant headlines.

5 Conclusion

We started knowing that summarization systems exhibit inaccurate reproduction of factual details, and inability to deal with out-of-vocabulary words, which is specifically true for Arabic. We found that attention mechanism is a great tool to get the details right. Moreover, we found that operating on a subword level is a tremendous help tin dealing with out-of-vocabulary words. Most news articles have numbers and rare nouns that cannot be reproduced without the help of a mechanism that extends the vocabulary size, like copying or byte-pair-encoding. Finally, we found that our models aim for direct reproduction of the main ideas in a piece of text. While that is acceptable in regular summarization tasks, it doesn't do as well in news headline generation since these require more creativity to hook the reader. Also, since creative headlines show less relevance to the article compared to summarizing headlines, predictive models that learns a mapping from the article to the headline, such as the Seq2Seq and the Transformer models, may not work very well in this task. Instead, one may explore conditional language generation with generative model like textual GANs to learn creative headline generation.

References

[1] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[2] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[3] Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

[4] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[5] M. Alhagri. Saudi newspapers arabic corpus (saudinewsnet), 2015.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[7] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[8] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*, 2017.

[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.

[10] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[11] Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

[12] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[14] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.