Headline Generation Using Recurrent Neural Networks

Heng-Lu Chang
Department of Electrical and Computer Engineering
The University of Texas at Austin
hengluchang@utexas.edu

Yung-Shen Chang
School of Information
The University of Texas at Austin
yscchang@utexas.edu

Linli Ding
School of Information
The University of Texas at Austin
linli.ding@utexas.edu

Jeremy Gin
Department of Electrical and Computer Engineering
The University of Texas at Austin
jgin@utexas.edu

ABSTRACT

Automatic text summarization is a long-standing task and a seemingly innately human one in many domains. We study and explore the application of neural network (NN) models, specifically a Long Short Term Memory (LSTM) model, to text summarization in the form of news headlines for news articles. We apply an encoder-decoder LSTM model with attention mechanism trained using the commercial cloud computing products, Amazon Web Services (AWS) Elastic Compute Cloud (EC2), using Google's deep learning framework, TensorFlow. Our primary dataset is the news collection by the Associated Press Worldwide from English Gigaword, a dataset of historical news articles and headlines from five reputable, international news agencies. We evaluate our work using a synthesis of user evaluation and system evaluation techniques. Users evaluate our model via original, precise user studies which allow us to statistically analyze human feedback and more accurately discuss our findings. Our model's preliminary results appear very promising according to both subjective user feedback and objective metrics. Based on our user evaluation, many machine-generated headlines were evaluated more favorably than actual headlines. We then provide our analysis of when and how system evaluation scores differ from user evaluation scores.

Keywords

Headline generation, Long Short Term Memory, Encoder-decoder model, Attention mechanism, User evaluation.

1. INTRODUCTION

Summarization is a useful technique for identifying the main idea and presenting information concisely and effectively. With the overwhelming amount of information available online, the demand for the text summarization becomes higher than ever. For the past half century, automatic text summarization techniques have been studied and improved with various approaches [1]. Recent advances in computing power and successful applications of recurrent neural networks makes it possible to improve the quality of machine summarization.

There are currently two methods of summarization, extractive and abstractive. Extractive is where the machine takes actual context and uses that as the summary. It is the equivalent of copying down the main points of a text without any changes. Abstractive is a technique used to mimic paraphrasing. This method makes summarization more condensed and human like.

It should be noted that there is a difference between the format of a summarization of a text and what newsreaders often view as a headline. Specifically, headlines are often eye-popping, contain urgent words like "Breaking!", "Live Update", or "Recent". However, based on our observations of Gigaword, this is generally not true of written news media like it may be in television news media. We subjectively observe that the headlines are very often factual, terse summarizations of the first paragraph. Hence, we are building our model with this insight in mind. Some documents do, in fact, include attention-grabbing phrases, and our model's training reflects that.

As a deep learning problem, writing good summaries is a noteworthy task. It is possible that needlessly including attention-grabbing words that do not add to the facts of the document only obscure the summaries. We maintain that news summarization is still an interesting problem space because it (1) inherently uses diverse proper nouns referring to people, places, and events happening internationally and (2) describes events and happenings that are so recent that no model could be accurately trained on them.

In this paper, we will explore ways to implement a recurrent neural network to automatically generate headlines that summarizes the main idea from the text of the news articles.

This work draws on the multiple technical backgrounds of each group member and seeks to build on topics of mutual interest. To state it broadly, we want to explore the effec-

DOI: 10.1145/1235

tiveness and feasibility of employing neural network models to text summarization in the news domain. Some of the questions we are interested in exploring are as follows. Can an encoder-decoder LSTM model perform better than existing methodologies of text summarization? Does the cloud, specifically AWS EC2, sufficiently address the memory and computation requirements of training LSTMs and other RNN models? To what extent can Google's TensorFlow deep learning framework, coupled with AWS, aid us in achieving a trained, effective model? Can user studies more holistically evaluate the dually qualitative and quantitative results of text summarization? How well does our encoder-decoder architecture perform based on user feedback? And most importantly, inspired by the Turing test, can users distinguish between machine-generated news headlines versus human-generated news headlines?

Through our work, we believe we have explored and begun to answer many of these questions. We believe that our work is significant because we design our user studies to prioritize human significance rather than simply mathematical significance. Based on our results, we found the need for user studies very convincing. We have come away from this semester with a deeper, hands-on experience with using commercial cloud computing products like, Google's TensorFlow, one of many machine learning and deep learning frameworks publicly available. We have a deeper appreciation for well-curated datasets and the processing that goes into data-intensive models. Finally, we have achieved a more practical understanding of using subjective user feedback as a solid metric by which to evaluate technical work.

In previous works discussing the evaluation of information retrieval methods, there has been debates regarding the pros and cons of using system-oriented evaluation or user evaluation [6]. User evaluation could not only be expensive but also time-consuming. Researchers has developed automatic evaluation such as Recalled-Oriented Understudy for Gisting Evaluation (ROUGE) [2] and Bilingual Evaluation Understudy (BLEU) [5]. Both evaluations measured the word pairs between computer-generated summaries and human-created actual summaries. Nevertheless, system-oriented evaluation also faced criticism for lacking user involvement and ignoring user interaction. User-centered evaluation could provide rich information of users' need and interaction with information. Hence, in the proposed study, we aim to use and compare the evaluation results of automatic evaluation and human evaluation on text summarization.

2. RELATED WORK

2.1 Text Summarization

Our summarization techniques draws from previous works in encoder-decoder LSTM and attention mechanisms. One example is Lopyrev [3], who used encoder-decoder recurrent neural networks with LSTM and attention to generate headlines from news articles. This is accomplished by feeding the Gigaword dataset as the input text of a news article into the encoder, one word at a time. Each word is then transformed into a distribution representation. The distribution representation is next combined using a multi-layer neural network with hidden layers generated after feeding in the previous word. The next step brings the hidden layers to the decoder. First, an end-of-sequence symbol is input into the decoder, which then uses the embedding layer to transform the symbol into a distributed representation. By use of the softmax layer and attention mechanism, the decoder generates each of the words of the headline, ending with the end-of-sequence symbol. After generating each word, the same word is re-fed in as input when generating the next word. The last step is also known as "Teacher forcing" [3], where the expected word in the headline is fed back into the decoder. Overall, Lopyrev's experiment was successful in generating a 50-word summary which was mostly valid and grammatically correct. However, there were some flaws that stood out. One example is that the neural network attempts to fill in details it is missing. For example, it would simplify the text from "72 people died when a truck plunged into a gorge on Friday" to "72 killed in truck accident in Russia" [3]. The model creates its own idea of the location of the accident. This was probably due to the small decoding beams used in the experiment, where the model stops decoding the sentence early [3].

Another related work is from Rush et al. [8]. Their work explores a data-driven approach for generating abstractive summaries. The encoder is modeled off of the "attention-based encoder which learns a latent soft alignment over the input text to help inform the summary" [8]. The method also incorporates a beam-search decoder as well as additional features to model extractive elements. Their results show that after training on the DUC 2004 dataset using ROUGE, it significantly outperformed several abstractive and extractive baselines [8].

Other advancements in extractive summarization by Gambhir and Gupta [1] proposed two new techniques for automatic summarization: Modified Corpus Based Approach (MCBA) and Latent Semantic Analysis-based TRM technique (LSA + TRM). MCBA is a trainable summarizer that depends on a score function and is able analyze important features for generating summaries like positive and negative keywords. The method uses two ideas: it denotes the importance of various sentence positions with a ranking system, and a Genetic Algorithm (GA) trains the score function to obtain a combination of feature weights. The other approach, LSA + TRM, first uses LSA to determine a documents semantic matrix. Then, TRM analyzes the relationship between sentences and builds a relationship map for the semantic text. The end result is a selection of sentences that best fits the document's summary.

2.2 Methods for Evaluating Summaries

The goal of summary evaluation methods is to determine how readable or useful a summary is compared to its original source. Summarization evaluation methods can mainly be divided into two categories: intrinsic and extrinsic. Most summarization evaluations use the intrinsic method, which is to have users directly judge the quality of the summarization by analyzing the summary. According to Steinberger and Jezek [9], intrinsic methods could be further divided into text quality evaluation and content evaluation. The concept of text quality measure is to have human annotators assess the grammaticality, reference clarity, and coherence and structure of the summary. Text quality measures would require human to be involved and cannot be done automatically. As for content evaluation, typically it is done by first comes up with an actual summary by a professional abstractor or merging summaries generated by multiple users. The

actual summary would then be compared with the output of the system-generated summary. Four of the most used content evaluation measures are precision, recall, F-score, and ROUGE. Precision, recall, F-score are standard measures for information retrieval while ROUGE measures the summary quality by counting the overlapping units between the candidate summary and actual summary. Summary informativeness is also another aspect that is mainly focused in intrinsic evaluation. Summary informativeness assesses how much information the generated summary has preserved compared to the original source.

On the other hand, extrinsic evaluation, which is also named as task-based evaluation, measures the efficiency and acceptability of the generated summary and how well the summary assists in performing a task instead of analyzing the content of the summary. For example, task-based evaluation would evaluate how the generated summary could answer certain questions relative to the source. Document categorization, information retrieval and question answering are three of the most important tasks that would need the use of extrinsic evaluation.

Based on the taxonomy of summarization evaluation, some of the evaluation methods would require users while other methods rely on the system. In past research, user studies had been conducted to generate summary or evaluate the quality of the summaries. In Radev et al. study [7], they built a multi-document summarizer and had users test the model. During the evaluation phase, they recruited six judges and computed the cross-judge agreement on cluster-based sentence utility (CBSU) and cross-sentence informational subsumption (CSIS) tasks. The result based on user study showed that interjudge agreement on sentence utility was very high. Another study was conducted by the US government to have users evaluate a summary system [4]. The evaluation had the user decide whether either a source or a user-focused summary was relevant to a topic. Overall, it is not uncommon but valuable to have a user evaluate the outcome of the automatic summarization. As a result, in our research we aim to have users assess the coherence, informativeness of the summaries, and the similarity between the actual summary and generated summary.

3. METHODOLOGY

3.1 Dataset

The model is trained using the English Gigaword 2nd edition dataset, as available from the University of Texas library. This dataset consists of several years of newswire text data from five major domestic and international news services: Agence France-presse English Service, Associated Press Worldstream English Service (APW), the New York Times Newswire Service (NYT), Central News Agency of Taiwan English Service, and the Xinhua News Agency English Service. As Gigaword contains many spurious headline-article pairs, we will filter the articles for training.

After working with a subset of the NYT from 2000-2004, we concluded that the preprocessing of the inconsistently formatted and poorly curated dataset was too arduous to use for our purposes. The most significant problems we encountered was that (1) many different articles were listed under identical headlines and (2) duplicate headline-article pairs. Both of these undermines the theory of machine learning which dictates that training and testing data are randomized and kept strictly separate during training and testing.

Hence, we ultimately used a subset of the APW's data from 1994-2004. We found this dataset to be free of the aforementioned problems and therefore optimal for our use.

3.2 Data Preprocessing

This is the process which we will call data preprocessing. In its raw form, the English Gigaword dataset contains hundreds of compressed files distributed across five directories for the five news organizations. Each compressed file contains on the order of 10,000 articles in Standard Generalized Markup Language (SGML) format, an HTML-like markup language. Sections in each datum include: document metadata (unique ID and document type), the headline of the news article, and multiple paragraphs of text which comprise the article.

We use a markup language parsing Python library called BeautifulSoup4. This enables us to parse the data in a way that is flexible to our needs. First, our Python scripts read in all of the uncompressed files. Because our methodology will only be influenced by the headline and the first paragraph of text, the preprocessing step requires stripping each article of everything but the headline and the first paragraph. We do this because this paragraph most significantly influences the headline. We no longer need the unique IDs in the metadata because the document's index into our Python list is enough to identify it in our data structures.

We further truncated our dataset based on a number of criteria which availed themselves throughout the preprocessing and training process. We detail these and the underlying reasoning below.

Included in this dataset are non-human-readable headlines including non-English words or grammatically incorrect headlines. Examples of this from the dataset are the following headlines: "NYTR-PRIU;UAFOOTBALLNOTE", "AM-NYT-BUDGET", and "COX-ARAFAT-OBIT-TAKE3". These articles are not realistic representations of our problem space. Additionally, many articles have very short first paragraphs, due most likely to anomalous or unusual formatting in only a handful of articles. Examples of this from the dataset are the following first paragraphs: "KINSEY", "Washington Bureau", and "Attn Sports Editors:". We further truncate our dataset by throwing out any headline-paragraph pairs whose headlines contain the "-" character or whose first paragraphs do not exceed three words in length. We found a decent number of headlines which contain "(" and/or ")". Many of these included unclosed parentheses, and we deem the rest generally unfitting for an accurate summarization. We truncate all such headlines from our training data. Finally, we found many articles to be inconsistently SGML-tagged or missing important tags. These headline-article pairs comprise the last of our truncation.

The preprocessing step tailors our dataset for our methodology and our problem space. Preprocessing also formulates and stores the data in the appropriate data structure for our model. At this point, our preprocessing creates and populates two lists: one storing each headline and the other storing each corresponding first paragraph. By extracting only the useful information, this step optimizes memory usage and the complexity of interacting with data. Ultimately, this leads to shorter, more efficient training and more accurate testing.

We build 2 Python lists with the remaining processed

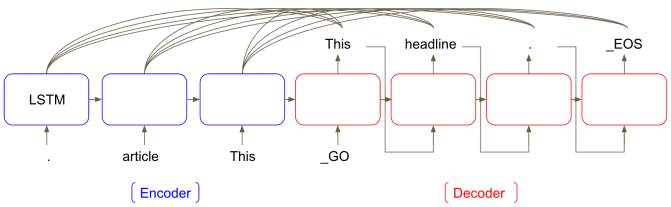

Figure 1: Encoder-decoder LSTM model with attention mechanism. "This article." is the input sentence and "This headline." is the output sentence. _GO is the start of decoding token and _EOS is the end of sentence token.

headlines and articles. During the truncation step in preprocessing, we truncate: 296,471 pairs due to "-", 2,086 pairs deemed too short, 278,167 pairs due to inconsistent tags, and 3,535 pairs due to "(" or ")". Our final training and testing dataset contains 1,344,565 headline-sentence pairs. All 1.3 million of the pairs come from the APW 1994-2004 dataset.

3.3 Word Length Bins

During the training step, we found it helpful and necessary to fine tune the model to be trained differently on headlines, sentences, and headline-sentence pairs of different lengths and combinations of lengths. For example, when summarizing articles of 50-60 words in length, the model should draw on knowledge garnered from training articles of 50-60 words in length. We therefore instituted part of the preprocessing step to include gathering frequency statistics for headlines and sentences of different lengths. We use the following thresholds to create 9 sentence length bins: 20, 30, 35, 40, 45, 50, 60, 70. We use the following thresholds to create 5 headline length bins: 10, 12, 15, 18. We create histograms (see figure A1 to A5 in Appendix A) to inform our training bins. Our goal is to minimize padding for headlines and sentences when input into the LSTM by optimizing bin sizes.

3.4 Encoder-decoder LSTM Model

We first familiarized ourselves with the Long short term memory (LSTM) model, which is a recurrent neural network we used for both encoder and decoder. We created a toy LSTM model for number sequence prediction and uploaded the source code[1] to GitHub. We use the encoder-decoder LSTM model with attention mechanism to generate a headline given the first sentence of a news article. Encoder-decoder model is a sequence to sequence model which are widely exploited in tasks including machine translation [10], chat-bot [11] and text summarization [3]. Our work will be closely related to [3] in which the author also used encoder-decoder LSTM model with attention mechanism to generate headline. The largest difference between our paper to [3] are two fold. First, we use five buckets and optimized bucketing length to minimize the number of paddings used in the sentences while [3] only use one bucket. Second, we use a greedy decoder rather then a beam search decoder to generate headline words.

An encoder-decoder LSTM model with attention is shown in Figure 1. This model contains two parts. The first part is a LSTM encoder which encodes an input sentence "This article." The second part consists of a decoding LSTM which will generate an output sentence "This headline." In this paper, we use the first sentence of the news article as input and its corresponding news headline as output. We fed the input words in reverse to the encoder so that the first few words in the input sentence will be closer to the first few output words to capture short term dependencies. Reversing the input sentence has shown better results in [10] in machine translation task. After being generated, each predicted word will be fed as input when generating the next word. Using Figure 1 as an example, "This" is the first generated word which will be fed into the input to predict the next word "headline". The attention mechanism is to give weights on both the output hidden states in the encoder and the current hidden state to decide which word to pay attention to while predicting the current headline word. Attention mechanism was implemented in [10] as a decoder for machine translation and in [8] as a encoder for sentence summarization.

We used TensorFlow library to implement our encoder-decoder LSTM model with attention mechanism. Our code is build from both sequence-to-sequence model[2] from TensorFlow and a GitHub repository[3] where the author also utilized sequence-to-sequence model to build a chat-bot. Specifically, we use three LSTM hidden layers in both encoder and decoder model, an output dropout rate of 0.2 between layers, 512 hidden units per LSTM cell, and we trained word embeddings to 512 dimensions. We embedded the most frequent 80,000 words in both sentences and headlines, and label all other words as unknown words (_UNK) or often known as out-of-vocabulary (OOV) words. Lastly, we use five buckets ((30, 10), (30, 20), (40, 10), (40, 20), (50, 20)) (i.e, five different encoder-deocder LSTM model) based on the statistics acquired in section 3.3 to reduce the number of padding tokens. In bucket representation (a, b), a refers to the length of the sentence and the b is the length of headline. If a sentence is longer then 50 tokens or a headline longer then 20 tokens, we automatically place it in the last ((50, 20)) bucket. We use a special token (_PAD) to pad both the sentences and headlines to its bucket size. The source code[4] of our model will be uploaded to GitHub soon.

4. RESULTS

We splitted the 1.3 million articles and headlines pairs acquired from 1994-2004 APW newswire into 70% training set, 20% evaluation set, and 10% testing set after preprocessing discussed in section 3.2. We used batched training of size 128, a learning rate of 0.5 using gradient descent optimizer with a decay factor of 0.99 and set the maximum gradient norm to 5. We then train our model on AWS EC2 g2.2xlarge (NVIDIA K520 with 15G memory) instance for 10 epochs, which took around 20 hours.

We show three good quality predictions and three poor quality predictions in table 1. All six sentences are in the testing set and were never seen before from the training or evaluation set. In the first three good quality predictions, we see that our model can well-summarized the sentence and sometimes generating new words that are not in the

[1]https://github.com/hengluchang/NumberSequencePrediction

[2]https://github.com/tensorflow/tensorflow/blob/master/tensorflow/models/rnn/translate/seq2seq_model.py

[3]https://github.com/suriyadeepan/easy_seq2seq

[4]https://github.com/hengluchang/newsum

Table 1: Examples of predicted headlines

Sentence	Actual Headline	Predicted Headline
1. President Vladimir Putin announced plans Monday for the Russian military to hold exercises in the former Soviet republic of Kyrgyzstan, the ITAR-Tass news agency reported	Russia plans military exercise in Kyrgyzstan	Putin to launch military exercises in Kyrgyzstan
2. Japanese Foreign Minister Keizo Obuchi arrived in Moscow on Saturday for talks aimed at laying the groundwork for a bilateral peace treaty between Russia and Japan.	Japanese foreign minister arrives for talks with Russian	Japanese foreign minister arrives in Moscow for talks on peace
3. South Korea, responding to a U.N. appeal, will donate dlrs 6 million in emergency food aid to impoverished North Korea, government officials said Thursday.	Seoul to donate dlrs 6 million for North Korea food aid	South Korea to provide food aid to North Korea
4. Former cycling track Olympic champion Stefan Steinweg has been banned for two years on a doping charge, the German cycling union said Wednesday.	Former cycling Olympic champion banned for doping	Former Olympic champion banned for doping
5. A light plane bringing visitors back from a hunting lodge crashed in western Canada, killing three people including a Colorado state legislator who was piloting the aircraft.	Three Die in Canada Plane Crash	Three people killed in Canada bus crash
6. Three U.S. soldiers were killed and six more injured in a traffic accident in northern Iraq, the military said Saturday.	three U.S. soldiers dead, six injured in road accident	U . S . soldiers killed in car accident in

sentence. We also observed some poor quality predictions and we picked three of them to represent three different issues. First, the predicted headline tend to miss important information like missing "cycling" before "Olympics" in example 4. Secondly, the model filled in details that is not correct like "bus crash" instead of the actual "plane crash" in example 5. Lastly, several headlines was forced to end before it hits the end-of-sentence token due to bucketing issue like example 6. The last issue can be solved by changing to a larger bucket whenever the end-of-sentence token has not been hit. For future work, we think that visualizing the attention weights of each words can help us understand more with the issues of missing information and filling incorrect details.

5. EVALUATION

In our study, we implemented an encoder-decoder LSTM model. To evaluate the outcome of the proposed model, we used both system-oriented automatic evaluation and user-oriented evaluation and compared the results. For user evaluation, we set up an one-factor within-subject design evaluation survey[5]. The factor would be the methods of summary generation: actual summary and summary generated by LSTM model. In our experimental design, we use a Google Form to design and distribute the survey. The survey consists of two major parts. The first part asked for the users' demographic information such as gender, age, nationality, and native language. The purpose of asking the users' nationality and native language was to later confirm that native language difference had no effect on the evaluation. The second part was the evaluation phase. During the evaluation phase, users were first asked to read through a news sentence from the dataset. Then users were provided with two headlines: actual headline and generated headline. Users respectively evaluated how well they thought the two headlines summarized the news sentence on a five-point Likert scale (with 1 meaning very poor and 5 meaning 5 well). Afterward, users would evaluate the similarity between the computer-generated summary and actual headline on a five-point Likert scale (with 1 meaning very dissimilar and 5 meaning very similar). Last of all, user are asked to select which of the two summaries best depicted the focus of the news. Overall ten news articles are provided to the users. The procedure of the user evaluation is shown in Figure 2. Overall, it would take approximately 5 to 10 minutes to complete the evaluation. The survey was distributed via researchers' social media, canvas, School of Information's e-mail lists.

As for system-oriented automatic evaluation, BLEU was used to evaluate the similarity between actual headline and predicted headline. BLEU is a precision-based measure. The uniqueness of BLEU is that it measures the percentage of overlapping n-grams between the candidate summary and reference summary. In our evaluation, we used k = 1 (unigram) to evaluate the similarity between two headlines. Also, in order to compare the relationship between the result of system evaluation and user evaluation, we chose news sentences that had BLEU score ranging from 0.1 to 0.8, instead of just selecting news that had the best BLEU score for user evaluation.

To analyze the data, paired samples t-test was used to compare users' judgment of how well they thought the two headlines summarized the news sentence. Spearman's rank-order correlation was used to measure the correlation between system and user evaluation on the similarity between two headlines. Last of all, frequency was shown as which headline did the users considered as the better headline.

6. DATA ANALYSIS

6.1 Descriptive Analysis

72 users participated in the evaluation. However, missing

[5]https://goo.gl/forms/hNI77c5NiViaY7YF2

Table 2: Paired sample t-test comparison between user's evaluation on how well the two headlines summarized the news sentence.

	Predicted Headline		Actual Headline			
News	Mean	S.D.	Mean	S.D.	t	p
News 1	4.17	0.81	2.93	1.087	7.579	0.000***
News 2	3.76	0.963	2.93	0.961	4.416	0.000***
News 3	3.65	1.184	4.04	0.818	-2.429	0.018*
News 4	3.3	1.176	3.48	1.107	-0.881	0.381
News 5	3.92	0.906	3.69	1.226	1.139	0.258
News 6	4.34	0.716	2.87	1.095	10.786	0.000***
News 7	3.52	0.969	3.93	1.019	-2.163	0.034*
News 8	4.42	0.601	3.35	1.084	8.734	0.000***
News 9	3.27	1.068	4.18	0.743	-6.344	0.000***
News 10	3.76	0.978	3.92	1.143	-0.798	0.428
Overall News	3.81	1.023	3.53	1.133	4.504	0.000***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 3: System and user evaluation on the similarity between predicted and actual headline.

	Predicted Headline	Actual Headline	
News	BLEU	Mean	S.D.
News 1	0.429	3.27	0.940
News 2	0.118	2.75	0.982
News 3	0.429	3.68	1.144
News 4	0.441	2.89	1.41
News 5	0.445	3.20	1.103
News 6	0.600	2.99	1.007
News 7	0.667	3.14	0.899
News 8	0.800	3.70	1.139
News 9	0.833	3.55	1.053
News 10	0.429	3.45	1.131
Overall News		3.26	1.097

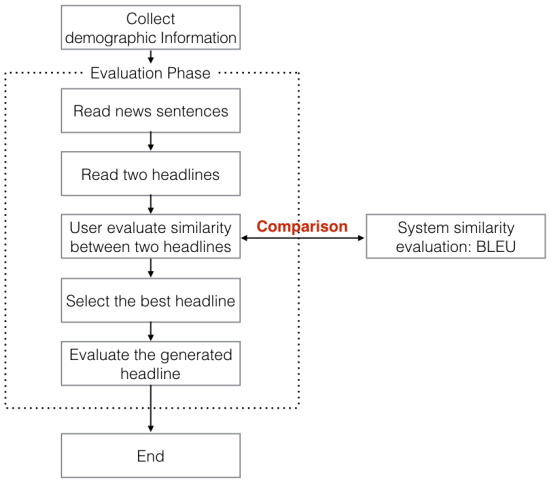

Figure 2: Procedure of the user evaluation. The user evaluation consists of two phases: demographic information and summary evaluation.

data was found in one of the user's response. Hence, overall 71 data were used for analysis. Among the 71 users, 42 were female (59 %) and 29 were male(41%). The average age was 28.52 (standard deviation 9.020). For user's nationality, 34 were from United States (47.8%), 29 were from Taiwan (40.8%), 2 Mexicans (2.8%), 2 Chinese (2.8%), 2 Indians (2.8%), 1 Canadian (1.4%), and 1 Malaysian (1.4%). As for their native language, 35 users' native language were English (49.2%), 33 were Chinese (46.4%), 2 were Hindi (2.8%), and 1 was Spanish (1.4%).

6.2 Comparison on Predicted and Actual Headline Evaluation

Before conducting data analysis, we first wanted to analyze whether native language difference would affect the evaluation outcome. Since most of the users either speak English or Chinese as their native language, we used independent sample t-test to analyze whether language difference would affect users' evaluation on how well the actual headline and predicted headline summarized the news sentence. The result showed that significant difference was found on predicted headline ($t^6 = 1.973$, $p^7 < 0.05$) but not on actual headline ($t = 0.920$, $p = 0.358$). Although significant difference was found on actual headline evaluation, the mean score of English users and Chinese users were respectively 3.90 and 3.74. Since the sample size for data analysis was 350,

[6]t = The t-statistic for a paired sample t-test
[7]p = The p-value (probability value) for the t-statistic

Table 4: Number of users and percentage of selection on the better headline between the two headlines.

	Predicted Headline		Actual Headline	
News	N	%	N	%
News 1	52	73.2	19	26.8
News 2	50	70.4	21	29.6
News 3	27	38.0	44	62.0
News 4	27	38.0	44	62.0
News 5	39	54.9	32	45.1
News 6	65	91.5	6	8.5
News 7	25	35.2	46	64.8
News 8	63	88.7	8	11.3
News 9	15	21.1	56	78.9
News 10	32	45.1	39	54.9
Overall News	395	55.6	315	44.4

large sample size could easily amplify the statistic results, causing insignificant results turning into significance. As a result, we concluded that native language difference have no significant effect on the overall user evaluation. Hence, we used 71 data to perform the following data analysis.

We first used paired sample t-test to compare users' evaluation on how well they thought the two headlines summarized the news sentence. Each news and the news overall score were analyzed individually. As shown in Table 2, There were significant differences on how users thought the predicted headline and actual headline summarized the news sentence on news 1 (t = 7.579, $p < 0.001$), news 2 (t = 4.416, $p < 0.001$), news 3 (t = -2.429, $p < 0.05$), news 6 (t = 10.786, $p < 0.001$), news 7 (t = -2.163, $p < 0.05$), news 8 (t = 8.734, $p < 0.001$), news 9 (t = -6.344, $p < 0.001$), and overall news (t = 4.504, $p < 0.001$). As for news 4 (t = -0.881, $p = 0.381$), news 5 (t = 1.139, $p = 0.258$), news 10 (t = -0.798, $p = 0.428$), no significant difference were found. Based on the predicted and actual headline mean value, users evaluated that the predicted headline summarized better than the actual headline on news 1, news 2, news 6, news 8, and the total scores on overall news. On the other hand, users thought that actual headline summarized better on news 3, news 7, and news 9.

6.3 Comparison on System and User Similarity Evaluation

The result of the system and user evaluation on similarity between actual and predicted headline is shown in Table 3. The BLEU score range from 0.118 to 0.800 while the user evaluation (mean = 3.26) range from 2.75 to 3.75 on a 5 point scale. To analyze whether there was any correlation between the predicted headline and the actual headline, we used Spearman's rank-order correlation. The result showed that there was weak and positive correlation between the system and user evaluation between the two headlines (r = 0.101, $p = 0.007$), indicating that the system and the user evaluation does not share consistent results.

6.4 Selection of Best Headline

At the end of each evaluation, users were to select which of the two headlines best depicted the news sentence. The number and percentage of users' selection on both headline are shown in Table 4. More users chose predicted headline better depicted the news sentence on news 1 (73.2%), news 2 (70.4%), news 5 (54.9%), news 6 (91.5%), news 8 (88.7%). On the contrary, more users seem to agree that the actual headline was better than the predicted headline on news 3 (62%), news 4 (62%), news 7 (64.8%), news 9 (78.9%), and news 10 (54.9%). Taken account of the overall news, 55.6 percent of the users agreed that the predicted headline depicted the news sentence better than the actual headline (44.4%).

7. DISCUSSION

In this study, we conducted both system and user evaluation on the similarity between the actual headline and the predicted headline. Also, we had users evaluated how well did both headlines summarized the news sentence. As the results showed, among the ten given news sentences, users thought that 4 of the predicted headlines summarized the news sentence better than the actual headline, while users thought 3 of the actual headline were better. Therefore, users seems to agree that only parts of the machine-generated headlines were better than the actual headline. In order to further understand the relationship between the system evaluation and user evaluation of summaries similarity, we analyzed the correlation of both evaluations. According to the result, system evaluation and user evaluation had weak and positive correlation. This could also be shown while selecting the 4 news that users thought the predicted headline was better in Table 2. Based on Table 4, 73.2%, 70.4%, 91.5%, and 88.7% of the users selected predicted headline as the better headline respectively on news 1, 2, 6, and 8. However, according to the BLEU metrics (see Table 3), news 1, 2, 6, and 8 scored 0.429, 0.118, 0.6, and 0.8. Apparently, among the 4 news that users evaluated as the better headline, their BLEU score ranged from 0.118 to 0.8, showing no consistent trend.

Based on our results, it is not suitable to merely rely on system evaluation while evaluating the similarity between human-generated summary and machine-generated summary. We believe there is a convincing need for user evaluation to play a significant role in the task of evaluating headline summarizations. The BLEU metrics poorly capture sentiment similarity. For example, the BLEU metrics give very high scores to headlines which differ in only one word. However, sometimes the words which differ are important numbers or the machine-generated word is slightly grammatically incorrect. These are telltale signs of a machine-generated headline which only reveal themselves in a word or a few characters. An example of an errant number in a

machine-generated headline can be observed in the following machine-generated headline: "How Women ' s Basketball Top 00 Fared". The human-generated counterpart is: "How Women's Basketball Top 25 Fared". The BLEU score given to the machine-generated headline was 0.875, where 1.0 represents 100% similarity.

Conversely, the BLEU score can be low for a machine-generated headline which accurately captures the article's sentiment but leaves out unnecessary details or is shorter than the human-generated headline. One example of this is for the following article: "CAIRO, Egypt _ Arab foreign ministers searched Monday for ways to invigorate their 22-member organization to make it more formidable in the face of regional challenges, and planned to submit a proposal to their leaders to make broad changes during their summit in Tunisia later this month." The machine-generated headline is: "Arab foreign ministers discuss Arab security". The human-generated headline is: "Arab foreign ministers meet to discuss reforming the Arab League". The BLEU score is 0.43. This is a relatively low BLEU score, but the title accurately captures the article's sentiment; it is simply shorter than the human-generated one.

As a result, it could explain why the BLEU score range widely among the news that users thought the predicted headlines were better than the actual headline. It could also explain why users did not select the predicted headline as the better headline on some news sentence or no significant evaluation differences were found among the two headlines.

8. CONCLUSION

In this paper, we implemented an encoder-decoder LSTM model with attention mechanism to generate headlines. We trained our model using APW dataset and evaluate our results using BLEU and user evaluation. Our results showed weak and positive correlation between BLEU and user evaluation, and 55.6% of the users picked the machine generated headline over human written headline for better summarizing the news. We also conclude that user evaluation is necessary to complement BLEU in evaluating headline summarizations.

9. REFERENCES

[1] M. Gambhir and V. Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, pages 1–66, 2016.

[2] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[3] K. Lopyrev. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712*, 2015.

[4] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim. The tipster summac text summarization evaluation. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 77–85. Association for Computational Linguistics, 1999.

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. 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.

[6] D. Petrelli. On the role of user-centred evaluation in the advancement of interactive information retrieval. *Information processing & management*, 44(1):22–38, 2008.

[7] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[8] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[9] J. Steinberger and K. Ježek. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275, 2012.

[10] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[11] O. Vinyals and Q. Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

APPENDIX

A. HISTOGRAMS OF LENGTH OF SENTENCE

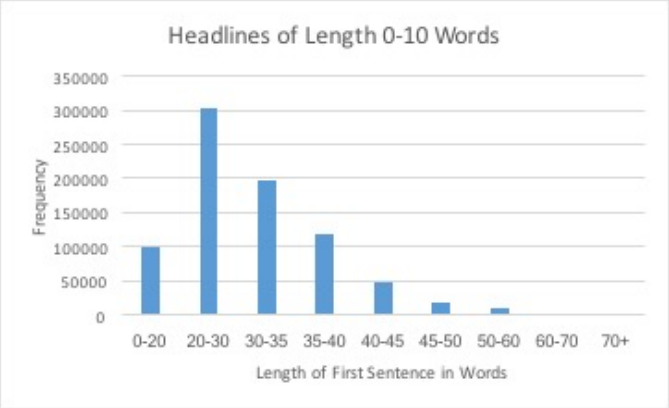

Figure A1: This histogram records the frequencies of headline-sentence pairs with headlines of length 0-10 words and varying sentence lengths.

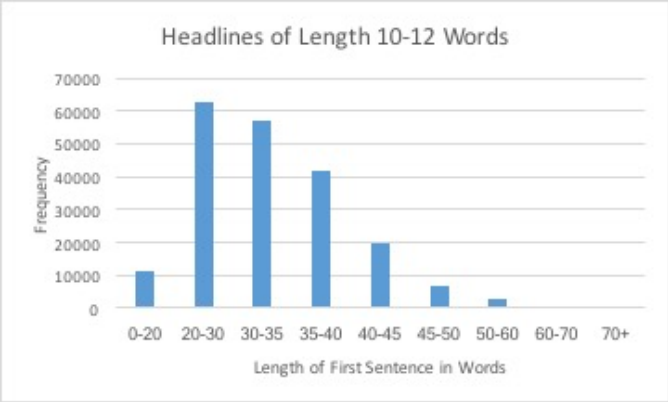

Figure A2: This histogram records the frequencies of headline-sentence pairs with headlines of length 10-12 words and varying sentence lengths.

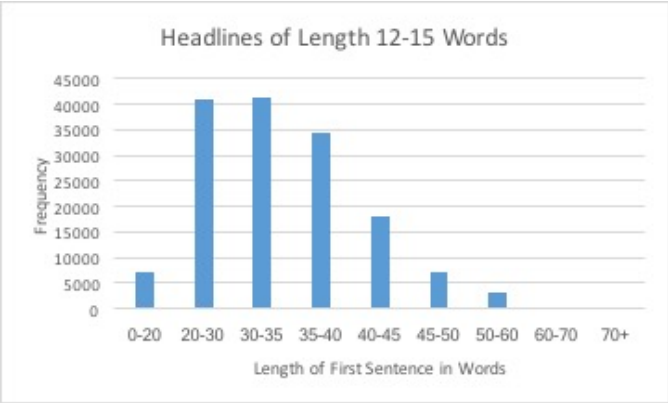

Figure A3: This histogram records the frequencies of headline-sentence pairs with headlines of length 12-15 words and varying sentence lengths.

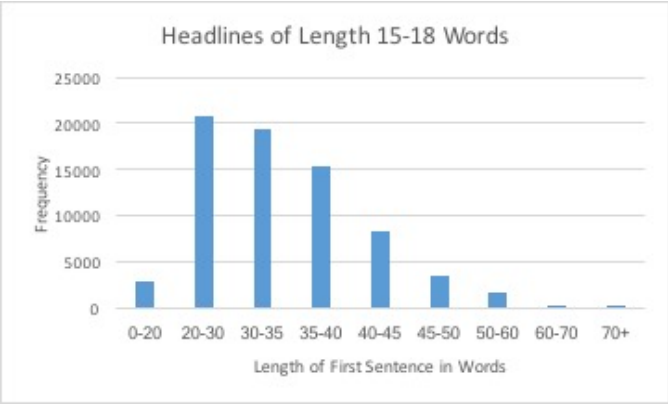

Figure A4: This histogram records the frequencies of headline-sentence pairs with headlines of length 15-18 words and varying sentence lengths.

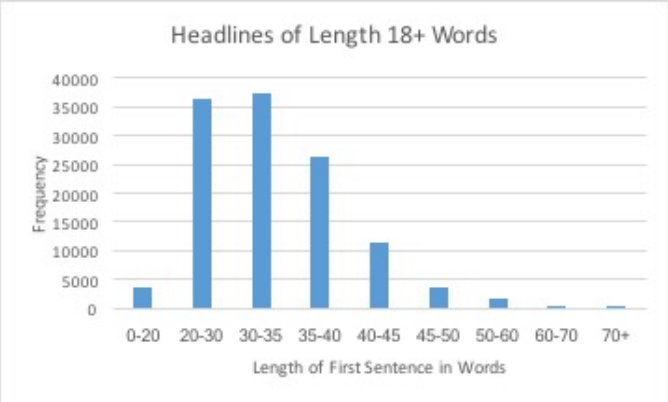

Figure A5: This histogram records the frequencies of headline-sentence pairs with headlines of length 18 or more words and varying sentence lengths.