Detection and Rephrasing for Chinese Clickbait News Headlines

Abstract

Nowadays, "Clickbait" floods the Internet world, and people are dissipated of time and energy by all kinds of eye-catching headlines. The previous research mainly focused on the detection of "Clickbait", and we put forward a complete solution for the problem of the proliferation of "Clickbait": Detection and Rephrasing. Firstly, we propose two methods, LSTM with hot words knowledge and BERT with hot words knowledge. We found that the hot words knowledge improves both the LSTM and BERT in detection. Secondly, we developed the "Clickbait" news headline rewriting task and fine-tuned the pre-training model for it. We consider the architecture of the transformer-based pre-training model and propose two methods as Decoder-Only (GPT-2) Model and Encoder+Decoder(Bert+GPT-2) Model. The Encoder+Decoder Model performs better in generating non-clickbait while the Decoder-Only Model performs better in summarizing.

1 Introduction

Before the era of Internet Media, formal accuracy surveys for news have been conducted. Back to 1936, Charnley [1], who mailed 1,000 news items clipped from three Minneapolis dailies to persons named in the stories, asking for their perceptions of inaccuracies. For these years, the role a misleading headline plays has been noticed [2], and been intensified since the current state of online media is one that heavily incentivizes speed and spectacle over restraint and verification.[3]

Clickbait, which is used for describing a headline, image or social media post that baits (or lures) people to click through to an article or video, has really become a problem.[4] CS-based methods for processing the problem was needed. In this paper, we divide the problem into two parts:
1) Detection: detect the clickbaits from given news (including their contents and headlines).
2) Rephrasing: rephrase the "non-clickbait" headlines from the given contents.

2 Related Works

The researchers of machine learning have done some works on both Detection and Rephrasing which we have defined.
Detection: Based on a random forest classifier, Potthast et al.[5] developed a news "clickbait" detection model. Cao et al.[6] filtered out many unrelated news features and using random forest regression for detection. Wei and Wan [7] constructed a corpus of news content with headlines and then used the SVM-based machine learning technique for ambiguous
or misleading headline identification. The dataset they built is exactly what we used in this paper as the benchmark data.

**Rephrasing**: There are numerous works on headlines generation. Rajalakshmy and Remya [8] extract all possible bigrams from the news content and selected sentences of most frequent bi-grams as the headlines. Colmenares et al. [9] presented a sequence prediction technique for learning how the headline of news stories can be generated. Ayana et al. [10] incorporate syntactic and semantic features for headline generation task. Tan et al. [11] used document summarization technique and hierarchical attention methods to select meaningful sentences for generating news headlines. Liu et al. [12] first considered the use of key phrases that attract users. Multi-source transformer decoder is employed to generate multiple key phrases of the news, and then generate key-phrases-relevant news headlines. [13]

Apparently, the progress made in the field of detection haven’t been properly followed up by headline rephrasing. And the earlier work of headline generation didn’t take the clickbaits factors into their consideration.

3 Problem Defining and Dataset

3.1 Problem Defining

Although we may have an empirical sense of clickbait, defining it formally should be where we start. Subject to the previously published dataset, Corpus[7], news headlines are classified into the three categories:

**Accurate Headlines**

An accurate headline is a headline that is congruent in meaning with the content of the news story.

**Ambiguous Headlines**

An ambiguous headline is a headline whose meaning is unclear relative to that of the content of the story.

**Misleading Headlines**

A misleading headline is a headline whose meaning differs from that of the content of the story.

**Example:**

*Contents*: Amazon’s AWS has acquired the encrypted messaging app Wickr for an undisclosed sum. AWS will be offering Wickr services effective immediately and Wickr customers, channel, and business partners would be able to continue to use Wickr’s services as they do now. Wickr has raised just under 60 million in funding to date, according to PitchBook data.

**Accurate Headlines**: Amazon acquires encrypted messaging app Wickr for undisclosed sum.

**Ambiguous Headlines**: Amazon enters the encryption field.

**Misleading Headlines**: Amazon secretly enters the encryption field.

In this paper, "Clickbait" news headlines are defined as the news headlines that are ambiguous or misleading.

3.2 Dataset

The dataset we use is the expanded version of Chinese "Clickbait” news data set constructed by Wei Wei. [7] It is composed of a total of 40000 articles in six different domains (domestic, world, society, entertainment, sports, and technology) from four major Chinese news sites (Sina, NetEase, Tencent, and Toutiao). 2,924 labeled data were provided. There are 645
ambiguous news and 843 misleading news in the labeled data. By translating English "Clickbait" word list and manually extend it to obtain better chinese adaptation, Wei Wei built a vocabulary of Chinese "Clickbait" hot words, which has been extended in this paper. As for the generation part, 7,842,149 news including headlines and contents have been gathered. 29,000 were used in this paper.

4 Detection

In the detection part, we have used two separate methods: 1) LSTM as the SOTA one and 2) BERT as the pre-training approach.

4.1 LSTM

Recurrent neural networks with Long Short-Term Memory (which we will concisely refer to as LSTMs) have emerged as an effective and scalable model for classifying. The central idea behind the LSTM architecture is a memory cell which can maintain its state over time, and non-linear gating units which regulate the information flow into and out of the cell.[14] (See Figure One)

![Figure 1: Detailed schematic of the Simple Recurrent Network (SRN) unit (left) and a Long Short-Term Memory block (right) as used in the hidden layers of a recurrent neural network.[14]](image)

4.2 Detection with LSTM

4.2.1 Accuracy

By setting different random seeds, we have conducted 5 experiments in all, and got 0.8125 as our best accuracy and 0.7966 as the mean value. The results are shown in the Table 1.

<table>
<thead>
<tr>
<th>Index</th>
<th>Random Seed</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>996</td>
<td>0.7922</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0.7953</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td><strong>0.8125</strong></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.7906</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.7966</td>
</tr>
</tbody>
</table>

| Mean  | —           | 0.7966    |

Table 1: Comparison Results of Detection
4.2.2 How much the Hot Words impact

By comparing the LSTM models with hotwords and without hotwords, we can verified that the hotwords really improves the performance of the model. To formalize it, we assume that the Accuracy of the model is subject to a normal distribution over the random seeds. Under the additional independence assumption, the hypothesis testing could be conducted.

Table 2: Comparison Results of Detection

<table>
<thead>
<tr>
<th>Index</th>
<th>Random Seed</th>
<th>X</th>
<th>Y</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>996</td>
<td>0.7922</td>
<td>0.7813</td>
<td>0.0110</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0.7953</td>
<td>0.7652</td>
<td>0.0281</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td><strong>0.8125</strong></td>
<td>0.7703</td>
<td>0.0422</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.7906</td>
<td><strong>0.7828</strong></td>
<td>0.0078</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.7966</td>
<td>0.7531</td>
<td>0.0391</td>
</tr>
</tbody>
</table>

As seen in Table 2, $X_i$ is the $i$th Accuracy with Hotwords while $Y_i$ is the $i$th without Hotwords, $D_i$ is $X_i-Y_i$. Under our assumption,

$$D_i \sim N(\mu_D, \sigma_D^2). i = 1, 2, 3...n$$

We can derived the Statistical Quantities as

$$t = \frac{\overline{d}}{S_D\sqrt{n}}$$

Here in our experiment, $t_0 = 3.632$, with $p$ between 0.01 and 0.005, which verifying that the hot words significantly improves the performance of the model.

4.2.3 Word/Char-Level tokenization

In natural language processing, tokenization is the process of breaking human-readable text into machine readable components. The most obvious way to tokenize a text is to split the text into words, while subword tokenization serves as the alternative approach. We examined the difference between Word-Level tokenization and Char-Level tokenization [20] and found that the Word-Level tokenization proceeds to be better as seen in Table 3 with a $p < 0.005$.

Table 3: Comparison Results of Word/Char-Level tokenization

<table>
<thead>
<tr>
<th>Index</th>
<th>Random Seed</th>
<th>Accuracy-Word</th>
<th>Accuracy-Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>996</td>
<td>0.7922</td>
<td>0.7281</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0.7953</td>
<td>0.7500</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td><strong>0.8125</strong></td>
<td>0.7609</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.7906</td>
<td><strong>0.7750</strong></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.7966</td>
<td>0.7484</td>
</tr>
<tr>
<td>Mean</td>
<td>—</td>
<td>0.7966</td>
<td>0.7525</td>
</tr>
</tbody>
</table>

4.3 Detection with BERT

4.3.1 Over Review of BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers. It is a pre-trained model with a transformer encoder architecture. Back to 2013, transfer learning in computer vision occurred. The most renowned examples of pre-trained models are the computer vision deep learning models trained on the ImageNet dataset. In 2017, the need for transfer learning in NLP was at an all-time high. And transformers came. They take the entire sequence as input in one instead of token by token.
as what RNN like model did. Besides, it means that we don’t need labels to pre-train models. Thus, BERT came as the first successful huge pre-trained model for NLP. (See Figure 2)

Figure 2: Overall of BERT: Pre-Training and Fine Tuning[15]

4.3.2 Input/Output of BERT

[CLS] and [SEP] are two tokens of the basic compositions of the input. BERT was pretrained using the format:


It is necessary for the Next Sentence Prediction task: determining if sen B is a random sentence with no links with A or not. The [SEP] in the middle is here to help the model understand which token belong to which sentence. As for [CLS], it is the first position of the output which we cared for classification. (See Figure 3)

In this paper, we set the input as the concatenation of three segments with special separator tokens, namely [CLS], w₁, w₂, ..., wₙ, [SEP], h₁, h₂, ..., hₙ, [SEP], c₁, c₂, ..., cₙ, [SEP]. The first segment is the news headline, the second segment is the hot word, and the last segment is the news content. [CLS] token’s hidden representation is regarded as the aggregated sequence representation for "Clickbait" news detection.

**Original Input Format:**

CLS [News Headline] SEP [News Content]
4.3.3 Results of Bert

As shown in the Table 2, the hot words improves all BERT models’ performance of detection. "BERT-base-chinese-hotwords" refers the implemented "BERT-base-chinese" model with hotwords integration.

Table 4: Comparison Results of Detection

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-base-chinese[16]</td>
<td>0.8163</td>
<td>0.8214</td>
<td>0.7555</td>
<td>0.7871</td>
</tr>
<tr>
<td>BERT-base-chinese-hotwords</td>
<td>0.8519</td>
<td>0.8027</td>
<td>0.8613</td>
<td>0.8310</td>
</tr>
<tr>
<td>BERT-wwm[17]</td>
<td>0.8380</td>
<td>0.8314</td>
<td>0.7737</td>
<td>0.8015</td>
</tr>
<tr>
<td>BERT-wwm-hotwords</td>
<td>0.8565</td>
<td>0.8221</td>
<td>0.8431</td>
<td>0.8324</td>
</tr>
<tr>
<td>BERT-wwm-ext[17]</td>
<td>0.8210</td>
<td>0.8161</td>
<td>0.7774</td>
<td>0.7963</td>
</tr>
<tr>
<td>BERT-wwm-ext-hotwords</td>
<td><strong>0.8627</strong></td>
<td><strong>0.9039</strong></td>
<td>0.7555</td>
<td>0.8231</td>
</tr>
</tbody>
</table>

5 Rephrasing

After the detection of "Clickbait" news headline, one may want to rephrasing a "Non-Clickbait" One. It is simailar to the text generation task or summary task which GPT as a decoder is good at. In this section, we proposed two methods:

1) Decoder Only: GPT-2
2) Encoder-Decoder: BERT + GPT-2

5.1 Decoder Only: GPT-2

BERT was primarily developed for encoding text representations for NLU tasks (encoder-only architecture), whereas GPT-2, as a decoder-only architecture for language modeling. In a recent review, the retrospect of GPT is shown: "Equipped by the Transformer decoder as the backbone, GPT applies a generative pre-training and a discriminative fine-tuning. Theoretically, compared to precedents of PTMs, GPT is the first model that combines the modern Transformer architecture and the self-supervised pre-training objective. Empirically, GPT achieves significant success on almost all NLP tasks, including natural language inference, question answering, commonsense reasoning, semantic similarity and classification. Given large-scale corpora without labels, GPT optimizes a standard autoregressive language modeling, that is, maximizing the conditional probabilities of all the words given their corresponding previous words as contexts. In the pre-training phase of GPT, the conditional probability of each word is modeled by Transformer. As shown in Figure 4, for each word, GPT computes its probability distributions by applying multi-head self-attention operations over its previous words followed by position-wise feed-forward layers.”[18]

In 2019, GPT-2 was published by OpenAI as an upgraded version of GPT which was published before. It is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text.[19] GPT has good performances in many NLP tasks, including Text Generation and Summary. In this paper, we pick GPT-2 as our decoder-only model comparing to encoder-decoder model which would be introduced later.

5.2 Encoder+Decoder: Bert+GPT-2

It has been argued that the pre-training objective used by BERT is not well suited for tasks that require decoding texts, e.g., conditional text generation in machine translation and summarization. The current predominant approach to tackle sequence-to-sequence tasks are transformer-based encoder-decoder models often also called seq2seq transformer
models. Encoder-decoder models were introduced in 2017 [13] and since then have been shown to perform better on sequence-to-sequence tasks than stand-alone language models (i.e. decoder-only models). In essence, an encoder-decoder model is the combination of a stand-alone encoder, such as BERT, and a stand-alone decoder model, such as GPT2. [21]

As seen in figure 5, the encoder maps the input sequence \( X_1...X_n \) to a contextualized encoded sequence \( X_1...X_n \) in the exact same way BERT does. The decoder then maps the contextualized encoded sequence \( X_1...X_n \) and a target sequence \( Y_0...Y_{m-1} \) to the logit vectors \( L_1...L_m \). Analogous to GPT2, the logits are then used to define the distribution of the target sequence \( Y_1...Y_m \) conditioned on the input sequence \( X_1...X_n \) by means of a softmax operation.

Here in this paper, we used BERT as our Encoder and GPT-2 as our Decoder in the Encoder-Decoder Model.

5.3 results

5.3.1 proportion of "Clickbait" headlines

We picked 25,000 unlabelled data (all in Chinese) as our main training set for the generation models, and 1,000 as the main testing set. Both GPT-2 and BERT+GPT-2 share the same training set and testing set. For the comparison, all the hyperparameters were shared. We evaluate our models by chinese-bert-wwm-ext every 80 steps to examine the proportion of "Clickbait" headlines. The Precision, Recall and F-Score are shown in Figure 8. As shown in figure 6, BERT+GPT-2 performances better than GPT-2 in the proportion of "Clickbait" headlines, reaching 0.0098 after 32 epochs, while the GPT-2 reaching 0.0332.
5.3.2 Score

To evaluate a summary task, Rouge[22], or Recall-Oriented Understudy for Gisting Evaluation, has been serving as an important metric. We used Rouge-L to evaluate the generation model.

A sequence \( Z = [z_1, z_2, \ldots, z_n] \) is a subsequence of another sequence \( X = [x_1, x_2, \ldots, x_m] \), if there exists a strict increasing sequence \([i_1, i_2, \ldots, i_k]\) of indices of \( X \) such that for all \( j=1,2,\ldots,k \), we have \( x_{i_j} = z_j \) [23]. Given two sequences \( X \) and \( Y \), the longest common subsequence (LCS) of \( X \) and \( Y \) is a common subsequence with maximum length. LCS has been used in identifying cognate candidates during construction of N-best translation lexicon from parallel text. Melamed [24] used the ratio (LCSR) between the length of the LCS of two words and the length of the longer word of the two words to measure the cognateness between them. He used LCS as an approximate string matching algorithm. Saggion et al. [25] used normalized pairwise LCS to compare similarity between two texts in automatic summarization evaluation. [23] Let \( P \) be the proportion of "Clickbait" headlines generated, and Let \( R \) be the Rouge-L, then we can simply define the score (\( S \)) of the model as:

\[
S = (1 - P) + R
\]

By our definition, the higher the \( S \) be, the better the model would be. As shown in Figure 7, the score of GPT-2 is higher than BERT+GPT-2. After 32 epochs, GPT-2 reaches 1.7019 while BERT+GPT-2 reaches 1.4426.
5.3.3 Example

On may want to see an detailed example of our model in rephrasing a "Clickbait" headline. Here is one [26] rephrased by our Encoder-Decoder (Bert+GPT-2) model.

**Original "Clickbait" Headline:**
The Naning small carriage stopped in the middle of the road, and the driver unexpectedly did this in the car...
(一辆小车停在南宁大马路中间，司机竟在车内做这事......)

**Rephrased Headline:**
Nanning: Man fell asleep while driving while drunk
(南宁: 男子醉酒驾车睡着)

**Contents:**
Drank... At about 2:30 pm on June 7, the command center of the Tenth Brigade of Nanning Traffic Police received a report from the masses: A black Volkswagen was parked at the intersection of Jinyang Road, Xintun, Nanning Economic Development Zone, motionless.
I don’t know if the car was broken. However, the driver is still unwell, causing traffic congestion on this section of the road, which poses a safety hazard. Confirmed by the traffic police at the scene, there is a man in the car lying in the driver’s seat. After several minutes of knocking on the window and shouting, the other party did not respond. At first, worried about the man’s illness, the traffic police reported 120. At the same time, he invited the master of the car repair shop and opened the car door. Unexpectedly, when the door of the main driver’s seat was opened, a strong smell of alcohol came out... The police awakened the man, but he was extremely uncooperative and had an arrogant attitude. In the end, the police still called him according to the procedure. The gas alcohol test showed that the blood alcohol content in his body was as high as 123mg/100ml, and he was suspected of drunk driving. The police then took the man to the hospital for further blood tests. It was learned that the man, Zhou, was drinking at a nearby restaurant that day. He was drunk and unconscious when he drove away, so he stopped to sleep, but he parked his car in the middle of the main road.

5.3.4 How the amount of data impact

As seen in Figure 9, we did an experiment using different amount of data as the training set, respectively 1,843 and 25,000. We could see that the amount of training data, or equivalently computational resource since they are unlabelled, is critical.

Figure 9: The Comparsion of Bert+GPT-2 Score different amount of training data

6 Conclusion

In the detection section, we propose two methods, LSTM with hot words knowledge and BERT with hot words knowledge. We found that the hot words knowledge improves both the LSTM and BERT in detection. In the rephrasing section, we developed the "Clickbait" news headline rewriting task and fine-tuned the pre-training model for it. We consider the architecture of the transformer-based pre-training model and propose two methods as Decoder-Only (GPT-2) Model and Encoder+Decoder(Bert+GPT-2) Model. The Encoder+Decoder Model performs better in generating non-clickbait while the Decoder-Only Model performs better in summarizing.
Acknowledgments

The computing work was conducted on Heng Yuan Zhi Xiang Yun GPU Share Platform. We used two GeForce RTX 3090 for computing. Some of the training codes and ideas are from Zhangyue Yin, East Normal University. Thank him.

References


[26] https://new.qq.com/rain/a/20210608A0EF1G00