

A Survey on the Development of Text Summarization

1st Congyu Zhong
71122204

2nd Haoxuan Zang
57122402

3rd Zhihan Zhang
57122405

4th Yifei Zhao
57122410

Abstract—The goal of text summary task is to obtain the compressed text containing the main information of the original text, and its summary generation methods are mainly divided into extraction and generation. Extraction is to extract the key sentences in the original text and splicing them together to form an abstract, while generative is to generalize the original text by generating sentences according to the important content in the original text, which is closer to the form of manual summary written by human beings.

Index Terms—extractive, abstractive, multimodal

INTRODUCTION

In recent years, with the dramatic growth of the Internet, there has been a explosion in the amount of text data from a variety of sources. people are overwhelmed by the tremendous amount of online information and documents. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be use. Therefore, the technology of text summarization emerged as a result. Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning.

I. EXTRACTABLE TEXT SUMMARY

Extractive summarization techniques produce summaries by choosing a subset of the sentences in the original text. These summaries contain the most important sentences of the input. Input can be a single document or multiple documents.

A. Unsupervised Summarization

The traditional extractive summarization methods use graph or clustering methods to achieve unsupervised summarization, most of which are based on statistical level, that is, maximizing the representation ability of the abstract on the original text. Mainly including Lead-3, TextRank, and clustering.

1) *Lead-3*: Lead-3 is a rule-based approach that assumes that the author of the article will indicate the topic at the beginning of the article, so the first three sentences of the article are used as the abstract of the article.

The advantage of this model is its simplicity and ease of implementation. However, its disadvantage is that it cannot capture the contextual and semantic information of the text, which may lead to inaccurate and incomplete summaries generated.

2) *TextRank*: The idea of TextRank is inspired by PageRank, which divides an article into sentences, treats each sentence as a node, constructs a node connection graph, iteratively updates the node values using the weights on the edges, and selects a summary based on the node values. The application of graph based ranking algorithms to natural language texts consists of the following main steps:

- Step 1: Identify text units that best define the task at hand, and add them as vertices in the graph.
- Step 2: Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.
- Step 3: Iterate the graph-based ranking algorithm until convergence.
- Step 4: Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

Edge weights were taken into account when computing the score associated with a vertex, and the following formula was used to integrate vertex weights. An important aspect of Tex-

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

tRank is that it does not require deep linguistic knowledge, nor domain or language specific annotated corpora, which makes it highly portable to other domains, genres, or languages.

3) *clustering*: By the figure1, the clustering method is to use the entire article as the clustering center, calculate the similarity between each sentence and the clustering center, and then sort the calculated similarity to obtain a summary.

Generally clustering problems are determined by four basic components: (1) the (physical) representation of the given data set; (2) the distance/dissimilarity measures between data points; (3) the criterion/objective function which the clustering solutions should aim to optimize; and, (4) the optimization procedure. In this article author developed a distinct differential evolution algorithm to optimize the objective functions.

Table 6
Comparison of evaluation metrics values for NGD-based measure and Euclidean distance (DUC02 dataset).

Methods	Dissimilarity measure	Average ROUGE-1	Average ROUGE-2	Average F ₁ -measure
F_1	NGD-based	0.45658 (+4.65%)	0.11364 (+6.72%)	0.46931 (+3.39%)
	Euclidean	0.43628	0.10648	0.45394
F_2	NGD-based	0.44289 (+5.03%)	0.11065 (+7.46%)	0.46097 (+4.00%)
	Euclidean	0.42167	0.10297	0.44324
F	NGD-based	0.46694 (+3.75%)	0.12368 (+5.48%)	0.47947 (+2.58%)
	Euclidean	0.45007	0.11725	0.46741

Fig. 1. Results of experiment have showed that proposed by us NGD-based dissimilarity measure outperforms the Euclidean distance

B. Supervised Summarization

Traditional methods are simple and efficient, but their generalization ability is insufficient. With the development of deep learning technology, research on extraction methods has gradually shifted towards supervised direction.

1) *SummaRuNNer Model*: In supervised methods, text summarization tasks are regarded as binary classification tasks, and neural networks are used to learn the corresponding relationships between sentences and their labels. Nallapati et al. proposed the SummaRuNNer model, which uses a hierarchical neural network to extract sentence features, capture the hierarchical relationships between words, sentences, and documents, and then binary the resulting sentence representation to determine whether the sentence is a summary. The model diagram is as follows:

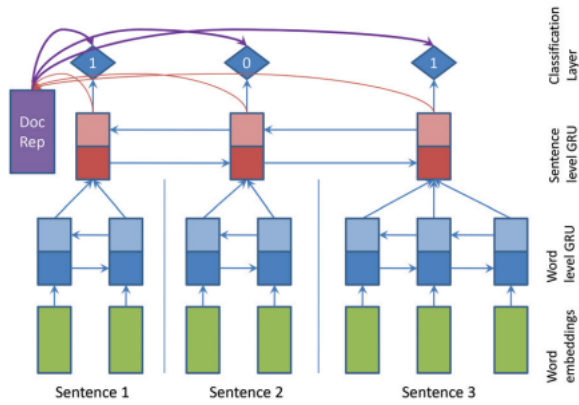


Fig. 2. SummaRunner

SummaRuNNer: A two-layer RNN based sequence classifier: the bottom layer operates at word level within each sentence, while the top layer runs over sentences. Double-pointed arrows indicate a bi-directional RNN. The top layer with 1's and 0's is the sigmoid activation based classification layer that decides whether or not each sentence belongs to the summary. The decision at each sentence depends on the

content richness of the sentence, its salience with respect to the document, its novelty with respect to the accumulated summary representation and other positional features.

2) *Scoring Method(NEUSUM)*: Zhou et al. proposed a new scoring method that takes into account the interrelationships between sentences by recording past sentence extraction and using sentence revenue as the scoring method.

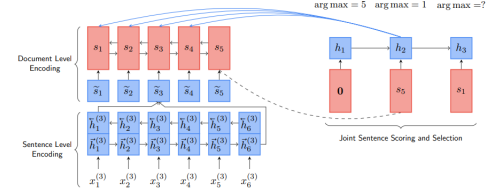


Fig. 3. Scoring Method(NEUSUM)

The model extracts S5 and S1 at the first two steps. At the first step, we feed the model a zero vector 0 to represent empty partial output summary. At the second and third steps, the representations of previously selected sentences S5 and S1, i.e., s5 and s1, are fed into the extractor RNN. At the second step, the model only scores the first 4 sentences since the 5th one is already included in the partial output summary.

This model combines sentence scoring and selection into one phase. Every time it selects a sentence, it scores the sentences according to the partial output summary and current extraction state. ROUGE evaluation results show that the proposed joint sentence scoring and selection approach significantly outperforms previous separated methods.

3) *BertSum*: Liu et al. applied the pre trained language model to the field of abstract extraction for the first time and proposed BertSum. Based on the Bert model, this model adds [cls] markers in front of each sentence to obtain the features of each sentence, and obtains the abstract through the abstract judgment layer.

To use BERT for abstract extraction, it is necessary to output the representation of each sentence. However, due to BERT being trained as a masking language model, the output vector is based on tags rather than sentences. Meanwhile, although BERT has segmentation embeddings for indicating different sentences, it only has two labels (sentence A or sentence B), rather than extracting multiple sentences from the abstract. Therefore, it is necessary to modify the input sequence and embedding of BERT in order to extract the abstract.

As illustrated in Figure, we insert a [CLS] token before each sentence and a [SEP] token after each sentence. In vanilla BERT, The [CLS] is used as a symbol to aggregate features from one sentence or a pair of sentences. We modify the model by using multiple [CLS] symbols to get features for sentences ascending the symbol. Interval Segment Embeddings We use interval segment embeddings to distinguish multiple sentences within a document. For senti we will assign a segment embedding EA or EB conditioned on i is odd or even. For example, for [sent1, sent2, sent3, sent4, sent5] we

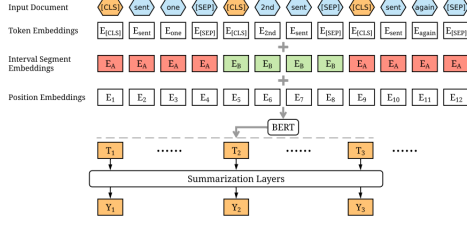


Fig. 4. Encoding Multiple Sentences

will assign [EA, EB, EA, EB, EA]. The vector T_i which is the vector of the i -th[CLS] symbol from the top BERT layer will be used as the representation for senti. The experiment proved that the BERTSUM with inter-sentence Transformer layers can achieve the best performance.

II. ABSTRACTIVE

A. Seq2Seq model

The abstractive is mainly implemented based on sequence to sequence [9] (Seq2Seq), which mainly uses the LSTM structure. Firstly, a multi-layer LSTM is used to map the input sequence to a fixed-dimensional vector (encoder), and then another multi-layer LSTM is used to extract the output sequence from the vector (decoder).

In fact, RNN has been used to solve the sequential learning problem before, but because of the long distance dependence problem, RNN is more difficult to train and LSTM structure can learn long-term dependence, so seq2seq uses LSTM structure.

The goal of LSTM is to estimate conditional probabilities $P(y_1, y_2, \dots, y_{T'} | x_1, x_2, \dots, x_T)$, of which (x_1, \dots, x_T) is the input sequence, $(y_1, \dots, y_{T'})$ is the corresponding output sequence, where the input sequence length T and output sequence length T' are not necessarily equal. LSTM first obtains a fixed-dimensional vector representation v through an input sequence, and then the vector v is used as the initial hidden state of LSTM-LM(language model) to calculate the probability of the output sequence $(y_1, \dots, y_{T'})$. The conditional probability formula is as follows:

$$P(y_1, \dots, y_{T'}) = \prod_{t=1}^{T'} (P(y_t | v, y_1, \dots, y_{t-1}))$$

Where the distribution of $P(y_t | v, y_1, \dots, y_{t-1})$ is expressed as the probability of all words in the word list at that position, and the probability value is calculated by softmax. In order for the model to define a distribution over all possible sequence lengths, that is, the sequence probabilities of all lengths add up to 1, use ϵ EOS $_{\epsilon}$ Tag as sentence end tag. The following diagram clearly illustrates the model structure described above:

In a word, Seq2Seq model contains two core components, encoder and decoder. The encoder is responsible for extracting semantic information of the original text, and the decoder is

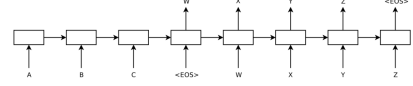


Fig. 5. LSTM model

responsible for obtaining important information in the semantic information extracted by the encoder. And it is processed and edited to generate a text summary.

B. Bahdanau attention

Bahdanau et al. proposed to apply the attention mechanism to the original Seq2Seq to solve the problem of Seq2Seq's poor ability in processing long sequences, and the decoder used the attention mechanism to dynamically extract encoded information.

The emergence of Bahdanau Attention had a significant impact on the development of seq2seq, giving it a second life. Whenever we use a new operator/framework, we think about what it does, for example by comparing it to other methods. So, what's the difference between Attention and the commonly used Full-Connection, RNNs, and CNNs? What difference can this new approach make? With these questions in mind, let's first look at what the paper says. "Intuitively, this implements a mechanism of attention in the decoder. The decoder decides parts of the source sentence to pay attention to. By letting the **decoder have an attention mechanism**, we relieve the encoder from the burden of having to **encode all information** in the source sentence into a fixedlength vector. With this new approach the information can be spread throughout the sequence of annotations, which can **be selectively retrieved by the decoder** accordingly." Summarize the above few sentences. In general, in order for the decoding part to have the function of attention mechanism. Generally speaking, it is to let the decoding part can selectively use the information of the encoding part.

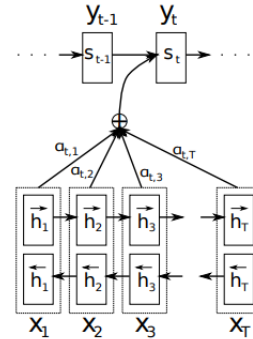


Fig. 6. Bahdanau Attention structure chart

As can be seen from the figure above, the output of each moment in Decoder is determined by several variables together, including the hidden state vector (h_1, \dots, h_n) of each moment in Encoder, the output y_{t-1} of the previous moment, and the hidden state vector s_t of the current moment

in Decoder. So, $p(y_t)=g(y_{t-1},s_t,attention_t(h_1,...,h_n))$. If we revisit the output of time t in the traditional RNN Encoder-Decoder $p(y_t)=g(y_{t-1},s_t,C)$. As you can see, instead of using a fixed semantic coding vector C , we use a dynamic semantic coding vector C' , which is computed by the hidden state vector in the Encoder at every moment. So, $C'_t=attention_t(h_1,...,h_n)$.

From this, the Seq2Seq attention model was born, which will serve as the baseline model for many models in the future.

C. Improvements in seq2seq attention models

TABLE I
IMPROVEMENTS IN SEQ2SEQ ATTENTION MODELS

<i>problems</i>	<i>methods</i>
OOV/UNK	copy
repetition	coverage
repetition	temporal attention
repetition	frequency control

1) *Copy mechanism*: To avoid the problem of Out of Vocabulary that would occur in the decoding stage, Gu et al. [11] and Zeng et al. [12] used a replication mechanism to copy words from input to output. Gu's model is based on a mixture of two probabilistic modes, the generation mode and the copy mode, the model can select the appropriate clause and generate some OOV words. Zeng presents two problems in the Attention-based seq2seq model, which is popular in generative abstracts. The second problem is the problem of UNK (words that are not in the scope of the vocabulary). Most current approaches to solving UNK are to minimize this problem by increasing the vocabulary, but this takes up a lot of storage space and decoding time. To solve these two problems, this paper proposes two improvements to the model: 2, the copy mechanism is used to deal with OOV problems, and a very small thesaurus can be used to improve the efficiency of decoding.

2) *Temporal attention*: Nallapati et al. [13] proposed to solve the problem of temporal attention generation, and its main principle was to reduce the attention score of generated words and improve the attention score of ungenerated words. "Upon visual inspection of the system output, we noticed that on this dataset, both these models produced summaries that contain repetitive phrases or even repetitive sentences at times. Since the summaries in this dataset involve multiple sentences, it is likely that the decoder 'forgets' what part of the document was used in producing earlier highlights. To overcome this problem, we used the Temporal Attention model of Sankaran et al. (2016) that keeps track of past attentional weights of the decoder and explicitly discourages it from attending to the same parts of the document in future time steps."

3) *Coverage*: See et al. [14] introduced the Coverage mechanism, which considers the attention weight of the time step before each step of decoding, and avoids the consideration of the part with high weight in combination with Coverage loss. This mechanism can effectively alleviate the problem of

generating duplication. They add a coverage vector, whose value is the sum of the decoder's attention distribution of past time steps.

$$c^t = \sum_{t'=0}^{t-1} a^{t'}$$

This vector represents the distribution of the extent to which past time steps cover the source word. The purpose is to calculate the attention of the current time step to the source text, taking into account the situation that the previous time step has paid attention to the source text (for example, the total attention of the previous time step to a certain word in the source text has been very high, then we hope that the current step will pay less attention to this word, to avoid repetition, of course, This is determined by the model test, hopefully the model will learn this information), and the ultimate goal is to reduce duplication.

4) *word frequency limitation*: Suzuki et al. [15] used word frequency information to limit repetition generation. The model predicted the maximum frequency of each word in the abstract, so that the decoded word would not exceed the estimated frequency to solve the problem of repetition generation. The basic idea of our method is to jointly estimate the upper-bound frequency of each target vocabulary that can occur in a summary during the encoding process and exploit the estimation to control the output words in each decoding step. They refer to our additional component as a word-frequency estimation (WFE) sub-model. The WFE sub-model explicitly manages how many times each word has been generated so far and might be generated in the future during the decoding process. Thus, we expect to decisively prohibit excessive generation.

D. Application of pre-trained models

With the emergence of a large number of pre-trained models, the structure of the generative summary model has been changed. Most of the early pre-trained model structures were single-encoder structures, so most of the generative summary models at this time used the pre-trained model to initialize the encoder, and then fine-tuned with the randomly initialized decoder. Liu et al. [16] added Transformer decoder to BertSum's encoder to enable BertSum to perform generative text summarization tasks. Tan Jinyuan et al. [17] of Dalian Institute of Technology combined traditional features and thematic features with BERT features to describe sentences in Chinese news texts in a more granular manner and improve the contextual semantic representation performance of sentences in the text. Zhang Shizhong [18] of Wuhan Institute of Posts and Telecommunications combined BERT and PGN, and integrated keywords into the attention mechanism through a keyword search algorithm, so that the model paid more attention to the gist information of the text in the process of generating the abstract. Later, some pre-training models of Seq2Seq structure appeared, such as MASS[19], T5[20], BART[21], etc. Most of the pre-training tasks of these models were in the form of natural language generation, which made

these models more suitable for natural language generation tasks. By using these pre-training models, the performance of generative summary models was greatly improved.

III. MUTI-MOUDAL

In recent years, multimodal summarization has attracted extensive attention in the academic community. Compared with pure text summarization, multimodal summarization requires inputting multiple modal information, usually including text, speech, pictures, video and other information, and outputting a core summary after considering multiple modal information. On top of the original text summarization technology, combining picture features is the research hotspot of multimodal summarization, which realizes the alignment of picture semantics and text semantics by combining with the picture feature extraction technology, which opens up a new technical research in the field of summary generation and extends the research focus from the previous single modality to multimodal data. According to the current research [1], multimodal summarization not only relies on textual information, but information in the visual modality can also help the model to improve the quality of the generated summaries.



<p>Source sentence: a house explosion rocked a neighborhood in eastern maryland , killing a gas utility worker and injuring four residents and ## firefighters .</p> <p>Reference summary: <i>house explosion</i> in maryland kills gas worker injures ##</p> <p>Text-only model: gas explosion in us kills gas explosion</p> <p>Multi-modal model: <i>house explosion</i> rocks maryland killing ##</p>	
<p>Source sentence: the flood death toll in southern malaysia has risen to ## , an official said thursday .</p> <p>Reference summary: <i>flood</i> death toll rises to ## in southern malaysia</p> <p>Text-only model: southern malaysia death toll rises to ##</p> <p>Multi-modal model: death toll from heavy <i>floods</i> rises to ##</p>	

Figure 1: Example summaries generated by different models. Multi-modal model successfully predicts the main event objects (in green and *italic*).

A. VLBERT

Li et al [1] proposed to employ layered attention for the fusion of picture features and text features, where the bottom layer focuses on the picture and text separately within each, and balances the two modalities at the top layer, and proposes a filtering mechanism for the picture noise. Su et al [2] proposed a two-stream multimodal pre-training model, VILBERT, which is a dual-stream architecture that models each modality separately. They are then fused together through a small set of attention-based interactions. VLBERT is an upgraded version of the BERT model, and similar to the BERT model, it also has a multi-layer bidirectional Transformer encoder. Unlike the BERT which only deals with words in a sentence, VL-BERT accepts both visual and linguistic elements as input, which are defined in the region of interest (RoI) in the image and the subwords in the input sentence, respectively.

VL-BERT is designed as a generalized feature representation for a variety of visual-verbal tasks. Fine-tuning VL-BERT

to suit different downstream tasks is relatively simple. We simply provide inputs and outputs in the appropriate formats and then fine-tune all network parameters end-to-end. For inputs, the typical formats ;Caption, Image; and ;Question, Answer, Image; cover most visual-linguistic tasks. VL-BERT also supports more sentences and more images as long as appropriate segment embeddings are introduced to recognize different input sources. On the output side, typically, the final output features of [CLS] elements are used for sentence-image relation level prediction. The final output features of words or RoIs are used for word-level or RoI-level prediction.

The input elements contain visual elements (image region RoI), linguistic elements (words), and special elements ([CLS], [SEP], etc.), and each input element representation is stitched together by four parts: token embedding, visual feature embedding, segment embedding, and position embedding. The visual feature embedding includes the splicing of visual representational features and geometric position embedding. Through the Transformer layer, the input elements can be freely interacted and associated to realize visual language interoperability. Finally, sentence-level prediction can be performed based on [CLS] and other output features, and fine-grained prediction can be performed based on word or RoI features.

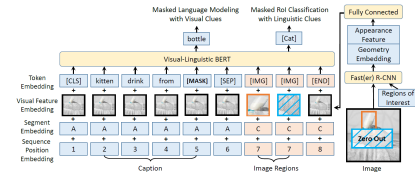


Fig. 7. VL-bert

B. HCSCL

Typically, visual images and textual articles have heterogeneous structures. Directly mapping visual and textual inputs into global vectors does not effectively learn important information about the two modalities from each other, and even noisy information is added to degrade the performance of summarization. Meanwhile, the correlation between visual content and textual articles presents a unique feature, where low-level objects in the image constitute high-level semantics called scenes through their interactions. In another data space, words are also the basic textual information in an article, while combinations of words, called sentences, present more abstract semantic information. In addition to intra-modal associations, semantic objects in images and articles are associated at different levels. For this reason, a hierarchical learning model, HCSCL, is proposed to learn intra- and inter-modal correlations in multimodal data. HCSCL[3] consists of three modules: a modal feature encoder to encode each modality, a hierarchical semantic correlation fusion module to learn intra- and inter-modal correlations, and a multimodal output summary generator to generate multimodal summaries using hierarchical

correlations. When we input something HCSCL will separate processing of images and text. First, article sentences are encoded using LSTM and object features in the images are extracted using Faster-RCNN. Then a hierarchical semantic association fusion module is passed in to learn relevance at two levels: word-object fusion and sentence-scene fusion. Word-object fusion uses an attention-based cross-modal encoder (CME) to learn inter-modal relations (b in Fig. b). The CME consists of three parts: a cross-attention layer, a self-attention layer, and a feedforward layer. Residual connectivity and LayerNorm are also incorporated in each of these sublayers. The three steps are repeated several times to obtain fused word representations and fused object representations. Sentence-scene fusion consists of two aspects (c in Fig. c), on the one hand, the word representation is passed into the LSTM model to get a representation of the whole sentence. On the other hand, a portion of an object in the image is associated to form a scene to represent a more abstract concept or activity. First, object bounding boxes are extracted based on the image encoder, and an IOU score is computed for every two objects. Next, a relational graph is constructed with an adjacency matrix A. If the IOU score exceeds a threshold, = 1, otherwise = 0. The feature score of object i with respect to object j is then computed according to the following formula:

$$s_{ij}^{feature} = w_1^T \sigma(w_2 \cdot \text{Concat}(v_i^1, v_j^1))$$

The weights of the directed edges are then obtained by combining the IOU scores and the feature scores:

$$s_{ij}^{edge} = \exp(s_{ij}^{IOU} \cdot s_{ij}^{feature}) / \sum_{t \in N(i)} \exp(s_{it}^{IOU} \cdot s_{it}^{feature})$$

After that, the characterization of a series of sub-scene graphs is obtained by combining the adjacency matrix A:

$$\begin{aligned} \tilde{v}_i^1 &= \sigma(v_i^1 + \sum_{j \in N(i)} s_{ij}^{edge} A_{ij} w_3 v_j^1) \\ \{v_p^2\} &= \text{readout}(\tilde{v}_1^1, \tilde{v}_2^1, \dots, \tilde{v}_q^1) \end{aligned}$$

Finally, sentence-scene fusion feature sums are computed by CME.

The multimodal output summarizer generates a text summary accompanied by a most relevant image. For text summarization generation, a hierarchical attention mechanism is used. At time slice t-1, the fused sentence features are passed in to get the hidden state, and then the weights of i-th sentences and the weights of j-th words in i-th sentences are computed according to the following formula, respectively.

$$\beta_i^{sent} = \text{softmax}(\text{score}(h_i^3, h'_{t-1}))$$

Then, the context and predicted words at time t are computed according to the following formula.

$$\beta_{i,j}^{word} = \text{softmax}_{i,j}(\beta_i^{sent} \cdot \text{score}(h_{i,j}^1, h'_{t-1}))$$

$$c_t = \sum_{i=1}^N \sum_{j=1}^M \beta_{i,j}^{word} h_{i,j}^1$$

$$p(y_t | y_{1:t-1}) = \text{softmax}(V^T F F(h'_t, c_t))$$

C. CLIP

[4] Compared with other models, the input of this model is an data pair, so the advantage of this is that it can be compared to learn and predict N*N pairs of graphic data, converting the image classification task into a graphic matching task. As the image shows, there are the following advantages

a) Dual stream, 2 encoders process text and image data respectively, text encoder uses Transformer, image encoder uses 2 models, ResNet and Vision Transformer(ViT);

b) Calculate the cosine similarity between the 2 modalities, so that the N matching image-text pairs have the maximum similarity and the mismatched image-text pairs have the minimum similarity;

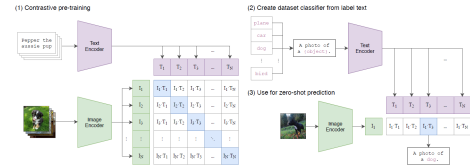


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Fig. 8. CLIP

In order to obtain better pre-training, the model uses natural language as a supervised signal, i.e., a method of learning visual representations using textual descriptions as training signals. The advantage of using natural language supervision over other supervision is that natural language supervision utilizes a large amount of easily accessible web text, without the need for manual annotation. It can provide a large amount of found your data and express a wide range of concepts, not limited to the terminology used to categorize the image, but also Bain with downstream migration work.

The use of image-text matching as a pre-training target is proposed for the first time. Initial experiments using images as conditions for generating language models with textual descriptions were computationally inefficient. Then we tried to use image-text matching as the prediction target, which is also 3-4 times less efficient than comparative learning. These two approaches have one thing in common: they try to predict the exact content of the text, i.e., the caption, for each word. But this is difficult because there are many ways to express the same meaning. The idea of comparative learning was eventually borrowed to explore the idea of training a system to solve the potentially simpler agent task of predicting only which text as a whole is paired with which image, rather

than the exact words of that text. Starting from the same bag-of-words encoding baseline, we replaced the predictive objective with a contrastive objective and found a 4x efficiency improvement on ImageNet's zero-shot transfer.

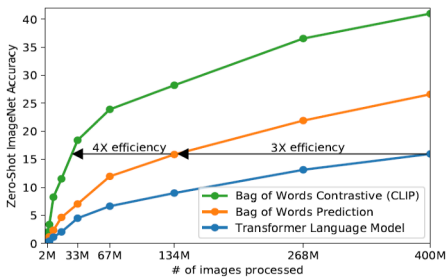


Figure 2. CLIP is much more efficient at zero-shot transfer than our image caption baseline. Although highly expressive,

Fig. 9. CLIP

D. Three models of progress

Three model have its own innovation and progress. VL-BERT: The multimodal domain lacks a powerful pre-trained model like BERT. Previous multimodal models focus more on unidirectional mapping and do not adequately model the interaction between linguistic and visual features. But it simply splices image region features and text features, and does not fully model the association of different semantic levels.

HCSCL: proposes a hierarchical cross-modal similarity computational system that can match images and text at different semantic. But its pre-training dataset is still limited, and the semantic relations are mainly confined to specific objects and attributes. CLIP: Utilizing an ultra-large image-text dataset, generalized image-text co-embedding is obtained through comparative learning, which serves as an excellent pre-training model for multimodal downstream tasks.

SUMMARY OF RESEARCH STATUS

From the current research results at home and abroad, the development of the field of text summary has greatly benefited from the rapid development of deep learning. There are mainly two technical routes of text summary: extraction and generation. The extraction abstract has a low error rate in grammar and syntax, but there are some problems, such as: the extracted sentences are redundant, the coherence of the abstract is poor and not flexible. Instead of abstracting a certain part of the original text directly, generative abstracts try to understand the meaning of the original text, organize the language to generate the corresponding abstract, which is more flexible than abstracts. The current research focus is generative text summary.

With the rapid growth of multimedia data on the Internet, multimodal abstracts have gradually caused extensive research. Existing research [2] has proved that, compared with plain text abstracts, multimodal abstracts can improve the quality of generated abstracts by using image features. In addition, multimodal output can significantly improve users' satisfaction

with summary information [3]. At present, the research on multi-modal abstract mainly focuses on how to interact multi-modal data, which can be generally divided into single-stream model and dual-stream model. Single-stream model refers to the text input and image input are splicing together and then directly input into the encoder, and modal information fusion occurs earlier, while the two-stream model is processed by the encoder of the respective modes. Cross-mode interaction is usually achieved through Cross Attention. The parameter utilization efficiency of single flow in the two modes is high, because the same parameter is used for different modes, while the parameter utilization efficiency of double flow is low because different parameters are required for different modal data and parameters of the cross-attention layer are required. The existing research on multimodal summary also adds some auxiliary tasks to the original summary generation task to enhance the effect of summary generation.

REFERENCES

- [1] MIHALCEA R, TARAU P. TextRank: Bringing Order into Text[J]. 2004.
- [2] R. M. Aliguliyev. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications, 2009.
- [3] NALLAPATI R, ZHAI F, ZHOU B. SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents[J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022.
- [4] ZHOU Q, YANG N, WEI F, et al. Neural Document Summarization by Jointly Learning to Score and Select Sentences[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia. 2019.
- [5] LIU Y. Fine-tune BERT for Extractive Summarization.[J]. Cornell University - arXiv, 2019.
- [6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. 2017. Text Summarization Techniques: A Brief Survey. ArXiv e-prints (2017). arXiv:1707.02268
- [7] Mridha MF, Lima AA, Nur K, Das SC, Hasan M, Kabir MM (2021) A Survey of Automatic Text Summarization: Progress, Process and Challenges. In: IEEE Access, vol 9, pp 156043– 156070
- [8] SUTSKEVER I, VINYALS O, LE Quoc V. Sequence to Sequence Learning with Neural Networks[J]. 2014.
- [9] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Cornell University - arXiv, 2014.
- [10] GU J, LU Z, LI H, et al. Incorporating Copying Mechanism in Sequence-to-Sequence Learning[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany.2016.
- [11] ZENG W, LUO W, FIDLER S, et al. Efficient Summarization with Read-Again and Copy Mechanism[J]. Cornell University - arXiv, 2016.
- [12] NALLAPATI R, ZHOU B, DOS SANTOS C, et al. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond[C/OL]//Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany. 2016.
- [13] [SEE A, LIU Peter J, MANNING Christopher D. Get To The Point: Summarization with Pointer-Generator Networks[J]. arXiv: Computation and Language, 2017.
- [14] SUZUKI J, NAGATA M. Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization[J]. Cornell University - arXiv, 2016.
- [15] LIU Y, LAPATA M. Text Summarization with Pretrained Encoders[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China. 2019.
- [16] Haoran Li;Junnan Zhu;Tianshang Liu.Multi-modal Sentence Summarization with Modality Attention and Image Filtering[A].27th International Joint Conference on Artificial Intelligence (IJCAI)[C],2018

- [17] SU WeijieJ, ZHU X, CAO Y, et al. VL-BERT: Pre-training of Generic Visual-Linguistic Representations[J]. International Conference on Learning Representations (ICLR) [C],2020
- [18] ZHANG L, ZHANG X, PAN J, et al. Hierarchical Cross-Modality Semantic Correlation Learning Model for Multimodal Summarization[J].
- [19] RADFORD A, KIM J, HALLACY C, et al. Learning Transferable Visual Models From Natural Language Supervision[J]. Proceedings of Machine Learning Research