Multi-lingual Wikipedia Summarization and Title Generation On Low Resource Corpus

Wei Liu, Lei Li, Zuying Huang and **Yinan Liu**
Center of Intelligence Science and Technology
School of Computer Science
Beijing University of Posts and Telecommunications
{thinkwee, leili, zoehuang, lyinan}@bupt.edu.cn

Abstract

MultiLing 2019 Headline Generation Task on Wikipedia Corpus raised a critical and practical problem: multilingual task on low resource corpus. In this paper we proposed Quality-Diversity Automatic Summarization(QDAS) model enhanced by sentence2vec and try to apply transfer learning based on large multilingual pre-trained language model for Wikipedia Headline Generation task. We treat it as sequence labeling task and develop two schemes to handle with it. Experimental results have shown that large pre-trained model can effectively utilize learned knowledge to extract certain phrase using low resource supervised data.

1 Introduction

MultiLing 2019 is an accepted RANLP 2019 workshop, focused on the multi-lingual aspect of summarization, but also its value across different settings. It holds three community tasks including: Headline Generation, Financial Narrative Summarization and Summary Evaluation. We have participated in the Wikipedia part of Headline Generation task, which is described as follows: Given Wikipedia articles from 42 language, for each article the title(Wikipedia Entry) and subtitles are masked with title length, as well as the summary. Researchers should reconstruct the title and subtitles of masked articles.

The classic seq2seq architecture for generating headlines is not suitable for this task since the given corpus is not large enough to train a seq2seq model from scratch for each language. Another downside of seq2seq is that it can not handle summarization tasks with large compression ratio, such as taking a whole Wikipedia article as input and a title(usually a phrase) as output. So in this paper we propose a two-steps model for Wikipedia Headline Generation:

- **Reconstruct Summary: Extractive Summarization** Extract some sentences from the whole Wikipedia article to formulate summaries. We reconstruct titles based on the extracted summaries not the whole article.

- **Reconstruct Title: Sequence Labelling** Unlike other corpus, Wikipedia titles are phrases and can often be found in original sentences. So we transform this language generation task into sequence labeling task. For each summary sentence we try to mark some positions as title phrase and choose the best one.

2 Background

2.1 Summarization

Classification for automatic summarization is based on whether a sentence is from the raw document or not. Recently, brand-new proposed researches pay a lot attention to the abstractive summarization: applying structure and semantic methods to generate new sentences(Alfonseca et al., 2003) as summary is common in the early years, while it is now time for the neural network to perform. Seq2seq model(Lopyrev, 2015), as a typical abstractive summarization approach can map one long sentence (article) to another short sentence (summarization). However, an abstractive method is always limited in short papers and requires more advanced technology for natural language processing. As for long papers, for instance, single-document from Wikipedia, an extractive way seems like an easier and more convenient target, and this simple but robust method even gives its best shot when put into practical use.

Proceedings of the Multiling 2019 Workshop, co-located with the RANLP 2019 conference, pages 17–25
Varna, Bulgaria, September 6, 2019.
http://doi.org/10.26615/978-954-452-058-8_004

Thus, in this paper, we would like to adopt extractive summarization due to its increased feasibility.

2.2 Headline Generation

As a special application scenario of abstractive summarization, headline generation gains a lot of attention in recent years. In the HEADS(Colmenares et al., 2015) system researchers formulate the headline generation as a discrete optimization task in a feature-rich space. (Sun et al., 2015) combined extractive and abstractive summarization to detect a key event chain in article and generate titles based on it. (Takase et al., 2016) tried to incorporate structural syntactic and semantic information into a baseline neural attention-based model. There are also works focusing on extending sentence compression to document headline generation (Tan et al., 2017). Most of these works use seq2seq model, which is not suitable for Wikipedia Headline Generation task in MultiLing 2019 due to low resource multilingual training corpus. So we apply Pre-trained model to utilize the semantic knowledge learned in large unsupervised corpus on low-resource supervised task.

2.3 Pre-trained Language Model

Pre-trained Language Model(LM) is one of the most important research advances in Natural Language Processing(NLP) which focus on how to make use of language information in large corpus with unsupervised learning. Word2vec(Mikolov et al., 2013) and Glove(Pennington et al., 2014) have successfully learned semantic information in word embeddings and have been widely used in NLP tasks as inputs for model. Pre-trained language models explore more by learning syntactic and more abstractive features. These language models enrich embeddings information by adding encoders in pre-trained parts, producing context-aware representations when transfer to downstream tasks. Representative works including ULMFiT(Howard and Ruder, 2018), which captures general features of the language in different encoder layers to help text classification; ELMo(Peters et al., 2018), which learns embeddings from Bidirectional LSTM language models; BERT(Devlin et al., 2018), a successful application of training Transformer encoders on large masked corpus and reach eleven state-of-the-art results. After the release of BERT, many super-large-scale Transformer-Based models have been raised including GPT-2(Radford et al., 2019), MASS(Song et al., 2019) and XLNet(Yang et al., 2019).

3 Pipeline Overview

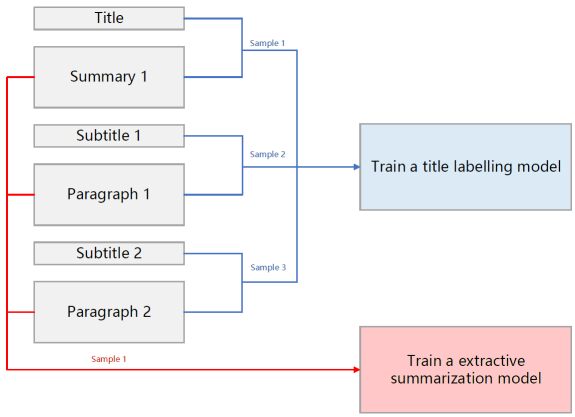

Figure 1: Pipeline overview during training. Red lines refer to samples for training a extractive summarization model and blue for training a title labelling model.

Figure 1 shows the pipeline during training. Given a Wikipedia article, The extractive summarization should extract summaries from paragraphs. The summarization model is unsupervised so actually there is no explicit training sample for summarization but we design features based on some statistics from paragraphs-summary pairs. For headline generation, we aim to provide sequence labelling data for model. In each article the title-summary and all subtitle-paragraph pairs are extracted to formulate training pairs. The process of transforming text pair into tagging sequence is described in section 7.

During test phrase, First we use summarization model to extract summaries from paragraphs and then for each sentence in the extracted summaries, title positions will be tagged out using the title labelling model. For subtitles, no summary is need and they are directly tagged out from corresponding paragraph sentences. There are maybe multiple candidate for each title or subtitle. The test corpus provides gold length so we pick up the candidate which has length closest to gold as final result.

4 Determinantal point processes

4.1 Definition

A discrete, finite point process P on a ground set D is a probability measure over its subsets, where determinant offers a kind of quantitatively analysis on this probability. P is called a determinantal point processDPPs if, when Y is a random subset drawn according to P, so that for every $A \subseteq D$:

$$P(A \subseteq \mathbf{Y}) = \det(K_A) \tag{1}$$

where $\mathbf{Y}$ is a specific instantation for random variable Y. That is to say, $\mathbf{Y}$ contains all the sentences the DPPs sampling method selects from the raw document. Set A provides a metric to measure the probability that two or more correlated sentences are extracted at the same time. The probability restriction is somehow related to a real symmetric matrix K that indexed by the elements of D.

Suppose there are N sentences in total, $D = \{x_1, x_2, ..., x_N\}$, here $K_A \equiv [k_{ij}]_{x_i, x_j \in A}$. Take $A = \{x_i, x_j\}$, then:

$$P(x_i, x_j \in \mathbf{Y}) = \begin{vmatrix} k_{ii} & k_{ij} \\ k_{ji} & k_{jj} \end{vmatrix} \tag{2}$$

$$= k_{ii}k_{jj} - k_{ij}k_{ji} \tag{3}$$

where k_{ij} or k_{ji} can be thought of the of similarity between sentences x_i and x_j, so that highly similar sentences are unlikely to appear together.

Since P is a probability measure, there are some frigid rules to obey when matrix K is constructed, i.e. K itself must be positive semidefinite to guarantee that all principal minors $\det(K_A)$ of K must be nonnegative; or $0 \leqslant K \leqslant I$, which ensures the probability to be in $[0, 1]$.

L-ensemble defines a DPP through another real, symmetric kernel L, also indexed by the elements of D:

$$P_L(\mathbf{Y} = Y) \propto \det(L_Y) \tag{4}$$

$$P_L(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\det(L + I)} \tag{5}$$

To be clear, an L-ensemble is still a DPP, and its marginal kernel K is:

$$K = L(L + I)^{-1} = I - (L + I)^{-1} \tag{6}$$

L-ensemble provides an original method of scale to liberate the strict restriction on determinant, and (5) directly specifies the atomic probabilities for every possible instantiation of Y while K merely gives marginal probability of one certain item to be selected in one particular sampling process.

4.2 Quality vs. Diversity

Interpretability remains a common concern when we put a DPP into practical use. The DPP kernel L can be written as a Gram matrix:

$$L = B^\top B \tag{7}$$

where the columns of B are vectors representing sentences in the set D We now take this fact one step further, write each column B_i as the product of its norm q_i and a vector of normalized ϕ_i, so that the entries of the kernel can now be written as:

$$L_{ij} = q_i \phi_i^\top \phi_j q_j \tag{8}$$

We call q_i as a measure of quality of a sentence x_i, since the norm has a distance interpretation in Euclidean space. $\phi_i^\top \phi_j$ refers to a measure of similarity we assume S between sentences x_i and x_j.

In this way, we first calculate quality and similarity separately and then fuse them in a unified model to construct a kernel L. The determinant of a matrix, which the latter sampling process relies on, also has an intuitive geometric interpretation:

$$P_L(\mathbf{Y} = Y) \propto \det(L_Y) = Vol^2(\{B_i\}_{i \in Y}) \tag{9}$$

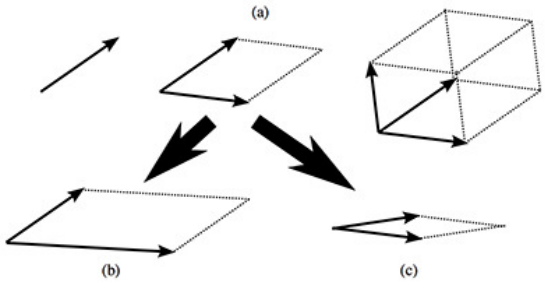

Figure 2: A geometric view of DPPs

Fig. 2.: Geometric view of DPPs (suppose there are two sentences in Y):
(a) The probability of a subset Y is the square of

the volume spanned by B_i and B_j.
(b) As quality increases, the norm stretches, so does the probability of subset containing sentence x_i.
(c) As two sentences x_i and x_j become more similar, the angle decreases, so does the probability containing both of them.

4.3 Sampling Algorithm

Input: q_i, S, D, max_len.
$quality\_vec$= [q_i for i in D]
matrix_L= $quality\_vec * S * quality\_vec^T$
$(\mathbf{v}_n, \lambda_n)$ =eigen_decompose(matrix_L)
$J = \emptyset$
for $n = 1, 2, \ldots, N$ **do**
    $J = J \cup \{n\}$ with prob. $\frac{\lambda_n}{\lambda_n+1}$
    $V = \{\mathbf{v}_n\}_{n \in J}$
End for
$Y = \emptyset$
While $|V| > 0$ **do**
    Select i^{th} sentence from D
        with $Pr(i) = \frac{1}{|V|} \sum_{v \in V} (\mathbf{v}^\top \mathbf{e}_i)^2$
    $Y = Y \cup D[i]$
    $V = V_\perp$, an orthonormal basis for the
        subspace of V orthogonal to e_i
    $|V| - -$
End While
Output: summary Y

An expected sample result based on the determinant of kernel L takes not only quality of items but also the interior cohesion into account. In order to explain the sampling algorithm more precisely, there are some extra principle properties of DPPs worth to be mentioned.

- A DPP with kernel L is a mixture of elementary DPPs
- If Y is drawn according to an elementary DPP with a set of orthonormal vectors $\mathbf{v}_i, i = 1, 2, ..., k, k < N$, then $|Y| = |V|$. Also, let $\lambda_i, i = 1, 2, ..., N$ be the eigenvalues of L, then $|Y|$ is distributed as the number of successes in N Bernoulli trials where trial n succeeds with probability $\frac{\lambda_n}{\lambda_n+1}$.

A DPP is called elementary if every eigenvalue of its marginal kernel is either 0 or 1, so that all principal minors $\det(K_A)$ is either 0 or 1, due to the fact that determinant equals the product of all eigenvalues. The multiplication of any normalized vector and its transpose $\mathbf{v}\mathbf{v}^\top$ happens to be a matrix with such property. Since we have already obtained the kernel L and its corresponding eigenvectors, based on the conversion relationship between kernel L and marginal kernel K according to (6), this theory points out another representation of marginal probability of A in a mixture way.

$$P(A \subseteq \mathbf{Y}) = \det(K_A) = \det(\sum_1^N \frac{\lambda_n}{\lambda_n + 1} W_n) \quad (10)$$

where W_i was spanned by the eigenvector from kernel L of corresponding sentence x_i in A, and λ_i refers to its eigenvalue.

From these two properties above, we notice that a DPP is initially defined through marginal kernel with continuous probability in $[0, 1]$, while a elementary one provides merely two outcomes: to be selected or not. This perhaps inspires the sampling process to choose an elementary DPP with probability equal to its mixture component in the first loop, and the cardinality $|Y|(= |V|)$ is determined meanwhile. To be clear, the mixture way can be regarded as the mathematical expectation from multiple trials, but when it comes to an instantiation, selection with probability is used to simulate the results from Bernoulli distribution.

A sample Y is produced during the second loop phase. Since the new elementary theory is also defined through determinants, using its analogical geometric interpretation by the base * height formula for the volume of a parallelepiped we have:

$$Vol^2(\{B_i\}_{i \in Y}) = \|B_1\| Vol(\{Proj_\perp e_1\}_{i=2}^k) \quad (11)$$

where B_1 denotes the 1^{st} sentence to be selected, e_1 stands for its one-hot representation and $Proj_\perp e_1$ refers to the projection operator onto the subspace orthogonal to e_1. Assume we have already selected the best B_1, and then the V need to be updated to an orthonormal basis for the subspace of the original V perpendicular to e_1 for diversity. Proceeding inductively, the loop goes on. During each iteration, the first vector in V that contributes to the norm of B_1, which makes its quality the best, is eliminated.

5 Reconstruct Summary

Our Quality-Diversity Automatic Summarization (QDAS) framework merely requires general preprocessing like sentence splitting and word segmentation, and then it can be applied in multilingual environment. When it comes to document

representation, first we construct matrix L from holistic perspectives, through $L_{ij} = B_i^\top B_j$ from Sent2Vec directly. Furthermore, we build matrix L from partial perspectives, through $L_{ij} = q_i S_{ij} q_j$ concretely. we extract quality q_i for a sentence, and calculate cosine similarity S_{ij} between every two sentences. Given the matrix L, the sampling method based on DPPs introduced by Kulesza and Taskar (Kulesza et al., 2012) ($O(N^2)$) can automatically choose diverse sentences with high quality. When constructing a semantic space using embedding expressions, quality refers to the length of a vector in the semantic space. Sentences that indicate strong semantic feature are called high quality and preferred for summarization.

6 Reconstruct Title

We use two kinds of BERT(Devlin et al., 2018)(Bidirectional Encoder Representations from Transformers) Based sequence labeling schemes to label the title phrases, which are CRF Model and NMT Model. We developed our code based on sberbank-ai's open source project[1].

6.1 Baseline

Based on the fact that most titles in Wikipedia articles are entries or concepts that often appear in the first sentence, we set up two simple but effective baselines to extract titles:

- **NER** Use Named Entity Recognition to extract named entities in the first sentence of summary. We simply choose the first entity as title.
- **SUB** Based on dependency parse we found the subject of the first sentence in summary and choose it as title.

We use Spacy(Honnibal and Montani, 2017) to perform the NER extraction and dependency parsing.

6.2 BERT

BERT is used to formulate the encoder of sequence labeling model. It is trained for language modeling task on large corpus and can be easily applied to several natural language tasks, using its token embeddings or sentence embeddings.

BERT consists of multiple bidirectional Transformer layers and perform two unsupervised tasks on large corpus:

- Masked LM: standard conditional language models can not been trained in two directions since tokens would indirectly "see itself" in multi-layer bidirectional model. BERT randomly mask some tokens and predict these masked tokens. Furthermore, to prevent the mismatch problem between pre-training and fine-tuning, BERT do not simply mask the token to the symbol $[MASK]$, but replace the chosen token with (1) the $[MASK]$ 80% of the time (2) a random token 10% of the time (3) the unchanged token 10% of the time.
- Next Sentence Prediction: to capture the syntactic and context-aware information of language, BERT adds a sentence-level task: a binarized next sentence prediction task. Adjacent sentence pairs are fed to BERT and only 50% of the time the second sentence is the actual next sentence that follows the first sentence. 50% of the time it is a random sentence from the corpus.

The input representation is sentence pairs that are packed together into a single sequence. The first token of every sequence is a special classification token($[CLS]$) which has final hidden state as the aggregate sequence representation for classification task. Sentence pairs are separated by a special token($[SEP]$). The input embedding for each token contains three parts: Token Embedding, Segment Embedding and Position Embedding.

We chose official pre-trained Multilingual Cased Base version of BERT as encoder for sentences, which has 110M parameters developed on 104 languages. We make sequence labeling data using sentences from gold summaries. The multilingual BERT model can use data from all 42 languages instead of training separate model for each language. The pre-trained BERT can be fixed as a "context-aware embedding look-up table" or fine-tuned together with downstream model. We chose the former way for two reasons:

- The task dataset are the same as pre-trained BERT data source, which is Wikipedia.
- The supervised training set is too small for the whole BERT model to transfer. Fine-tuning will make sharp parameters adjustments which harms the performance of pre-trained model.

[1] https://github.com/sberbank-ai/ner-bert

6.3 BERT Based CRF Model

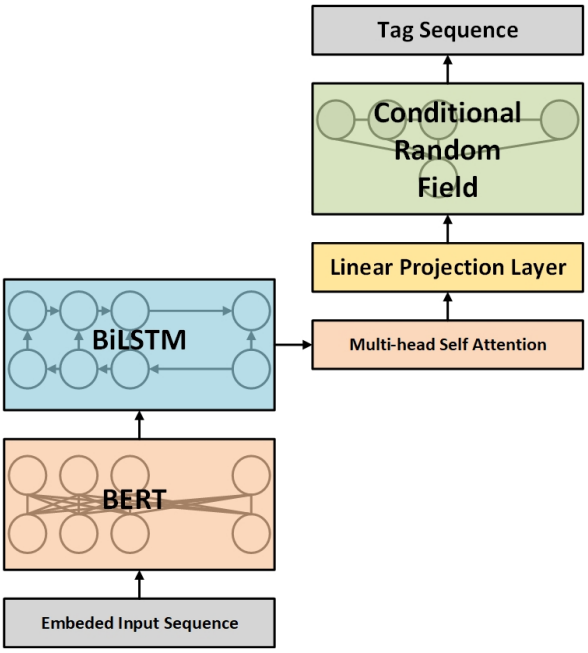

Figure 3: Bert Based CRF Model

The representations for all tokens in a sequence from BERT's last layer are fed to decoder for tagging title phrase. The encoded information are first fed to a bidirectional LSTM layer then a multi-head self attention layer and a liner projection layer to generate tag probabilities. Last a CRF(Conditional Random Field)(Lafferty et al., 2001) layer is added to adjust the tag sequence. The model architecture is shown in Figure 3.

6.4 BERT Based NMT Model

The BERT Based NMT(Neural Machine Translation) Model is almost the same as BERT Based CRF Model except for the decoder. NMT model uses sequence to sequence architecture to generate tag sequence. The encoded information from BERT and bidirectional LSTM is decoded by a unidirectional LSTM decoder. Classic attention mechanism using dot alignment function is applied between encoder and decoder to focus on different parts of encoded information when generate one tag. The decoder also accept decoder input embeddings, which is the same as encoder input embeddings. The model architecture is shown in Figure 4.

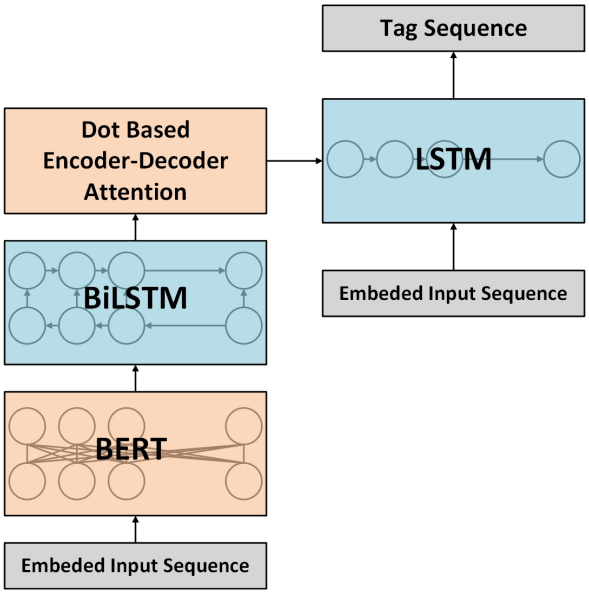

Figure 4: Bert Based NMT Model

7 Experimental Setup

The MultiLing 2019 Headline Generation task provides 9293 Wikipedia articles from 42 languages. Most of the languages have only 30 articles for training. Several languages have more articles. The details are shown in table 1.

Language	# Articles
EN	3793
DE	2112
ES	1024
HE	639
IT	277
ZH	178
AR	175
JA	51

Table 1: Dataset overview(only show parts of 42 languages which have more than 30 articles) .

Usually the title of a Wikipedia article is a entry which is defined in the first sentence of summary. So we check every sentence in gold summary and pick up those sentences with title included. Then like other sequence labeling tasks we use BIO symbols to tag the position where title appears. We use symbol $[B\_MISC]$ to mark the beginning of the title phrase and $[I\_MISC]$ to mark rest parts of the title. We collect 26494 samples and make a language-wise division to generate train/valid/test dataset by a ratio of 8:1:1.

Subtitles(headers) in articles are more flexible than entries and can not be extracted directly. Those languages with large training corpus like English may train a seq2seq model but most low resource languages can not train a independent model. So we just use the same sequence labeling model to tag subtitles. When test each subtitle will try to tag on sentences from the corresponding paragraph.

Model	BERT+CRF	BERT+NMT
L(BERT)	12	12
H(BERT)	768	768
A(BERT)	12	12
A(Encoder)	3	-
H(BiLSTM)	256	256
H(LSTM)	-	256
E	-	128
# Parameters	1150675	1908755

Table 2: Hyperparameter setup for BERT based models.

We denote the number of BERT Transformer layers as L, the hidden size as H, the number of self-attention heads as A, the embedding hidden size as E. The hyperparameter setup is shown in table 2. The Hyperparameters stay the same as Google's version of BERT. We use the default setting of sberbank-ai on designing the BiLSTM and LSTM. Both BERT models are trained for 10 epochs, from which we observed that the validation precision can not be improved more.

8 Results

8.1 Extractive Summarization

We generate a single document summary first for all the given Wikipedia feature articles on training set from 42 languages provided. We use ROUGE package that measures skip n-gram overlap with the golden summaries for evaluation; we provide F-measure results and denote them by ROUGE1, ROUGE2 and ROUGE-F. The results are listed below in table 3.

From the table 3 we can see that even on corpus with low corpus the extracted summaries still get high ROUGE scores due to the unsupervised method. The top four results in the table have been bolded and the score reached nearly 0.5.

Language	ROUGE1	ROUGE2	ROUGE-L
AF	0.36309	0.07003	0.17838
AR	0.32613	0.05661	0.12668
AZ	0.17838	0.17838	0.08077
BG	0.31729	0.05438	0.14489
BS	0.20603	0.02805	0.10914
CA	0.41166	0.10017	0.16878
CS	0.36954	0.06100	0.13375
DE	0.34239	0.05847	0.16809
EL	0.42561	0.10546	0.22027
EN	0.46123	0.12328	0.19016
EO	0.32418	0.06765	0.17613
ES	**0.50390**	**0.14792**	**0.24045**
EU	0.18970	0.03056	0.09727
FA	0.37201	0.07671	0.15731
FI	0.18228	0.02617	0.09727
FR	**0.48166**	**0.15034**	**0.24975**
HE	0.18020	0.03957	0.09027
HR	0.23411	0.02854	0.11514
ID	0.32536	0.06755	0.13595
IT	0.40780	0.09950	0.20031
JA	0.38426	0.09298	0.17918
JV	0.24805	0.03990	0.12624
KA	0.17031	0.03514	0.09736
KO	0.22891	0.03904	0.10062
LI	0.22833	0.02795	0.12987
LV	0.19157	0.02774	0.09728
MR	**0.50092**	**0.15771**	**0.23461**
MS	0.29083	0.06304	0.13938
NL	0.37004	0.07344	0.17570
NN	0.27399	0.02045	0.13399
NO	0.35866	0.04805	0.14446
PL	0.31028	0.05631	0.14301
PT	**0.49376**	**0.16303**	**0.24452**
RO	0.38691	0.07458	0.15742
RU	0.26514	0.04773	0.12516
SK	0.21378	0.02534	0.09533
TH	0.46316	0.16334	0.16393
TR	0.26181	0.05757	0.10476
TT	0.12043	0.01173	0.06345
UK	0.12143	0.01198	0.06975
VI	0.45210	0.14085	0.15224
ZH	0.31551	0.06381	0.12747

Table 3: Performance on MultiLing Single-document Summarization

8.2 Entry Extraction

All results shown in this section are precision of predicting the title, not including subtitles.

First we test our unsupervised rule-based baselines. The Spacy toolkit can only support parts of the languages so we just collect results on these languages. Table 4 shows that on certain languages like DE, FR and PT, the first entity in the first sentence of summary can point out the entry of whole Wikipedia article with a probability of about 0.4. Even though the entry is not the first entity, we identify the subject entity using dependency parsing and get better results. The results from baselines prove that about half of the samples make a explicit description for the entry in the first sentence.

precision	NER	SUB
EN	0.105	0.507
DE	0.48	0.49
ES	0.013	0.451
FR	0.397	0.4
IT	0.021	0.523
PT	0.404	0.433
EL	-	0.4
NL	-	0.567
RU	0.25	-

Table 4: Results of Baselines. '-' means that Spacy does not support this language on NER or dependency parsing tasks.

	precision	recall	f1-score
B_MISC	0.779	0.603	0.680
I_MISC	0.721	0.668	0.693
micro avg	0.753	0.629	0.686
macro avg	0.750	0.635	0.687
weighted avg	0.755	0.629	0.685

Table 5: Results of BERT Based CRF Model.

As for BERT Based models, BIO precision shows that compared to baselines, BERT Based CRF model learned more rules to label the entry of an article and gains great precision improvement. The model is trained and tested on a language-mixed and shuffled dataset. We randomly divide the dataset for ten times and calculate the average precision.

The CRF Model reaches 0.779 and 0.721 precision on $[B\_MISC]$ and $[I\_MISC]$, which is a average precision on all 42 languages. It proves that CRF Model can make full use of pre-trained language model and perform well on low resource languages. The span precision, recall and f1-score of CRF Model are 0.703, 0.607 and 0.651 respectively.

The NMT Model reaches 0.810 and 0.782 precision on MISC tags. The span precision, recall and f1-score of CRF Model are 0.755, 0.780 and 0.767 respectively. The NMT Model outperforms CRF Model on all metrics.

Both the CRF and NMT Model reaches f1-score higher than baselines. It is worth noting that the f1-score is an average on all 42 languages and the baseline can only perform on few languages, which proves that the BERT based model can learn common syntactic rules from multilingual corpus and transfer well on low resource languages.

	precision	recall	f1-score
B_MISC	0.810	0.802	0.806
I_MISC	0.782	0.840	0.810
micro avg	0.798	0.817	0.808
macro avg	0.796	0.821	0.808
weighted avg	0.799	0.817	0.808

Table 6: Results of BERT Based NMT Model.

Although BERT are not pre-trained for language generation task like Neural Machine Translation, the BERT Based NMT Model still gets higher precision compared to CRF Model. There may be several reasons:

- Compared to CRF Model, NMT model incorporate another LSTM as decoder which expand the total amount of parameters and have large capacity when fitting data.
- NMT model uses embeded sequence as input both on encoder and decoder. With two supervised signal input the NMT can converge better than CRF Model when training the same epochs.

The NMT Model gets 4 points higher precision both on BIO precision and span precision compared to CRF Model.

9 Conclusion

We proposed a two-steps model for Wikipedia Headline Generation task. First we extract summaries that contain the key information of the whole article then a sequence labelling model using pre-trained language model is applied to further pick up key entry phrases. We test our extractive summarization model and sequence labelling model independently and reach good results compared to baselines.

References

Enrique Alfonseca, José Maria Guirao, and Antonio Moreno-Sandoval. 2003. Description of the uam system for generating very short summaries at duc-2003. In *DUC 2003: Document Understanding Conference, May 31–June 1, 2003, Edmonton, Canada*.

Carlos A Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. Heads: Headline generation as sequence prediction using an abstract

feature-rich space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 133–142.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear* 7.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* .

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5(2–3):123–286.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data .

Konstantin Lopyrev. 2015. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* .

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8).

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450* .

Rui Sun, Yue Zhang, Meishan Zhang, and Donghong Ji. 2015. Event-driven headline generation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pages 462–472.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. pages 1054–1059.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *IJCAI*. pages 4109–4115.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* .