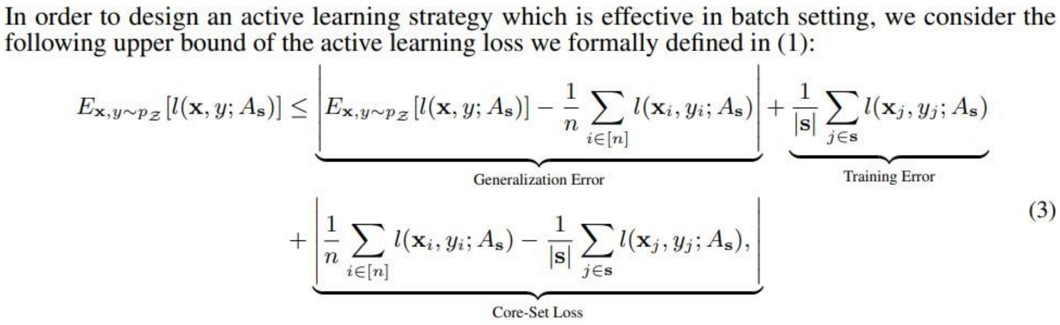
□ Emotional score[1]

Active learning : Active learning provides a dynamic approach that receiving human feedback from oracle to acquire data and learning from acquired data before selecting new and intriguing input-output data(Pickering et al 2022). Active learning is used to assembling efficient training dataset. Active learning is a reliable solution, not a compromise solution; It directs oracle to annotate on hard work.

Method introduction:

Core-set: From the analysis of the loss upper bound obtained by active learning, the loss upper bound of the model obtained by active learning picking out samples for training consists of three components: generalization error, training error, and core set error(Senzor et al 2017). The authors assume that the goodness of fit of the deep model ensures that both generalization error and training error are small, so the goal of active learning is to reduce the core set error. The authors convert the upper bound of the core set loss to the maximum distance between the remaining training samples and the selected labeled samples by the Lipschitzness of the loss function of the CNN model.



In order to design an active learning strategy which is effective in batch setting, we consider the following upper bound of the active learning loss we formally defined in (1):

$$E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] \le \underbrace{\left| E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] - \frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}})\right|}_{\text{Generalization Error}} + \underbrace{\frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}})}_{\text{Training Error}} + \underbrace{\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}}),\right|}_{\text{Core-Set Loss}} \tag{3}$$

Thus, core set is converted to a k-center set cover problem. Since n is fixed, we need to find a δ_s that minimize the upper bound of core-set loss.

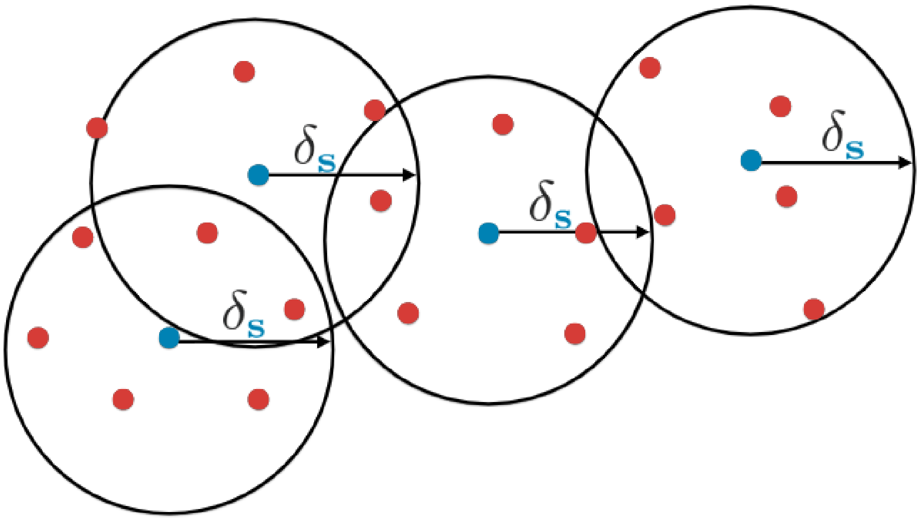
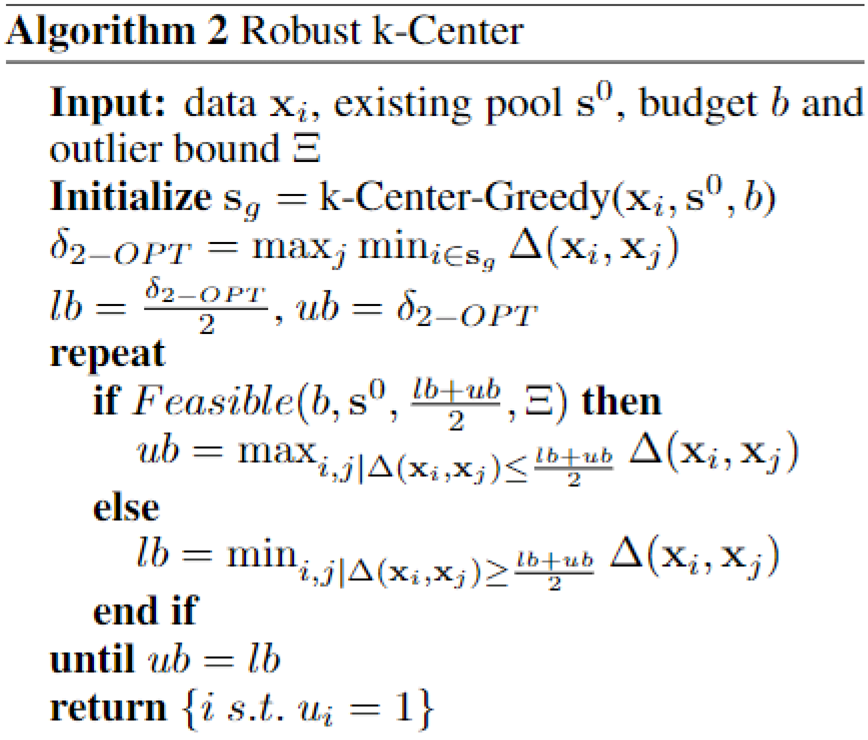


Figure 1: **Visualization of the Theorem 1**. Consider the set of selected points s and the points in the remainder of the dataset $[n] \setminus \mathbf{s}$, our results shows that if s is the $\delta_{\mathbf{s}}$ cover of the dataset, $\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i, y_i, A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j, y_j; A_{\mathbf{s}})\right| \leq \mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$

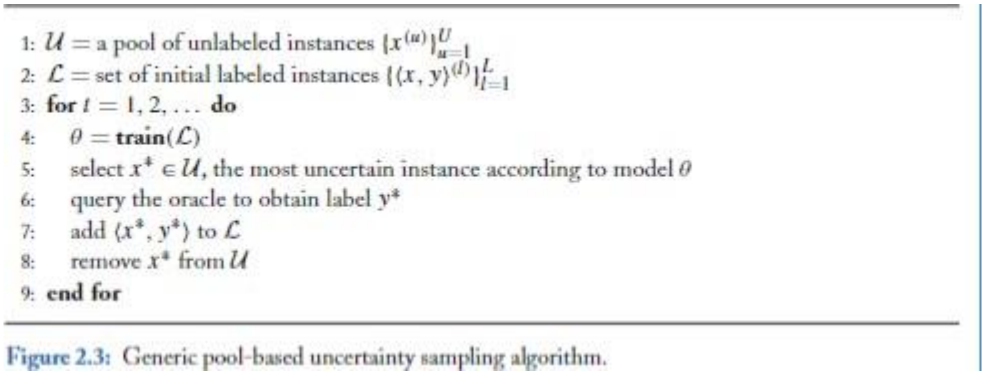
To solve the k-center problem, the authors proposed the following algorithm:



```
Algorithm 2 Robust k-Center
  Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$, budget $b$ and outlier bound $\Xi$
  Initialize $\mathbf{s}_g$ = k-Center-Greedy($\mathbf{x}_i, \mathbf{s}^0, b$)
  $\delta_{2-OPT} = \max_j \min_{i\in\mathbf{s}_g} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $lb = \frac{\delta_{2-OPT}}{2}$, $ub = \delta_{2-OPT}$
  repeat
    if $Feasible(b, \mathbf{s}^0, \frac{lb+ub}{2}, \Xi)$ then
      $ub = \max_{i,j|\Delta(\mathbf{x}_i,\mathbf{x}_j)\leq\frac{lb+ub}{2}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
    else
      $lb = \min_{i,j|\Delta(\mathbf{x}_i,\mathbf{x}_j)\geq\frac{lb+ub}{2}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
    end if
  until $ub = lb$
  return $\{i \ s.t. \ u_i = 1\}$
```

They start with an initialized set S_g using the greedy algorithm, and obtain the real value of OPT by binary search. They replace δ=OPT in MIP to return {i s.t. u_i = 1}.

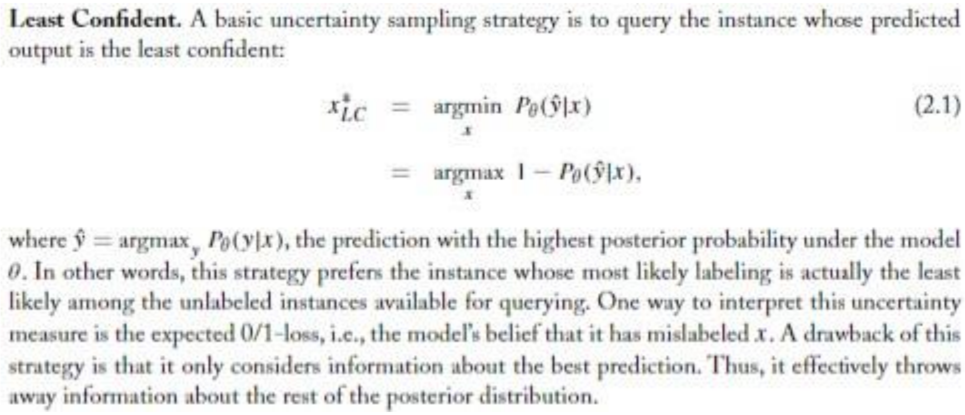
Uncertainty Sampling: Uncertainty Sampling(Settles et al. 2012) is a pool-based method that applied to measure the uncertainty of data and select the most representative samples. For every sample in the unllabeled pool, we learning from the accuired samples first before selecting new samples. After the model selected new samples, we query the oracle to label the selected samples. We store the labeled samples into labled pool and remove them from the unlabled pool. However, uncertainty sampling still has the possibility to select abnormal sample. A large number of queries may occur in sparse, noisy, or irrelevant areas of the input space.



```
$\mathcal{U}$ = a pool of unlabeled instances $\{x^{(u)}\}_{u=1}^{U}$
$\mathcal{L}$ = set of initial labeled instances $\{\langle x, y\rangle^{(l)}\}_{l=1}^{L}$
for $t = 1, 2, \ldots$ do
    $\theta$ = train($\mathcal{L}$)
    select $x^* \in \mathcal{U}$, the most uncertain instance according to model $\theta$
    query the oracle to obtain label $y^*$
    add $\langle x^*, y^*\rangle$ to $\mathcal{L}$
    remove $x^*$ from $\mathcal{U}$
end for
```

Figure 2.3: Generic pool-based uncertainty sampling algorithm.

Here we take advantage of least Confidence method(Settles et al. 2012) to select the least confident data from the pool of unlabeled instances.



Least Confident. A basic uncertainty sampling strategy is to query the instance whose predicted output is the least confident:

$$x^*_{LC} = \underset{x}{\operatorname{argmin}}\ P_\theta(\hat{y}|x) \qquad (2.1)$$
$$= \underset{x}{\operatorname{argmax}}\ 1 - P_\theta(\hat{y}|x),$$

where $\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$, the prediction with the highest posterior probability under the model θ. In other words, this strategy prefers the instance whose most likely labeling is actually the least likely among the unlabeled instances available for querying. One way to interpret this uncertainty measure is the expected 0/1-loss, i.e., the model's belief that it has mislabeled x. A drawback of this strategy is that it only considers information about the best prediction. Thus, it effectively throws away information about the rest of the posterior distribution.

Citation:

Sener, Ozan, and Silvio Savarese. "Active learning for convolutional neural networks: A core-set approach." arXiv preprint arXiv:1708.00489 (2017).

Settles, Burr. "Active learning." Synthesis lectures on artificial intelligence and machine learning 6.1 (2012): 1-114.

Chinese Data:

l Model introduction

The Bidirectional Encoder Representation from Transformers(BERT) model is different from previous one-way language models. It is pre-trained by Masked Language Model and uses a deep bi-directional transformer component to build the entire model to generate deep bi-directional language representations (Devlin 2018). In the Fig.1 left block, it demonstrates the pretraining step in the Transformer model that several inputs are musked while the model needs to generate with the same output. In our task, we want to predict if a piece of news is positive or negative.We need to transfer and adapt the pretrain model(Cui et al 2019) into our task.

2 TANDA:

To transfer the pre-trained Transformer language model to our emotianl score prediction task, we firstly perform the standard fine-tuning step with a large and unlabled news dataset(Siddhant et al.). But the model is still not stable due to the lack of labels so that we perform the second fine-tuning step to adapt the model into our task. In practice, We use 50 labeld news to do the first fine-tuning step per epoch and implement 19 epoches in total. In each epoch, we adopt active learning methods to select the most informative data. We adopt AdamW optimizer with learning rate of 5×10^{-5}, batch size of 50, and epoch equals to 50 for the transfer step. We set the maximum sequence length for Bert to 100 tokens.

After we finish training and get the best performance model, we pack it up and prepare for emotional score prediction task.

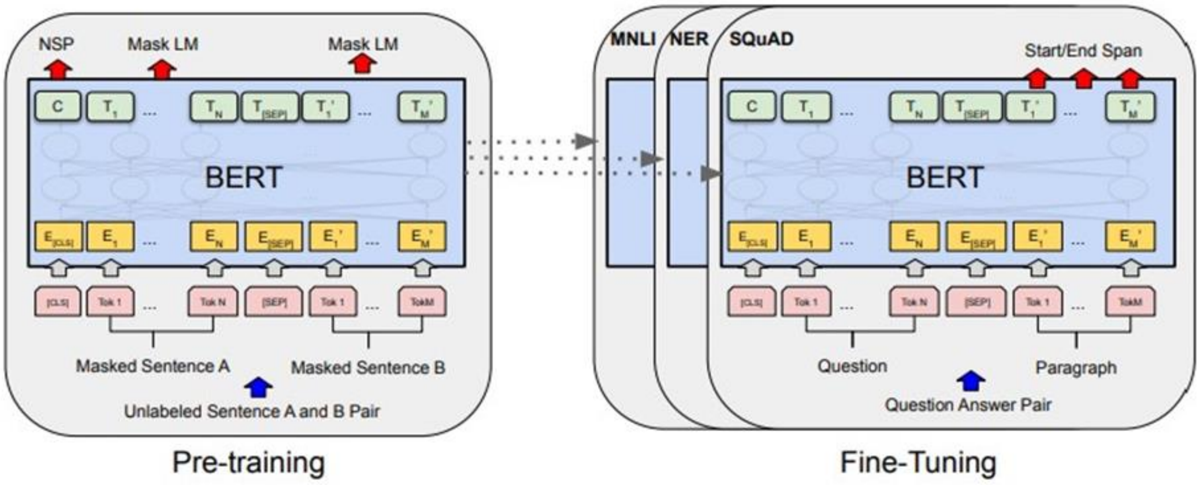


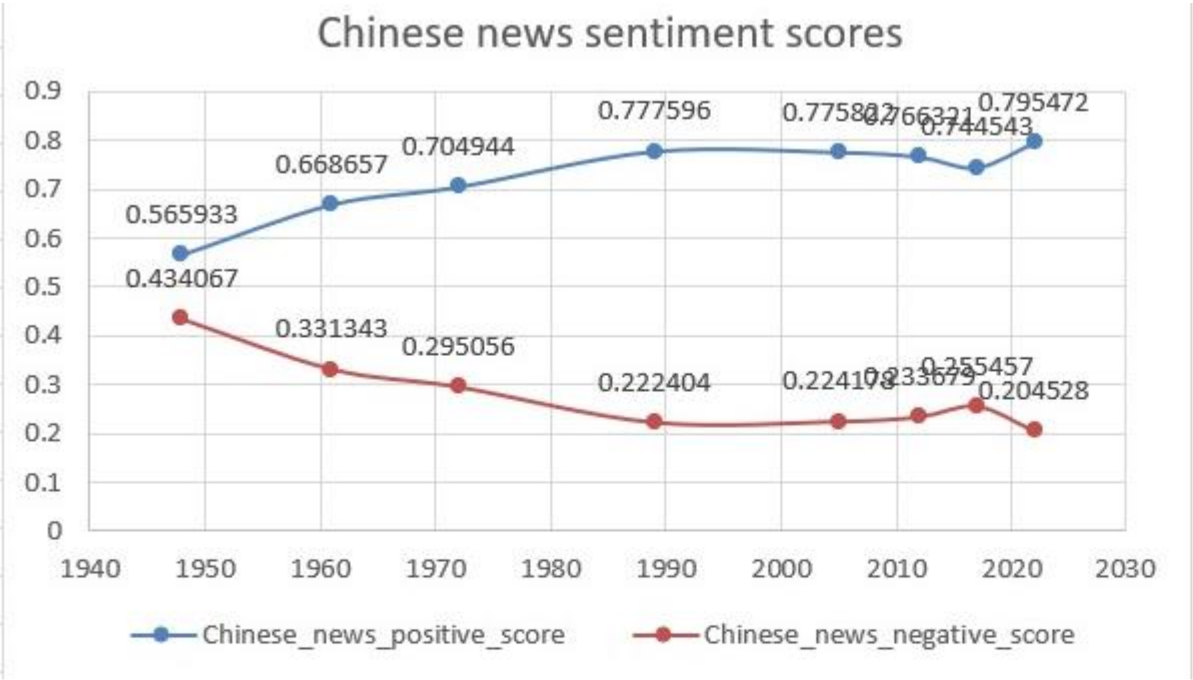
Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

l Data introduction

29,998 Chinese news transcripts.

l Empirical process and results

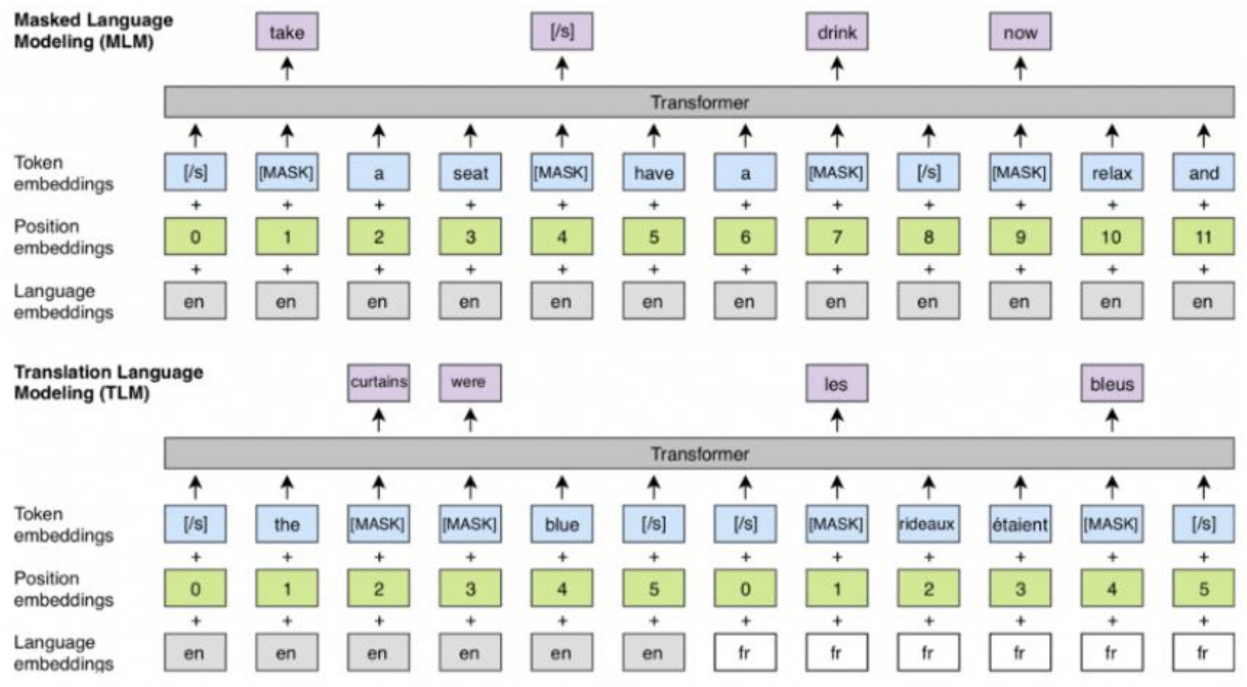
We clean punctuations and numbers in the text at first step. After that, we import our fine-tuned Bert model and divides each news into chunks of length 512. In the output layer of the model, we obtain the dynamic sentiment score of each chunks by softmax according to the number of categories to get the positive, negative, and neutral probability. And then, we merge and average the positice and negative probability of all the chunks in each news. Finally, we take the average probabilities as output scores.



Japanese Data:

l Model introduction

Xlm:is a Bert-based model that uses byte pair encoding (BPE) as input, unlike Bert's characters or words. xlm uses BPE to slice the input by the most common sub-word in all languages as a way to increase the vocabulary shared across languages. Then, each training sample of xlm contains two sentences with different meanings in the same language, so that the token that is masked in another language can be predicted by the context of one language. The model also accepts the language ID and the sequential information of tokens in different languages, which is the positional encoding, and these new metadata can help the model learn the relationship between tokens in different languages. (Horev 2019)


Masked Language Modeling (MLM)
take
[/s]
drink
now
Transformer
Token embeddings
[/s] [MASK] a seat [MASK] have a [MASK] [/s] [MASK] relax and
Position embeddings
0 1 2 3 4 5 6 7 8 9 10 11
Language embeddings
en en en en en en en en en en en en
Translation Language Modeling (TLM)
curtains
were
les
bleus
Transformer
Token embeddings
[/s] the [MASK] [MASK] blue [/s] [/s] [MASK] rideaux étaient [MASK] [/s]
Position embeddings
0 1 2 3 4 5 0 1 2 3 4 5
Language embeddings
en en en en en en fr fr fr fr fr fr

Comparison of a single language modeling (MLM) similar to BERT, and the proposed dual-language modeling (TLM). Source: XLM

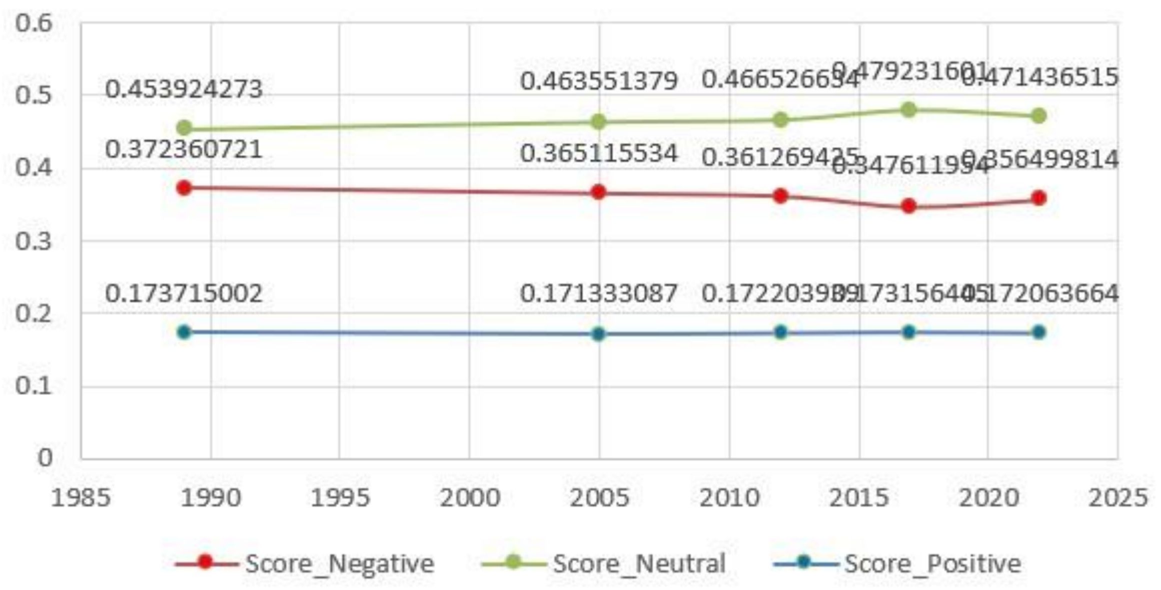
l Data introduction

29,998 Japanese news transcripts.

l Empirical process and results

Specifically, at the model fine-tuning stage, we directly use the xlm model after fine-tuning by Barbieri et al, based on 198M tweets(2). And we set the category to classfy eqauls to three. We clean punctuations and numbers in the text at first step. After that, we import our fine-tuned XLM model and divides each news into chunks of length 512. In the output layer of the model, we obtain the dynamic sentiment score of each chunks by softmax according to the number of categories to get the positive, negative, and neutral probability. And then, we merge and average the positice, negative and neutral probability of all the characters in each news. Finally, we take the average probability as output scores.

Results□



Horev, Rani. "Xlm—enhancing bert for cross-lingual language model." Towards Data Science, on Medium, February 12 (2019).

Barbieri, Francesco, Luis Espinosa Anke, and Jose Camacho-Collados. "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond." *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022.

Cui, Yiming, et al. "Revisiting pre-trained models for Chinese natural language processing." *arXiv preprint arXiv:2004.13922* (2020).

Cui, Yiming, et al. "Pre-Training with Whole Word Masking for Chinese BERT." *arXiv*, 2019, https://doi.org/10.1109/TASLP.2021.3124365.