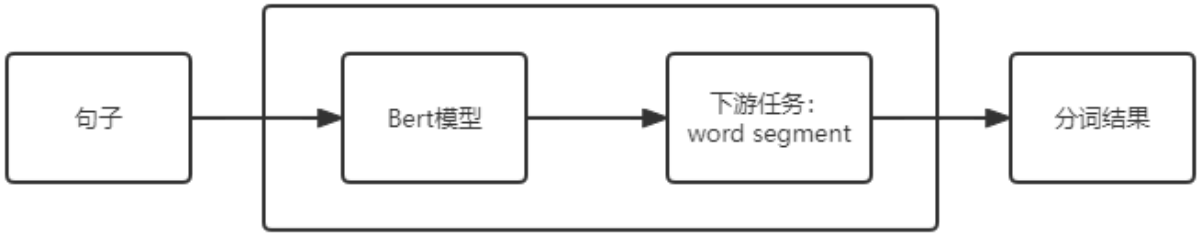
使用bert等预训练模型进行分词

对于ex2中的jieba分词部分，采用bert预训练模型进行分词任务。

整体结构如下



bert预训练模型的下载和导入

相关预训练模型我们通过🤗huggingface进行查找下载，关于huggingface网站的介绍可参考：Hugging Face使用向

这里我们选用了`ckiplab/albert-tiny-chinese-ws`，`ckiplab/albert-base-chinese-ws`，`ckiplab/bert-tiny-chinese-ws`，`ckiplab/bert-base-chinese-ws`，预训练模型的测试表现如下

> ### Model Performance
>
> The following is a performance comparison between our model and other models.
>
> The results are tested on a traditional Chinese corpus.
>
> 以下是我們的模型與其他的模型之性能比較。
>
> 各個任務皆測試於繁體中文的測試集。

Model	#Parameters	Perplexity†	WS (F1)‡	POS (ACC)‡	NER (F1)‡
ckiplab/albert-tiny-chinese	4M	4.80	96.66%	94.48%	71.17%
ckiplab/albert-base-chinese	11M	2.65	97.33%	95.30%	79.47%
ckiplab/bert-tiny-chinese	12M	8.07	96.98%	95.11%	74.21%
ckiplab/bert-base-chinese	102M	1.88	97.60%	95.67%	81.18%

> † Perplexity; the smaller the better.
>
> † 混淆度；數字越小越好。
>
> ‡ WS: word segmentation; POS: part-of-speech; NER: named-entity recognition; the larger the better.

‡ WS: 斷詞； POS: 詞性標記； NER: 實體辨識； 數字越大越好。

为了方便规划存储空间，我们在这里列出了四个预训练模型的下载大小

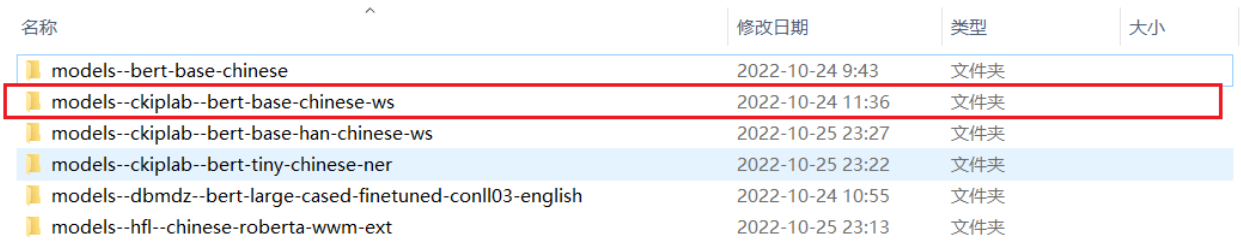
预训练模型	大小
`ckiplab/albert-tiny-chinese-ws`	15.3 MB
`ckiplab/albert-base-chinese-ws`	38.1 MB
`ckiplab/bert-tiny-chinese-ws`	43.7 MB
`ckiplab/bert-base-chinese-ws`	388 MB

以`ckiplab/bert-base-chinese-ws`为例

从🤗huggingface进行下载model和tokenizer的相关代码，并导入。

```
# 导入相关库
from transformers import AutoModelForTokenClassification,BertTokenizerFast
# 下载model和tokenizer
model_name = "ckiplab/bert-base-chinese-ws"
model = AutoModelForTokenClassification.from_pretrained(model_name)
tokenizer = BertTokenizerFast.from_pretrained(model_name)
```

下载完成后进入`C:\Users\用户名\.cache\huggingface\hub`便可以看见下载完成的预训练模型



下游任务构建word segment

✨ 注意，整个代码是基于`pytorch`

成功导入预训练模型后，我们需要构建下游任务，即中文分词。

```
class WordSegmenter():
    r"""
    使用预训练模型进行中文分词
    """
    def __init__(self,model,tokenizer,device: Union[int, torch.device] = -1)->None:
        self.model = model
        self.tokenizer = tokenizer
        if isinstance(device, torch.device):
            self.device = device
        else:
            self.device = torch.device("cpu" if device < 0 else f"cuda:{device}")

        self.model.to(self.device)
```

其中，如果支持使用gpu的话，可以通过`pytorch`相关函数进行设备设置。

整个下游任务的构建代码具体请看`wordsegment.py`文件。

调用流程

示例

> 引用的语句来自**林清玄《木炭与沉香》**

```
import torch
from wordsegment import WordSegmenter
from transformers import AutoModelForTokenClassification,BertTokenizerFast

model_name = "ckiplab/bert-base-chinese-ws"
model = AutoModelForTokenClassification.from_pretrained(model_name)
tokenizer = BertTokenizerFast.from_pretrained(model_name)
ws = WordSegmenter(model=model,tokenizer=tokenizer,device = torch.device("cuda" if
torch.cuda.is_available() else "cpu"))
sentence = ["人生的缺憾，最大的就是和别人比较，和高人比较使我们自卑；和俗人比较，使我们下流；和下人比较，使我们骄满。外来的比较是我们心灵动荡不能自在的来源，也是的大部分的人都迷失了自我，障蔽了自己心灵原有的氤氲馨香。"]
print(ws.segment(sentence))
```

结果如下

```
(pytorch_cpu) PS C:\Users\lenovo\Desktop\demo> python C:\Users\lenovo\Desktop\demo\test.py
[['人生', '的', '缺憾', '，', '最', '大', '的', '就', '是', '和', '别人', '比较', '，', '和', '高人',
'比较', '使', '我们', '自卑', '；', '和', '俗人', '比较', '，', '使', '我们', '下流', '；', '和', '下
人', '比较', '，', '使', '我们', '骄满', '。', '外来', '的', '比较', '是', '我们', '心灵', '动荡', '不
能', '自在', '的', '来源', '，', '也', '是', '的', '大部分', '的', '人', '都', '迷失', '了', '自我', '
，', '障蔽', '了', '自己', '心灵', '原有', '的', '氤氲', '馨香', '。']]
```

对于同样的句子，使用不同的预训练模型的结果也可能不一样

输入为

> 到了一个新地方，有人爱逛百货公司，有人爱逛书店，我宁可去逛逛菜市。看看生鸡活鸭、新鲜水灵的瓜菜、彤红的辣椒，热热闹闹，挨挨挤挤，让人感到一种生之乐趣。
>
> ——汪曾祺《做饭》

预训练模型	结果
`ckiplab/albert-tiny-chinese-ws`	[['到', '了', '一', '个', '新', '地方', '，', '有', '人', '爱', '逛', '百货', '公司', '，', '有', '人', '爱', '逛', '书', '店', '，', '我', '宁', '可', '去', '逛逛', '菜市', '。', '看看', '生', '鸡', '活', '鸭', '、', '新', '鲜', '水灵', '的', '瓜菜', '、', '彤红', '的', '辣椒', '，', '热', '热闹闹', '，', '挨挨', '挤', '挤', '，', '让', '人', '感到', '一', '种', '生', '之', '乐趣', '。']]
`ckiplab/albert-base-chinese-ws`	[['到', '了', '一', '个', '新', '地方', '，', '有', '人', '爱', '逛', '百货公司', '，', '有', '人', '爱', '逛', '书店', '，', '我', '宁', '可', '去', '逛逛', '菜市', '。', '看看', '生鸡', '活', '鸭', '、', '新鲜', '水灵', '的', '瓜菜', '、', '彤红', '的', '辣椒', '，', '热热', '闹闹', '，', '挨挨挤挤', '，', '让', '人', '感到', '一', '种', '生', '之', '乐趣', '。']]
`ckiplab/bert-tiny-chinese-ws`	[['到', '了', '一', '个', '新', '地方', '，', '有', '人', '爱', '逛', '百货', '公司', '，', '有', '人', '爱', '逛', '书', '店', '，', '我', '宁', '可', '去', '逛逛', '菜市', '。', '看看', '生', '鸡', '活鸭', '、', '新', '鲜', '水', '灵', '的', '瓜菜', '、', '彤红', '的', '辣椒', '，', '热热闹闹', '，', '挨挨挤挤', '，', '让', '人', '感到', '一', '种生', '之', '乐', '趣', '。']]
`ckiplab/bert-base-chinese-ws`	[['到', '了', '一', '个', '新', '地方', '，', '有', '人', '爱', '逛', '百货公司', '，', '有', '人', '爱', '逛', '书店', '，', '我', '宁可', '去', '逛逛', '菜市', '。', '看看', '生', '鸡', '活', '鸭', '、', '新鲜', '水灵', '的', '瓜菜', '、', '彤红', '的', '辣椒', '，', '热热闹闹', '，', '挨挨挤挤', '，', '让', '人', '感到', '一', '种', '生', '之', '乐趣', '。']]