run_summarization.py			
function_name	**description**	**attributes**	**return**
evaluate()	function for model evaluation and prediction.	args: commandline arguments	-
save_summaries()	Write the summaries in fies that are prefixed by the original files' name with the `_summary` appended.	original_document_names: List[string] (Name of the document that was summarized) path: string (Path were the summaries will be written) summaries: List[string] (summaries that we produced)	saves the summary as a text file to a given path
format_summary()	Transforms the output of the `from_batch` function into nicely formatted summaries.	translation: itirator (list or tuple) raw_summary: string	summary: tuple (formatted summaries)
build_data_itirator()	creates a data itiirator with the help of torch's DataLoader module.	args: commandline arguments tokenizer: BertTokenizer	itirator : batch
load_and_cache_examples()	process the documents that are located in the specified folder. The preprocessing will work on any document that is reasonably formatted.	args: commandline arguments tokenizer: BertTokenizer	TODO
collate()	Collate formats the datapassed to the data loader. In particular we tokenize the data batch after batch to avoid keeping them all in memory. We output the data as a namedtuple to fit the original BertAbs's API.	data: List tokenizer: BertTokenizer block_size: int device: CPU or cuda	batch : namedtuple fields: document_names batch_size src segs mask_src tgt_str
decode_summary()	Decode the summary and return it in a format suitable for evaluation.	summary_tokens: token_ids tokenizer: BertTokenizer	sentences: List(string)
main()			
document_dir_is_valid()	Check if provided path is valid.	path: string	boolean

References:
DataLoader : https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader
BertTokenizer : https://huggingface.co/transformers/model_doc/bert.html#berttokenizer