BBC News Summary and Text Summarization

This project aims to demonstrate a text summarization approach using a Sequence to Sequence (Seq2Seq) model on the BBC News Summary dataset.

Dependencies

The following Python packages are required to run this script:

- `os` : to handle directory and file operations.
- `chardet` : to detect the character encoding of the text files.
- `numpy` : for numerical operations on arrays.
- `tensorflow` : for building and training the neural network models.
- `keras` : part of TensorFlow, used here for model definition and training.

You can install these dependencies via pip:

```
pip install chardet numpy tensorflow
```

Project Structure

- **read_files(directory)**: This function reads all files in the specified directory, detecting and using the correct encoding for each file. It returns a list of file contents.
- **load_data(main_directory)**: This function loads and categorizes texts and their summaries from separate directories within the main directory. It uses predefined categories (e.g., business, entertainment).

Data Loading and Preprocessing

1. **Main Directory Setup**: Set the `main_directory` to the path where the BBC News Summary data is stored.
2. **Data Reading**: Use `load_data` function to read the news articles and summaries from the directory.

Text Tokenization and Padding

- **Tokenizer Setup**: Initialize a Keras Tokenizer and fit it on both the texts and summaries to create a word index.
- **Sequence Conversion**: Convert text and summaries into sequences of integers using the tokenizer.
- **Padding**: Pad these sequences to ensure consistent length inputs for training the model.

Model Building: Seq2Seq Architecture

1. **Encoder-Decoder Architecture**: Define an LSTM-based encoder-decoder model.
2. **Model Compilation**: Compile the model using the Adam optimizer and sparse categorical crossentropy as the loss function.

Training

- **Prepare Input and Target Data**: Configure input and target data for the decoder.
- **Model Training**: Train the model using the prepared data.

Inference Setup

1. **Encoder Model**: Define a model that captures the internal states of the encoder.
2. **Decoder Model**: Setup the decoder to predict the next word in the sequence given the previous word and the encoder states.

Summary Generation

- **decode_sequence function**: For a given input sequence, this function uses the trained model to generate a text summary.
- **Interactive Summarization**: Allows users to enter a text and get a summary generated by the model.

Example Usage

- After training, the script provides an example of summarizing a text.
- Users can also input their own texts to get summaries in real-time.

Installation of Dependencies

Make sure to install the necessary Python packages:

```
pip install chardet
```

This project highlights the practical implementation of neural networks in processing natural language for tasks like summarization, showcasing both the challenges and solutions in handling sequence data.