```
import wave
from abc import abstractmethod
from typing import Optional

from swarm_models.base_llm import BaseLLM


class BaseTTSModel(BaseLLM):
    """Base class for all TTS models.

    Args:
        BaseLLM (_type_): _description_
        model_name (_type_): _description_
        voice (_type_): _description_
        chunk_size (_type_): _description_
        save_to_file (bool, optional): _description_. Defaults to False.
        saved_filepath (Optional[str], optional): _description_. Defaults to None.

    Raises:
        NotImplementedError: _description_

    Methods:
        save: save the model to a file.
        load: load the model from a file.
        run: run the model on the given task.
        __call__: call the model on the given task.
```

```
    save_to_file: save the speech data to a file.

"""

def __init__(
    self,
    model_name,
    voice,
    chunk_size,
    save_to_file: bool = False,
    saved_filepath: Optional[str] = None,
):
    self.model_name = model_name
    self.voice = voice
    self.chunk_size = chunk_size
    self.save_to_file = save_to_file
    self.saved_filepath = saved_filepath

def save(self, filepath: Optional[str] = None):
    """Save the model to a file.

    Args:
        filepath (Optional[str], optional): _description_. Defaults to None.
    """

def load(self, filepath: Optional[str] = None):
```

```
    """Load the model from a file.

    Args:
        filepath (Optional[str], optional): _description_. Defaults to None.
    """

@abstractmethod
def run(self, task: str, *args, **kwargs):
    """Run the model on the given task.

    Args:
        task (str): _description_
    """

def __call__(self, task: str, *args, **kwargs):
    """Call the model on the given task.

    Args:
        task (str): _description_

    Returns:
        _type_: _description_
    """
    return self.run(task, *args, **kwargs)

def save_to_file(self, speech_data, filename):
```

```
"""Save the speech data to a file.


Args:
    speech_data (bytes): The speech data.
    filename (str): The path to the file where the speech will be saved.
"""
with wave.open(filename, "wb") as file:
    file.setnchannels(1)
    file.setsampwidth(2)
    file.setframerate(22050)
    file.writeframes(speech_data)
```