```
from typing import Any

from tenacity import retry, stop_after_attempt, wait_exponential

from swarms.prompts.prompt_generator import (
    prompt_generator_sys_prompt as second_sys_prompt,
)
from swarms.prompts.prompt_generator_optimizer import (
    prompt_generator_sys_prompt,
)
from swarms.utils.loguru_logger import initialize_logger

logger = initialize_logger(log_folder="ape_agent")


@retry(
    stop=stop_after_attempt(3),
    wait=wait_exponential(multiplier=1, min=4, max=10),
)
def auto_generate_prompt(
    task: str = None,
    model: Any = None,
    max_tokens: int = 4000,
    use_second_sys_prompt: bool = True,
    *args,
    **kwargs,
):
    """
    Generates a prompt for a given task using the provided model.

    Args:
    task (str, optional): The task for which to generate a prompt.
    model (Any, optional): The model to be used for prompt generation.
    max_tokens (int, optional): The maximum number of tokens in the generated prompt. Defaults to 4000.
    use_second_sys_prompt (bool, optional): Whether to use the second system prompt. Defaults to True.

    Returns:
    str: The generated prompt.
    """
    try:
        system_prompt = (
            second_sys_prompt.get_prompt()
            if use_second_sys_prompt
            else prompt_generator_sys_prompt.get_prompt()
        )
        output = model.run(
            system_prompt + task, max_tokens=max_tokens
        )
        print(output)
        return output
```

```
except Exception as e:
    logger.error(f"Error generating prompt: {str(e)}")
    raise
```