```
from termcolor import colored


from swarms.prompts.autobloggen import (
    DRAFT_AGENT_SYSTEM_PROMPT,
    REVIEW_PROMPT,
    SOCIAL_MEDIA_SYSTEM_PROMPT_AGENT,
    TOPIC_GENERATOR,
)


# Prompts
topic_selection_task = (
    "Generate 10 topics on gaining mental clarity using ancient"
    " practices"
)



class AutoBlogGenSwarm:
    """
    AutoBlogGenSwarm


    Swarm Agent
    Topic selection agent -> draft agent -> review agent -> distribution agent


    Topic Selection Agent:
    - Generate 10 topics on gaining mental clarity using Taosim and Christian meditation
```

Draft Agent:

- Write a 100% unique, creative and in human-like style article of a minimum of 5,000 words using headings and sub-headings.

Review Agent:

- Refine the article to meet PositiveMeds stringent publication standards.

Distribution Agent:

- Social Media posts for the article.

Example:

```
from swarms.autobloggen import AutoBlogGenSwarm
swarm = AutoBlogGenSwarm()
swarm.run()
```

"""

```
def __init__(
    self,
    llm,
    objective: str = "Clicks and engagement",
    iterations: int = 3,
    topic_selection_task: str = topic_selection_task,
```

```
        max_retries: int = 3,
        retry_attempts: int = 3,
         topic_selection_agent_prompt: str = f"Your System Instructions: {TOPIC_GENERATOR}, Your current task: {topic_selection_task}",
    ):
        self.llm = llm()
        self.topic_selection_task = topic_selection_task
        self.topic_selection_agent_prompt = (
            topic_selection_agent_prompt
        )
        self.objective = objective
        self.iterations = iterations
        self.max_retries = max_retries
        self.retry_attempts = retry_attempts

    def print_beautifully(self, subheader: str, text: str):
        """Prints the text beautifully"""
        print(
            colored(
                f"""
            -------------------------------------
            {subheader}
            ----------------------------


            {text}
```

```
        """,
            "blue",
        )
    )

def social_media_prompt(self, article: str):
    """Gets the social media prompt"""
    prompt = SOCIAL_MEDIA_SYSTEM_PROMPT_AGENT.replace(
        "{{ARTICLE}}", article
    ).replace("{{GOAL}}", self.objective)
    return prompt

def get_review_prompt(self, article: str):
    """Gets the review prompt"""
    prompt = REVIEW_PROMPT.replace("{{ARTICLE}}", article)
    return prompt

def step(self):
    """Steps through the task"""
    topic_selection_agent = self.llm(
        self.topic_selection_agent_prompt
    )
    topic_selection_agent = self.print_beautifully(
        "Topic Selection Agent", topic_selection_agent
    )
```

```
    draft_blog = self.llm(DRAFT_AGENT_SYSTEM_PROMPT)
    draft_blog = self.print_beatiufully("Draft Agent", draft_blog)

    # Agent that reviews the draft
    review_agent = self.llm(self.get_review_prompt(draft_blog))
    review_agent = self.print_beautifully(
        "Review Agent", review_agent
    )

    # Agent that publishes on social media
    distribution_agent = self.llm(
        self.social_media_prompt(article=review_agent)
    )
    distribution_agent = self.print_beautifully(
        "Distribution Agent", distribution_agent
    )

def run(self):
    """Runs the swarm"""
    for attempt in range(self.retry_attempts):
        try:
            for i in range(self.iterations):
                self.step()
        except Exception as error:
            print(
                colored(
```

```
                (
                    "Error while running AutoBlogGenSwarm"
                    f" {error}"
                ),
                "red",
            )
        )
        if attempt == self.retry_attempts - 1:
            raise


def update_task(self, new_task: str):
    """
    Updates the task of the swarm

    Args:
        new_task (str): New task to be performed by the swarm

    """
    self.topic_selection_agent = new_task
```