

ГЛАВА 1. Обзор предметной области

1.1 Задача автопромттинга

Автопромттинг – это задача автоматической генерации или оптимизации промптов для взаимодействия с языковыми моделями, с целью повышения качества, релевантности и точности их ответов. В рамках автопромттинга рассматриваются подходы доработки промптов, позволяющие повысить не только интерпретируемость контекста самой моделью, но и заданную пользователем полезность получаемого от модели ответа.

1.2 Возможные альтернативы

Существует несколько различных подходов улучшения взаимодействия с языковой моделью в рамках пользовательской задачи.

1.2.1 Ручной промптинг

В рамках ручного создания промптов пользователю приходится самому создавать и редактировать запросы к модели, опираясь на собственные выводы и рассуждения. Такой подход полностью завязан на теоретических знаниях человека, из-за чего часто бывает ограничен с точки зрения конечного результата. Как показывает практика, простое и наивное написание инструкций редко приводит к желаемому выводу. Конечно, всегда можно попробовать уточнить контекст задачи для модели в формате диалога, но это занимает определенное время, при этом все также не гарантируя значительных улучшений. Из чего можно сделать вывод, что ручное форматирование промптов – затратный с точки зрения времени способ, требующий экспертных знаний в области решаемой проблемы от пользователя.

1.2.2 Использование шаблонов инструкций

Многими исследователями в области взаимодействия с большими языковыми моделями уже было показано и доказано, что общение с нейросетью с использованием определенных шаблонов для запросов

существенно увеличивает шансы на получение качественных ответов. На текущий момент было представлено немалое количество методов промптинга, включающие в себя такие подходы, как Zero-Shot, Few-Shot, Chain-of-Thought и другие. В большинстве случаев они основываются на предоставлении модели дополнительных знаний о задаче, примеров правильных ответов на похожие вопросы, а также на генерации моделью рассуждений относительно проблемы. Примерная схема Few-Shot промптинга представлена на рисунке 1.

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

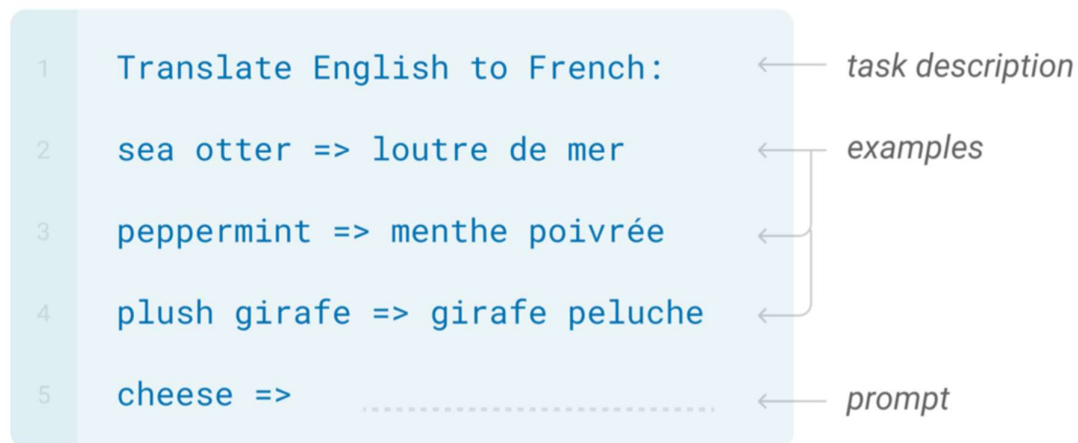


Рисунок 1 – Пример использования техники Few-Shot, показывающей модели варианты правильного исполнения запроса (в данном случае – перевода с английского на французский).

Из преимуществ данного подхода можно выделить то, что такие техники промптирования действительно работают и улучшают взаимодействия с моделью. Но все также требуют специализированных знаний, ведь не каждый пользователь знает похожие приемы или вообще хочет задумываться об их существовании при работе с моделью.

1.2.3 Обучение модели под конкретную задачу

Обучение модели на конкретных задачах, по-другому называемое дообучение или fine-tuning, — это тонкая дополнительная настройка модели на специализированных данных для определенных задач. Такой метод действительно практически убирает зависимость исполнения модели от промптов на данной задаче, но такое добавочное обучение — совершенно не бесплатная альтернатива. Во-первых, оно требует доступ к внутренним весам модели, во-вторых, нужны время и вычислительные ресурсы, чтобы эти веса оптимизировать, в-третьих, нужен специализированный набор данных под текущую задачу, на котором и будет оцениваться и оптимизироваться модель. Исходя из этого, дообучение — хорошая с точки зрения итогового качества альтернатива автопромπτингу, требующая немалых ресурсов для проведения.

1.3 Подходы в автопромπτинге

В рамках самого автопромπτинга также существуют различные методы и подходы по оптимизации. Далее будут рассмотрены самые главные из них.

1.3.1 Градиентные методы

Градиентные методы (gradient-based) автопромπτинга используют информацию о градиентах функции потерь для оптимизации параметров промпта. Эти подходы представляют промпт как набор векторов или токенов и обновляют их, основываясь на градиентном спуске, за счет обратного распространения ошибки. Такие алгоритмы очень завязаны на структуре промпта, так как количество обучаемых векторов токенов, из которых и будет собираться сам промпт — фиксировано, и задается при инициализации. За свою высокую итоговую точность градиентные методы платят большими вычислительными затратами (из-за пересчета градиентов всех весов модели на каждой итерации), а также зачастую полностью нечитаемыми с точки зрения людей итоговыми инструкциями. Примерами таких подходов являются Prompt tuning [4] и AutoPrompt [5].