

Автоматизация процесса создания семантической модели программных интерфейсов приложений

И.В. Евченко

Аннотация — В данной статье проанализирована проблема автоматизации создания семантической модели программных интерфейсов приложений, обеспечивающей преобразование пользовательских запросов, сформулированных на естественном языке, в корректные вызовы программных методов. Рассмотрены существующие подходы к разработке естественно-языковых пользовательских интерфейсов, включая использование методов обработки естественного языка, предобученных языковых моделей и специализированных тезаурусов. Предложен алгоритм автоматизированного формирования семантической модели программных интерфейсов приложений. Реализована тестовая среда для проверки разработанного подхода, обеспечивающая динамическое сопоставление пользовательских запросов с методами API. На основе полученных результатов можно утверждать, что предложенный метод повышает точность и полноту преобразования запросов, минимизирует ручное вмешательство и обеспечивает адаптацию модели к изменениям структуры API.

Ключевые слова — Обработка естественного языка, семантическая модель, API.

I. ВВЕДЕНИЕ

В современном мире программные интерфейсы приложений (от англ. Application programming interface, API) играют ключевую роль в обеспечении взаимодействия между системами и сервисами. Однако работа с API часто требует от пользователей высокого уровня технической подготовки, включая знание структуры запросов, параметров и форматов данных. Это ограничивает доступ к возможностям автоматизации и анализа данных для широкого круга пользователей.

Для преодоления этих барьеров актуальной становится задача создания естественно-языковых интерфейсов, которые позволяют преобразовывать привычные формы общения на естественном языке в корректные API-запросы.

Основой таких интерфейсов является семантическая модель, обеспечивающая точное и понятное отображение пользовательских запросов в вызовы API.

Однако разработка семантической модели для API сталкивается с рядом трудностей, таких как сложность обработки неоднозначных запросов, необходимость

поддержания актуальности модели в условиях изменения структуры API и зависимость от ручной настройки.

Разработка автоматизированного подхода к созданию семантической модели API позволит минимизировать ручное вмешательство, обеспечить точность обработки запросов и адаптацию модели к изменениям структуры API. Поэтому важно рассмотреть существующие подходы и их ограничения, а также определить направления, в которых возможно развитие методов автоматизации.

II. ОБЗОР СУЩЕСТВУЮЩИХ РАБОТ

В последние годы большое внимание уделяется разработке естественно-языковых интерфейсов, которые облегчают взаимодействие с программными интерфейсами приложений.

Существующие решения демонстрируют значительный прогресс в области обработки естественного языка (от англ. Natural Language Processing, NLP) и интеграции с API, однако сохраняются проблемы, связанные с автоматизацией формирования семантической модели.

В области аналитических систем, таких как Power BI и Tableau, естественно-языковые интерфейсы предоставляют пользователям возможность формулировать сложные запросы на естественном языке. [1] Однако, несмотря на удобство использования, эти подходы ограничены фиксированными правилами преобразования запросов и не всегда поддерживают динамическое обновление моделей при изменении API.

Реализации в других областях, таких как здравоохранение и электронная коммерция, также показывают потенциал использования семантических моделей для упрощения взаимодействия с API. Например, системы типа Nuance Dragon Medical [2] интегрируют голосовые интерфейсы с электронными медицинскими картами (от англ. Electronic Medical Record, EMR), а чат-боты компаний Amazon и Shopify успешно преобразуют естественно-языковые запросы в вызовы API для управления товарами и заказами. Однако эти системы, как правило, требуют значительных усилий для ручной настройки моделей и обучения их на специфичных наборах данных.



Рисунок 1. Пример интерфейса системы Power BI

Ряд научных исследований направлен на совершенствование подходов к формированию семантических моделей. Например, работы, использующие краудсорсинг для сбора данных [3], и архитектуры на основе трансформеров демонстрируют улучшение в генерализации моделей. [4]

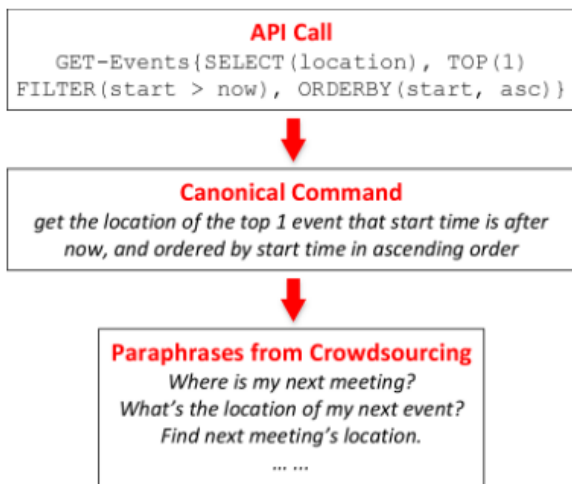


Рисунок 2. Схема сбора данных краудсорсинга

В то же время подходы, основанные на шаблонах предложений, такие как метод Р.В. Посевкина [5], сохраняют свою актуальность благодаря адаптированной реализации для программных интерфейсов приложений.

III. МЕТОДЫ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

Основными компонентами процесса преобразования пользовательских запросов в структурированные данные, пригодные для вызовов API, являются синтаксический, морфологический и семантический анализ, которые совместно обеспечивают точность интерпретации запросов.

Синтаксический анализ позволяет выделить структурные связи между элементами запроса и определить его тип. Это помогает системе понять, какие действия пользователь ожидает выполнить, и идентифицировать основные сущности. Например, глаголы действия, такие как "создать", "обновить" или "удалить", сопоставляются с соответствующими методами API.

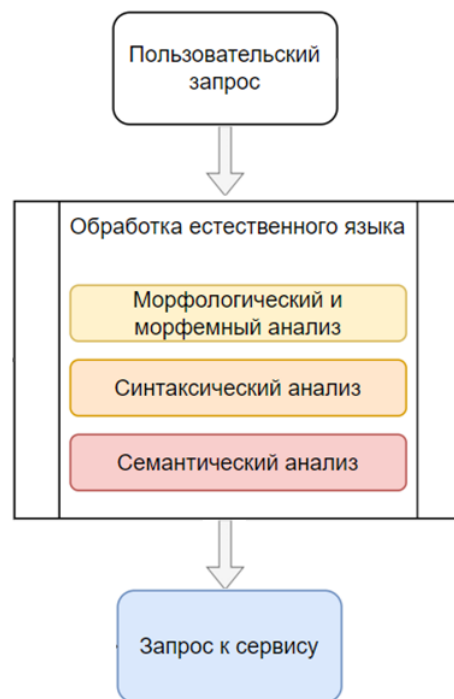


Рисунок 3. Схема обработки естественно-языкового запроса

Морфологический анализ определяет грамматические характеристики слов, такие как часть речи, падеж и склонение, что позволяет уточнить смысловые значения ключевых элементов запроса. Этот этап особенно важен для языков с богатой морфологией, таких как русский.

Формирование семантической модели API требует не только выделения сущностей, но и их сопоставления с методами и параметрами API. Для указанных аспектов предобученные языковые модели, такие как BERT [6], GPT [7], или T5 [8], демонстрируют свою высокую эффективность. Модели способны анализировать контекст, выявлять именованные сущности и определять их связь с методами API. Например, слово "ключ" может интерпретироваться как предмет или параметр доступа к API в зависимости от контекста запроса.

Современные подходы также включают динамическое обновление семантической модели на

основе изменений в структуре API или обратной связи от пользователей, что позволяет минимизировать ручное вмешательство и адаптировать систему к новым условиям. [9]

Рассмотренные методы обработки естественного языка обеспечивают основу для автоматизированного формирования семантической модели API.

IV. АЛГОРИТМ АВТОМАТИЗИРОВАННОГО ФОРМИРОВАНИЯ СЕМАНТИЧЕСКОЙ МОДЕЛИ API

Автоматизация процесса создания семантической модели API представляет собой последовательность шагов, направленных на извлечение информации о методах, анализ параметров и построение структурированных связей между запросами пользователя и вызовами API. Предложенный алгоритм интегрирует современные методы NLP для минимизации ручного вмешательства и повышения адаптивности системы к изменениям структуры программных интерфейсов приложений.

При рассмотрении алгоритма стоит обратить внимание на первый этап, в котором пользователь предоставляет данные: тезаурусы и справочники терминов. Например, в качестве специализированные тезаурусы подойдут, такие как RuWordNet [10], RuThes [11] и другие. Данные позволяют связать синонимы или альтернативные формулировки с конкретными методами и параметрами API.

На втором этапе извлекаются данные о доступных методах API, их параметрах, типах данных и возвращаемых значениях. Источниками информации могут быть документация API, лог-файлы запросов и примеры использования. Алгоритм обрабатывает текстовые описания методов, выделяя ключевые элементы, такие как названия методов, параметры и ожидаемые форматы данных.

На третьем этапе извлеченная информация используется для выявления взаимосвязей между методами API. К примеру, параметры, выступающие в роли идентификаторов сущностей (такие как ENTITY_ID), могут быть связаны между различными методами для создания единой структуры взаимодействия. Связи моделируются по аналогии с отношениями между таблицами в реляционных базах данных, где внешние ключи играют ключевую роль.

На четвертом этапе анализируется семантическое содержание названий методов и параметров, что позволяет идентифицировать назначение и уточнить тип данных. Например, название метода `getUserById` указывает на получение информации о пользователе, а параметр `userId` идентифицируется как уникальный идентификатор.

На предпоследнем этапе семантическая модель дополняется знаниями о значении данных и их связи с запросами пользователя. Для этого используются предобученные языковые модели, такие как BERT или GPT, которые выделяют сущности и действия в пользовательских запросах. Запрос "добавить пользователя в чат" автоматически интерпретируется

как вызов метода `chat.user.add` с соответствующими параметрами.

На финальном этапе пользователь может дополнять семантическую модель специфичными терминами и связями, которые не были учтены автоматически.

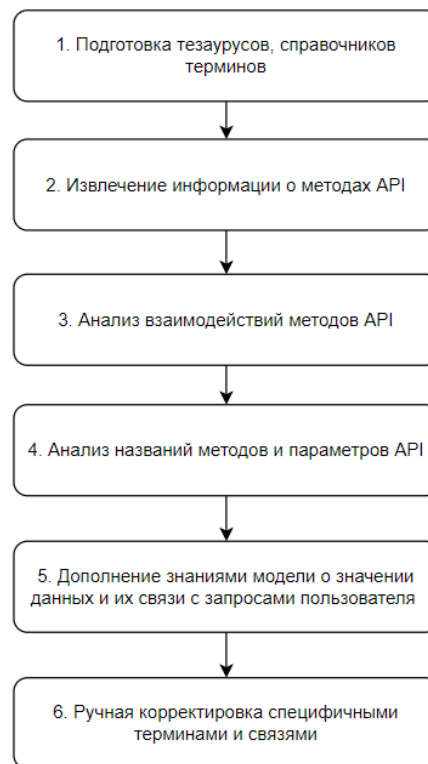


Рисунок 4. Алгоритм автоматизированного формирования семантической модели API

Особенностью предложенного алгоритма является возможность его использования не только для начального формирования семантической модели, но и для её последующего обновления. В случае изменения структуры API или добавления новых методов система автоматически адаптируется, обновляя соответствующие связи и правила. Например, при добавлении нового метода система может извлечь информацию из документации и сопоставить её с существующей моделью, минимизируя время и усилия на адаптацию.

Кроме того, использование языковых моделей обеспечивает высокую точность интерпретации пользовательских запросов, учитывая их контекст и возможные вариации формулировок.

V. ЭТАПЫ ФОРМИРОВАНИЯ СЕМАНТИЧЕСКОЙ МОДЕЛИ API

Каждый этап формирования семантической модели API состоит из последовательности этапов, каждый из которых направлен на детальное описание структуры API, установление связей между методами и параметрами, а также построение модели, способной адаптироваться к изменениям. Этапы основаны на алгоритме автоматизации, представленном ранее, и включают применение методов обработки

естественного языка для интеграции пользовательских запросов с вызовами API.

Даталогическая модель описывает API на уровне данных, включая методы, параметры и их взаимосвязи. На этапе формирования модели извлекаются методы API из документации, логов или примеров использования. Каждый метод анализируется для выявления его функций, таких как создание, обновление или удаление сущностей. Затем идентифицируются параметры, их типы данных. Например, параметр `userId` может быть определён как идентификатор пользователя, представленный целым числом. После этого анализируются взаимосвязи между методами на основе их параметров. К примеру, параметр `ENTITY_ID` метода `chat.get` может использоваться как внешний ключ для метода `chat.user.add`, что позволяет определить последовательность вызовов.

После предыдущего этапа наступает определение типов данных и правил валидации. Например, в зависимости от конкретной области, можно выделить следующие типы данных:

- 1) Идентификаторы (например, `UUID` (от англ. `Universally Unique Identifier`) или идентификаторы пользователей);
- 2) Географические координаты (широта и долгота);
- 3) Финансовые данные (счета, суммы транзакций);
- 4) Даты и временные интервалы.

На следующем этапе формируется инфологическая модель, которая описывает логику взаимодействия с API на уровне понятий и терминов: сопоставляются пользовательские запросы с методами API, анализируются логи работы API и пользовательские данные для уточнения терминологии.

В случае добавления новых методов или изменения структуры API система автоматически обновляет связи и термины, используя алгоритмы извлечения данных и предобученные языковые модели.

Для упрощения работы с семантической моделью создаётся таблица проекций, которая связывает терминологию пользователя с методами и параметрами API, как указано в таблице ниже.

Таблица 1. Таблица проекций на примере соответствия пользовательского запроса, метода и параметров API

Пользовательский запрос	Метод	Параметры
"Создать нового пользователя"	<code>user.create</code>	<code>name, email</code>
"Добавить пользователя в чат"	<code>chat.user.add</code>	<code>CHAT_ID, USER_ID</code>
"Удалить пользователя из чата"	<code>chat.user.remove</code>	<code>CHAT_ID, USER_ID</code>
"Получить список активных пользователей"	<code>user.list</code>	<code>status=active</code>
"Загрузить файл в облако"	<code>cloud.upload</code>	<code>file, folder_id</code>

"Отправить уведомление пользователю"	<code>notification.send</code>	<code>USER_ID, message</code>
--------------------------------------	--------------------------------	-------------------------------

Завершающим этапом является проверка модели на корректность и интеграция её в рабочую среду, что включает в себя тестирование на различных сценариях, проверку обработки сложных и неоднозначных запросов, а также анализ точности сопоставления запросов с методами API.

VI. РЕЗУЛЬТАТЫ

Разработанный алгоритм автоматизированного формирования семантической модели API и этапы её реализации были протестированы в специально созданной тестовой среде, предназначенной для анализа эффективности преобразования естественно-языковых запросов в корректные API-вызовы. Эксперимент включал оценку точности, полноты и адаптивности предложенного подхода в условиях, приближенных к реальным.

В эксперименте приняли участие 70 человек, которым было предложено формулировать запросы к API в свободной форме. Для оценки качества работы системы использовались 92 запроса, включающие как типовые, так и сложные, неоднозначные формулировки.

Результаты эксперимента показывают высокую эффективность предложенного подхода. Из общего числа сгенерированных запросов $|D_{retr}| = 89$ случаев были синтаксически корректными, релевантными оказались $|D_{rel} \cap D_{retr}| = 82$ случая, что демонстрирует способность модели точно интерпретировать запросы и сопоставлять их с методами API. В результате доля корректно определённых результатов среди всех предсказанных системой составила 0.92.

$$Pr = \frac{|D_{rel} \cup D_{retr}|}{|D_{retr}|} \quad (1)$$

Доля корректно определённых результатов среди всех действительно релевантных составила 0.89.

$$Re = \frac{|D_{rel} \cup D_{retr}|}{|D_{rel}|} \quad (2)$$

Гармоническое среднее между точностью и полнотой, отражающее баланс между ними составило 0.90.

$$F = \frac{2 * Pr * Re}{Pr + Re} \quad (3)$$

При анализе некорректных случаев в результате неоднозначных формулировок запросов или отсутствия в тезаурусе специфических терминов, используемых в узкоспециализированных предметных областях, были расширены тезаурусы и обновлены динамически модели с учётом обратной связи от пользователей и анализа новых запросов.

VII. ЗАКЛЮЧЕНИЕ

В данной статье рассмотрен процесс автоматизации формирования семантической модели программных интерфейсов приложений, обеспечивающей преобразование естественно-языковых запросов пользователей в корректные вызовы методов. Проведён анализ существующих подходов, выделены их преимущества и недостатки, а также предложен алгоритм, включающий этапы анализа методов и параметров интерфейсов, типизации данных, построения логических связей и динамического обновления модели.

Реализованная тестовая среда продемонстрировала высокую точность и полноту обработки запросов, минимизацию ручного вмешательства и успешную адаптацию модели к изменениям структуры программных интерфейсов. Полученные результаты подтверждают перспективность использования предложенного подхода для упрощения взаимодействия пользователей с информационными системами.

БИБЛИОГРАФИЯ

- [1] Tirupati K., Joshi A., Singh C, Akshun J., Shalu G. Leveraging Power BI for Enhanced Data Visualization and Business Intelligence // Universal Research Reports. 2024, P. 676-711.
- [2] Poulter. C. Voice recognition software – Nuance Dragon naturally speaking // Occupational Medicine. Volume 70, Issue 1, 2020, P. 75-76.
- [3] Su Y., Awadallah A., Khabsa M., Pantel P., Gamon M., Encarnacion M. Building Natural Language Interfaces to Web APIs. // In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 2017, P. 177-186.
- [4] Hosseini S., Awadallah A., Su Y. Compositional generalization for natural language interfaces to web apis // arXiv, 2021.
- [5] Посевкин Р.В., Бессмертный И.А. Естественно-языковой пользовательский интерфейс диалоговой системы // Программные продукты и системы. № 3, 2016, С. 5-9.
- [6] Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, P. 4171-4186.
- [7] Radford A., Narasimhan K., Salimans T., Sutskever I. Improving language understanding with unsupervised learning: tech. rep., 2018.
- [8] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I. Attention is All you Need // Advances in Neural Information Processing Systems 3. Curran Associates, Inc., 2017, P. 5998-6008.
- [9] Bourhis P., Reutter J., Vrgoč D. JSON: Data model and query languages // Information Systems, Volume 89, 2020.
- [10] Ерофеева И., Соловьев В., Байрашева В. Психосемантический эксперимент как инструмент объективации данных о синонимии (на материале русского языка) // Вестник ВолГУ. Серия 2: Языкознание. 2020.
- [11] Kirillovich, A., Nevzorova, O., Gimadiev, E., Loukachevitch, N. RuThes Cloud: Towards a Multilevel Linguistic Linked Open Data Resource for Russian. // Knowledge Engineering and Semantic Web (KESW). Springer, vol 786, 2017.

Статья получена 11 декабря 2024.

Игорь Владимирович Евченко, Московский инженерно-физический институт (национальный исследовательский ядерный университет), (e-mail: t.foreli@ya.ru).

Automating the development of semantic models for application programming interfaces

Igor V. Evchenko

Abstract — This article analyzes the problem of automating the creation of semantic models for application programming interfaces, which provides the transformation of user queries in natural language into accurate API method calls. Existing approaches to developing natural language user interfaces are reviewed, including the use of natural language processing techniques, pre-trained language models, and specialized thesauruses. The paper proposes an algorithm for the automated generation of semantic models for application programming interfaces. A test environment was implemented to evaluate the proposed approach, providing dynamic mapping of user queries to API methods. Based on the results, it can be concluded that the proposed method improves the accuracy and completeness of query transformations, minimizes manual intervention, and ensures the model's adaptability to changes in API structures.

Keywords — Natural language processing, semantic model, API.

REFERENCES

- [1] Tirupati K., Joshi A., Singh C, Akshun J., Shalu G. Leveraging Power BI for Enhanced Data Visualization and Business Intelligence // Universal Research Reports. 2024, P. 676-711.
- [2] Poulter. C. Voice recognition software – Nuance Dragon naturally speaking // Occupational Medicine. Volume 70, Issue 1, 2020, P. 75-76.
- [3] Su Y., Awadallah A., Khabisa M., Pantel P., Gamon M., Encarnacion M. Building Natural Language Interfaces to Web APIs. // In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 2017, P. 177-186.
- [4] Hosseini S., Awadallah A., Su Y. Compositional generalization for natural language interfaces to web apis // arXiv, 2021.
- [5] Posevkin R.V., Bessmertny I.A. Natural-language user interface of the dialog system // Software Products and Systems. № 3, 2016, C. 5-9.
- [6] Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, P. 4171-4186.
- [7] Radford A., Narasimhan K., Salimans T., Sutskever I. Improving language understanding with unsupervised learning: tech. rep., 2018.
- [8] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I. Attention is All you Need // Advances in Neural Information Processing Systems 3. Curran Associates, Inc., 2017, P. 5998-6008.
- [9] Bourhis P., Reutter J., Vrgoč D. JSON: Data model and query languages // Information Systems, Volume 89, 2020.
- [10] Erofeeva I., Solov'ev V., Bayrasheva V. Psychosemantic experiment as a tool for objectifying data on synonymy (on the material of the Russian language) // Vestnik VolSU. Series 2: Linguistics. 2020.
- [11] Kirillovich, A., Nevzorova, O., Gimadiev, E., Loukachevitch, N. RuThes Cloud: Towards a Multilevel Linguistic Linked Open Data Resource for Russian. // Knowledge Engineering and Semantic Web (KESW). Springer, vol 786, 2017.