TU фреймворк в автоматической обработке входящих заявок

Александр Тощев

Казанский (Приволжский) Федеральный Университет, Институт Механики и Математике имени Чеботарева, [atoschev@kpfu.ru](mailto:atoschev@kpfu.ru)

Максим Таланов

Казанский (Приволжский) Федеральный Университет, Институт Высшей школы ИТИС, [max.talanov@kpfu.ru](mailto:max.talanov@kpfu.ru)

*Введение*

Разработанный фреймворк предоставляет возможность обработки входящих сообщений на естественном языке, область его применения достаточно широка. От интернет помощников до интеграции в системы автоматической обработки инцидентов.

Модель базируется на тезисе о возможности понимания входящего запроса, а не слепой обработки. Понимание базируется на мышление. В теоритическую основу фреймворка легла модель мышления Марвина Мински[1]. В данной модели мышление делится на 3 категории:

1. Критик
2. Селектор
3. Путь мышления

*Критик*

Критик является по сути своей вероятностным предикатом. В упрощенным виде можно назвать их триггерами. После любого внешнего воздействия активируется несколько критиков. Они проверяют с помощью своего предиката сработали они или нет. После активации критик возвращает селектор. Например, с точки зрения входящих заявок может быть использован критик «Прямые инструкции». Примеры критиков:

1. Критик Выученных реакций
2. Рефлексивный критик
3. Само-рефлексивный критик

*Селектор*

Селектор возвращает ресурсы необходимые для обработки запросов. В узком смысле Селектор может вернуть либо другой критик, либо другой селектор.

*Путь мышления*

Путь мышления — это путь при помощи которого будет решена проблема. В фреймворке TU есть несколько типов путей мышления:

* Общая логика
* Специализированный путь мышления

Примерами специализированных путей мышления могут служить следующие пути мышления:

1. Система знает, как решить проблему, используя уже накопленные знания
2. Адаптация существующего решения (если проблема подобна)
3. Реформуляция, с помощью которой вырабатываются новый путь мышления, после решения проблемы, используя путь 2
4. Корреляция, с помощью который ведется поиск уже существующих знаний по входящим.
5. Помощь, путь при помощи которого система обращается к человеку за помощью

Архитектура фреймворка предполагает возможность использования нескольких путей параллельно, что подразумевает быструю обработку запросов. Таким образом можно избежать зависание системы в одном пути решения и гарантировать поиск наиболее оптимального решения.

Пример 1. Если проблема сгенерирована автоматически, система должна обработать ее согласна инструкциям из книги А.

Пример 2. Если описание проблемы уже есть в Базе Знаний, то можно использовать путь Адаптации (2), чтобы решить данную проблему.

С точки зрения программной архитектуры путь мышления проводит работу над данными в текущем контексте.

*Уровни мышления*

Мински предложил 6-ти уровневую модель. Каждый уровень мышления имеет свою функцию. Каждый следующий уровень имеет более сложную функцию.

1. Инстинктивный
2. Обученный
3. Уровень размышлений
4. Рефлексивный
5. Само-рефлексивный
6. Само сознательный

Первый уровень включает врожденные инстинкты, высшей уровень включает идеалы и персональные цели.

*Реализация прототипа*

В прототипе были реализованы следующие критики, селекторы и Пути Мышления:

1. Обработка естественного языка
2. Классификатор проблем
3. Симуляция
4. Реформуляция
5. Корреляция
6. Поиск решения

В прототипе были реализованы следующие пути Мышления:

1. Обученный
2. Уровень размышлений
3. Рефлексивный
4. Само рефлексивный
5. Само сознательный

Инстинктивный уровень планируется использовать в будущем для обработки автоматически сгенерированных инцидентов.

*Основной цикл мышления*

Основной идей решения является быстрое переключение система от одного метода решения к другому методы.

Вот почему архитектура приложения организована следующем образом. Когда система получает внешний запрос, иными словами «сталкивается с проблемой» она выбирает необходимые ей знания о типе проблемы и подходящий Путь мышления. Кроме того, система может обрабатывать проблемы, связанные с ее внутренними активностями. Текущая архитектура включает рефлексивные уровни между всеми уровнями системы:

* Рефлексивный-суждение о текущей задачей: дошли ли мы до промежуточный цели или же приблизились к основной
* Само рефлексивный – суждение о текущей задаче в рамках общего состояния текущей задачи: ресурсов, времени и т.д.
* Само –сознательный – суждение о текущей задаче в рамках глобальных состояний системы: ее общего состояния и т.д.

Примерный сценарий работы может быть таким:

1. Началась обработка входящий заявки
2. Активируются подходящие критики и возвращает селектор
3. Селектор возвращает необходимый Путь Мышления
4. Путь мышления модифицирует текущий контекст краткосрочной памяти
5. Процесс повторяется, пока не будет достигнута цель (найдено решение)

Пути мышления активируются параллельно, подобно человеческому специалисту.

*Краткосрочная и долгосрочная память*

Пути Мышления работают с краткосрочной памятью и модифицируют данные в ней. После успешной обработки память переписывается в долгосрочную и сохраняется уже на жестком диске. Иными словами, краткосрочная память хранится в оперативной памяти, долгосрочная же хранится на постоянно запоминающем устройстве.

*Доменная модель*

Доменная модель описывает знания текущие знания и содержит: решения, проблемы, пути мышления, критики, селекторы, рецепты решения и т.д. Таким образом критики, селекторы, пути мышления также являются «приобретенными знаниями».

*Описание уровней мышления, реализованных в прототипе*

Обученный

Содержит Менеджер процессов. Этот менеджер активирует несколько Способов Мышления, чтобы совершить предварительную обработку входящей информации. Цель данного критика создать семантическую сеть инцидента. Он активирует следующие пути мышления:

* Коррекция орфографии
* Поиск синонимов (поиск синонимов слова)
* Аннотация (по словам и из синонимам ищутся концепции внутри Базы Знаний)

nding existing concepts in Knowledge Base

Мышление

На данном уровне происходит классификация проблемы, согласно существующем категориям. Например,

* Прямые инструкции – содержит прямые инструкции к действиям
* Проблема с желаемым состоянием – содержит описание текущий ситуации и желание пользователя как должно быть

Рефлексивный

На данном уровне система выставляет цели и следит за контролем времени исполнения задачи. **Менеджер целей**. Осуществляет постановку целей. Основная цель «Помочь пользователю»

* Помочь пользователю
  + Решить инцидент
  + Понять тип инцидента
  + Смоделировать инструкции действий

Само рефлексиный

На данном уровне система контролирует следующие действия: инициализирует краткосрочную память, включает таймер, ведет коммуникацию с пользователям по средствам менеджера коммуникаций.

**Менеджер Коммуникаций.** Используется, если системы столкнулась с проблемой, которую не может решить без пользователя.

Само сознательный

Это вершина уровней. На нем выставляется общий эмоциональный статус системы.

*Тренинг*

На начальном этапе системы содержит базовые концепции:

* Объект
* Действие
* Форма вежливости – тип объектов необходим для корректирования семантического разбора.

*Выводы*

Основной задачей работы было создание фреймворка (системы) для демонстрации приложения модели Мышления Марвина Мински. По результатам удалось достичь 61% успешно обработанных входящих задач.

Список литературы

1. Liu H., Lieberman H.: Metafor: Visualizing Stories as Code. Cambridge, MIT Media Laboratory (2005).
2. Russel S., Norvig P.: Articial Intelligence. A Modern approach. Pearson (2010).
3. Minsky M.: The Emotion Machine. Simon & Schuster Paperbacks (2006)