

## **ПРОГРАММА СНАТВОТ — ЧАТ-БОТ ИЛИ ВИРТУАЛЬНЫЙ СОБЕСЕДНИК**

**В.А. Шовин**

научный сотрудник, e-mail: v.shovin@mail.ru

ФГБУН Институт математики им. С.Л. Соболева СО РАН, Омский филиал

**Аннотация.** Разработана программа виртуального собеседника или чат-бота на базе внешнего API, алгоритма поиска ответов в базе знаний расширенной разметки AIML, а также рекуррентной нейронной сети. Алгоритм позволяет находить ответы к вопросам релевантные вопросам из базы знаний. Процедура сортировки релевантных ответов включает в себя поиск по регулярному выражению, поиск по тематике, поиск по истории и поиск по максимальному совпадению слов в вопросах. Для повышения качества поиска релевантных ответов в интерпретатор внедрён модуль морфологического анализатора отдельных слов. Рекуррентная нейронная сеть задана на множестве слов всех вопросов и ответов базы знаний.

**Ключевые слова:** чат-бот, виртуальный собеседник, AIML, рекуррентная нейронная сеть.

### **Введение**

На сегодняшний день остаётся актуальным создание программ, имитирующих общение человека. Простейшей моделью общения является база вопросов и ответов к ним [1]. В данном случае возникает проблема описания базы знаний и реализация программы-интерпретатора. Язык разметки базы знаний может включать в себя паттерны вопросов и соответствующие им шаблоны ответов, также предысторию диалогов к ним и название соответствующей темы общения.

Чат-бот может выполнять дополнительные функции, например, такие как поиск музыки, картинок, фактов; также отражать калькулятор, прогноз погоды, вывод курса валют. Большинство таких функций имеют реализацию в интернете и доступны в качестве внешнего API.

Альтернативным вариантом создания программы виртуального собеседника является использование алгоритмов машинного обучения на базе диалогов общения, а именно искусственные нейронные сети. Подходящей моделью ИНС является рекуррентная нейронная сеть, способная хранить, обобщать и прогнозировать различные последовательности. В данной работе в качестве элементов последовательности предлагается использовать индексы, соответствующие словам в базе знаний вопросов и ответов.

## 1. AIML

Одним из форматов разметки базы знаний является стандарт языка разметки AIML (Artificial Intelligence Markup Language). Ключевыми словами в языке являются `category`, `pattern` и `template`:

```
<aiml>
  <category>
    <pattern>Привет! Как дела?</pattern>
    <template>
      <random>
        <li>Привет. Нормально.</li>
        <li>Привет. Отлично. Как у тебя?</li>
      </random>
    </template>
  </category>
</aiml>
```

Тег `category` является родительским к тегам `pattern` и `template`, хранящим шаблоны вопроса и ответов. Тег `random` позволяет указать несколько ответов к вопросу, выбираемых интерпретатором случайным образом. В работе предлагается ввести дополнительные теги, соответствующие истории и теме разговора.

```
<aiml>
  <category>
    <pattern>Чем занимаешься?</pattern>
    <pattern>Что делаешь?</pattern>
    <template>
      <random>
        <li>Читаю книгу.</li>
        <li>Смотрю телевизор.</li>
      </random>
    </template>
    <history>Привет! Как дела? Привет. Нормально.</history>
    <theme>Дела</theme>
  </category>
</aiml>
```

Несколько тегов `pattern` позволяют описать различные варианты вопросов, соответствующие данной категории, на которые должны последовать одни и те же варианты ответов. Тег `history` хранит историю диалога, предшествующую данному вопросу. Тег `theme` хранит название темы разговора. Данные теги позволяют интерпретатору подобрать паттерн вопроса, соответствующий предыстории диалога и теме общения, что должно сказаться на улучшении качества имитации общения посредством чат-бота.

Интерпретатор языка разметки должен позволять находить наиболее релевантные вопросы следующими дополняющими друг друга способами:

1. Поиск по всей фразе вопроса на основе регулярного выражения.
2. Поиск по количеству совпадающих слов в вопросе и паттернах.
3. Поиск по совпадению текущей темы и тем категорий.
4. Поиск по количеству совпадающих слов в текущей предыстории разговора и предысториях категорий.

Соответствующий алгоритм выбора лучшего совпадения формируется на базе алгоритма сортировки:

```
sortMatches = allMatches.sort(function(a, b) {
    if(a.pattern == inputText && b.pattern != inputText)
        return -1;
    if(b.pattern == inputText && a.pattern != inputText)
        return 1;
    if(a.matches < b.matches)
        return 1;
    if(a.matches > b.matches)
        return -1;
    if(a.theme == bot.theme && b.theme != bot.theme)
        return -1;
    if(b.theme == bot.theme && a.theme != bot.theme)
        return 1;
    if(a.historyMatches < b.historyMatches)
        return 1;
    if(a.historyMatches > b.historyMatches)
        return -1;
    return 0;
})
```

В результате выбирается первый элемент отсортированного массива совпадений.

Для повышения качества поиска релевантных ответов в процедуру сравнения слов был внедрён модуль морфологического анализатора, позволяющий находить базовые формы слов. Тем самым сравнение происходит по базовым формам слов, что исключает несоответствия слов, связанные с их склонениями.

## 2. Алгоритм классификации тем

Предлагается использовать алгоритм определения названий тем категорий, когда определена лишь их часть.

1. Пусть множество категорий, в которых определены темы, —  $T$ . Множество категорий, в которых темы не определены, —  $D$ . Элементам данных множеств ставится в соответствие объединение строк всех значений паттернов и шаблонов категории.

2. Вводится порог детерминации, например,  $p = 70\%$ .
3. Последовательно перебираются элементы множества  $D$ . Выбирается подмножество  $V$  элементов множества  $T$ , в котором  $p$  — процент уникальных слов — определён.
4. Если  $V$  пусто, то тема определяется строкой элемента множества  $D$ .
5. Если  $V$  не пусто, то выбирается элемент из  $V$ , для которого отношение количества совпадающих уникальных слов к общему количеству уникальных слов в строке этого элемента максимально. Соответствующая тема данного элемента  $V$  определяется как тема элемента  $D$ .

### 3. Рекуррентная нейронная сеть

Рекуррентная нейронная сеть — вид многослойного перцептрона, у которого сигналы с нейронов выходного слоя поступают на дополнительные нейроны входного слоя, т.н. нейроны контекста.

Входной вектор сигнала поступает на группу нейронов INPUT, на группе нейронов CONTEXT нулевой сигнал. Далее сигнал распространяется в группу нейронов скрытого слоя HIDDEN, а затем преобразуется ими и попадает на нейроны выходного слоя OUTPUT. На следующей итерации вместе с вектором сигнала INPUT на контекстную группу нейронов поступают копии сигналов с выходного слоя OUTPUT прошлой итерации (рис. 1).

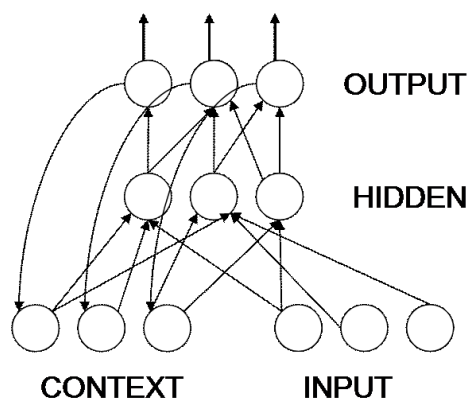


Рис. 1. Общий вид структуры рекуррентной нейронной сети.

Структура рекуррентной нейронной сети для запоминания предложений имеет следующий вид:

Слои CONTEXT, INPUT и OUTPUT имеют по одному нейрону, значение сигнала на выходе которого ставится в соответствие индексу слова в наборе слов. Дополнительно вводится слово `__end__`, соответствующее концу предложения [2]. Сеть последовательно обучается предложениям вида:

«Привет. Как дела? `__end__` Привет. Нормально. `__end__`».

Получение ответов на вопросы рекуррентной нейронной сетью происходит по следующей схеме (рис. 2)

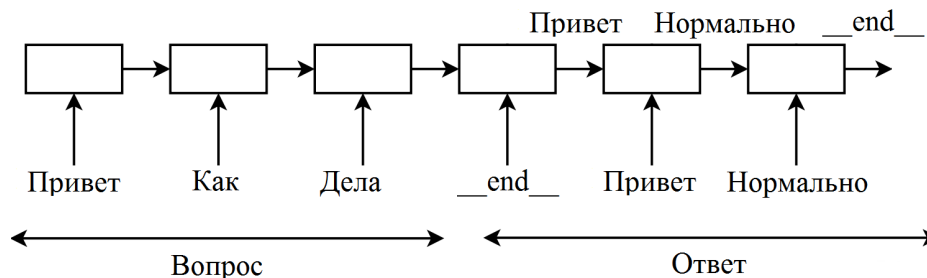


Рис. 2. Получение ответа на вопрос рекуррентной нейронной сетью.

Объём слоёв HIDDEN должен позволять запоминать весь набор предложений. Сеть обучается методом обратного распространения ошибки.

#### 4. Программная реализация

Программа чат-бота была реализована в качестве Android-приложения с возрастным ограничением 18+. В приложении доступно несколько режимов чат-бота:

- 1) на базе внешнего API [www.pandorabots.com](http://www.pandorabots.com);
- 2) на базе оригинального обобщения языка AIML;
- 3) на базе рекуррентной нейронной сети.

В качестве морфологического анализатора была использована свободная JavaScript-библиотека для обработки текстов на русском языке Az.js.

Для создания и обучения рекуррентной нейронной сети использовалась свободная JavaScript-библиотека RecurrentJS.

Во всех режимах доступен автоперевод ответов на язык вопросов пользователя на базе сервисов Yandex Translate API и Bing Translate API. Также в качестве внешних сервисов используется сервис поиска картинок Custom Search API и Bing Image Search API. Поиск знаний реализован на базе Google Knowledge Graph Search API. Поиск музыки реализован на базе SoundCloud API. Калькулятор, прогноз погоды, курс валют, время реализованы на базе Wolfram|Alpha API.

Приложение доступно по адресу:

<https://play.google.com/store/apps/details?id=svlab.chatbot>

#### 5. Заключение

Программно реализованы режимы чат-бота на базе внешнего API, расширенной разметки AIML и рекуррентной нейронной сети. Расширение разметки AIML включает в себя новые теги theme и history для более эффективного

поиска релевантных вопросов и ответов. В интерпретатор внедрён модуль морфологического анализатора отдельных слов и приведения их в базовую форму в процедуре поиска релевантных ответов. Рекуррентная нейронная сеть позволяет получать ответы на вопросы, которых не было в базе знаний, с помощью способности сети к обобщению. Приложение ChatBot доступно для платформы Android в Play Маркете.

## ЛИТЕРАТУРА

1. Проватар А.И., Клочко К.А. Особенности и проблемы виртуального общения с помощью чат-ботов // Научные труды Винницкого национального технического университета. 2013. № 3. С. 2.
2. Vinyals O. Quoc Le A Neural Conversational Model // arXiv preprint arXiv:1506.05869, 22 Jul 2015.

## CHATBOT PROGRAM — CHATBOT OR VIRTUAL COMPANION

**V.A. Shovin**

Scientist Researcher, e-mail: v.shovin@mail.ru

Omsk Branch of the Institution of the Russian Academy of Sciences Institute of Mathematics. S. Siberian Branch of RAS

**Abstract.** The program of the virtual person or chatbot based on an external API, algorithm of search of responses in the knowledge base of expanded markup AIML, as well as the recurrent neural network are developed. The algorithm allows us to find answers to relevant questions from the knowledge base. The procedure of relevant responses sorting includes regular expression search, search by category, search by history and search for the best match of words in questions. The recurrent neural network is defined on the set of words of Q & A knowledge base

**Keywords:** chatbot, virtual companion, AIML, recurrent neural network.

*Дата поступления в редакцию: 28.06.2016*