

## РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ ДЛЯ ПОИСКА АССОЦИАЦИЙ ВНУТРИ ЯЗЫКОВОЙ МОДЕЛИ

О. А. Голуб, П. А. Хаустов  
Томский политехнический университет  
taube03@yandex.ru

### Введение

«Кодовые имена» — настольная игра, придуманная Владей Хватилом и опубликованная в 2015 году издательством «Czech Games Edition». В России локализована издательством «GaGa Games» [1].

На сегодняшний день реализованы приложения для игры в «Кодовые имена» на мобильных устройствах. Однако их недостатком является невозможность играть в оригинальные правила количеством меньше четырёх человек. Возможным решением проблемы является написание искусственного интеллекта для игры.

Таким образом, целью работы является применение алгоритмов поиска ассоциаций внутри языковой модели к разработке игрового приложения «Кодовые имена» на платформе Android.

### Исследования в области ассоциаций

Словесная ассоциация — это закономерно возникающая связь между отдельными понятиями, отражённая в сознании индивида в виде связи между словами, обозначающими эти понятия, и закреплённая в его памяти. Существуют алгоритмы, пригодные для машинного анализа текстов и выявления в них ассоциаций. Эти алгоритмы зависят от вида искомых ассоциаций [2]. Различают два вида словесных ассоциаций [3]:

Синтагматические ассоциации — это пары слов, часто разной части речи, которые часто встречаются в предложении вместе. Например, в предложении «Мальчик идёт в школу» можно выделить такие пары синтагматических ассоциаций как «мальчик» и «школа», а также «идти» и «школа».

Парадигматические ассоциации — это пары слов одной части речи, которые являются антонимами или синонимами и обозначают схожие понятия. Например, можно выделить такие парадигматические ассоциации как «идти» и «бежать», которые являются близкими по значению понятиями, или же «мальчик» и «девочка», которые можно назвать антонимами.

В зависимости от типа ассоциаций различаются алгоритмы их обнаружения в тексте [4].

Для нахождения синтагматических ассоциаций тексты рассматриваются с помощью метода скользящего окна. Пары слов, находящиеся в одном окне, увеличивают свою степень ассоциативности.

После нахождения синтагматических ассоциаций каждому слову сопоставляется вектор его ассоциаций. Затем с помощью сравнения векторов находятся пары слов, связанные парадигматическими ассоциациями.

### Проектирование приложения

Для реализации приложения была выбрана клиент-серверная архитектура. Серверу назначались следующие функции:

- хранение базы данных слов и ассоциаций;
- генерация начального состояния поля;
- принятие решения о ходе игрока-бота;
- принятие решения о ходе капитана-бота.

На сервере была размещена база данных, содержащая слова и ассоциации (рисунок 1).

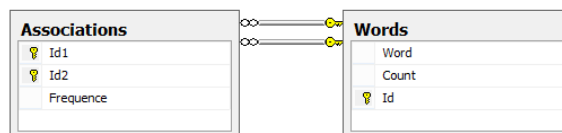


Рис. 1. Схема базы данных

Для создания базы ассоциаций был выбран британский национальный корпус. Британский национальный корпус — это корпус текстов из ста миллионов слов, содержащий образцы письменного и разговорного британского английского языка из широкого круга источников [5].

### Алгоритм создания базы данных

Алгоритм можно разбить на несколько этапов:

1. Исключение из рассмотрения служебных частей речи.
2. Лемматизация слов.
3. Подсчёт статистики отдельных слов и ассоциаций.
4. Фильтрация слов по частоте.
5. Выделение синтагматических ассоциаций.

В сравнении с эталонным алгоритмом [4] была изменена метрика схожести двух слов. Новая метрика пропорциональна частоте встречаемости данной пары слов в тексте и обратно пропорциональна частоте встречаемости данных двух слов по отдельности. Это позволяет использовать получившийся коэффициент как вероятность применения данной ассоциации игроком.

6. Выделение парадигматических ассоциаций. В сравнении с эталонным алгоритмом было увеличено быстродействие. Асимптотика исходного алгоритма составляла  $O(n^3 * \log_2 n)$ , где

$n$  – количество выбранных слов. Для увеличения быстродействия перебирались не все пары слов и все коэффициенты, а только пары слов, имеющие совпадающие ассоциации. Также пропускались коэффициенты, равные нулю. Асимптотику получившегося алгоритма можно оценить как  $O(n^3)$ , однако в реальности можно использовать оценку  $O(n^2)$ .

### Проектирование серверной части

Серверная часть реализует три запроса.

1. Генерация начального состояния поля. Все характеристики поля генерируются случайным образом.

2. Принятие решения о ходе игрока-бота. Бот выбирает из доступных слов на поле наиболее подходящее под данную ассоциацию.

3. Принятие решения о ходе капитана-бота. В процессе работы бот выбирает слово-ключ с максимальным математическим ожиданием количества набранных за него очков. Алгоритм имеет следующие шаги:

а) Максимальное математическое ожидание количества полученных очков принимается равным нулю.

б) Перебираются все слова на поле. Если слово уже открыто или не принадлежит команде капитана, осуществляется переход к другому слову к шагу б.

в) Перебираются слова-кандидаты на ответ, имеющие наиболее высокую степень ассоциации с текущим словом на поле. Если у текущего слова на поле больше нет слов-ассоциаций, или их степень ассоциации не превышает установленного порога, осуществляется переход к шагу б.

г) Вычисляется оптимальное количество слов для этого слова-кандидата. Вычисляется математическое ожидание количества полученных очков для текущего слова-кандидата. Если оно превышает максимальное математическое ожидание среди ранее найденных, то текущее слово-кандидат и количество очков, которые могут быть получены за него, заносятся в ответ.

д) Возвращается ответ.

Далее необходимо описать алгоритм вычисления математического ожидания получаемых очков и количества слов, называемых с текущим словом-кандидатом. Алгоритм основан на идее динамического программирования.

а) Выбираются все неоткрытые слова на доске, имеющие ненулевую степень ассоциации со словом-кандидатом.

б) Для каждого подмножества выбранных слов считаются следующие величины:

- количество слов в подмножестве;
- вероятность выбора игроком этого подмножества слов;
- количество очков, полученное за это подмножество слов;
- вероятность окончания хода игроком при достижении этого подмножества.

3. Для каждого выбранного количества слов подсчитывается математическое ожидание количества набранных очков. Для этого складываются следующие величины:

– математическое ожидание количества набранных очков, если игрок сделал ровно указанное количество ходов;

– математическое ожидание количества набранных слов, если игрок прервал ход заранее.

### Заключение

На сегодняшний день реализована как серверная, так и клиентская часть, приложение успешно работает (рисунок 2). Ассоциации, найденные ботом, осмысленны, встречаются случаи победы команды ботов над командами людей.



Рис. 2. Работа приложения

### Список использованных источников

1. Правила игры Codenames [Электронный ресурс]: Издательство GaGa Games (ООО «ГаГа Трейд»). – Режим доступа: [http://codenames-game.ru/assets/images/resources/Codenames%20russian%20rules%20\(compressed\).pdf](http://codenames-game.ru/assets/images/resources/Codenames%20russian%20rules%20(compressed).pdf), свободный. – Загл. с экрана.
2. Виды ассоциаций [Электронный ресурс]: Сеть словесных ассоциаций. – Режим доступа: <https://wordassociations.net/ru/kinds>, свободный. – Загл. с экрана.
3. Paradigmatic and syntagmatic [Электронный ресурс]: 2016 Encyclopedia.com. – Режим доступа: <http://www.encyclopedia.com/social-sciences-and-law/sociology-and-social-reform/sociology-general-terms-and-concepts-49>, свободный. – Загл. с экрана.
4. Rapp, Reinhard (2002). The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In Proceedings of COLING 2002, Taipei, Taiwan.
5. BNC – The British National Corpus. [Электронный ресурс]. Режим доступа: <http://www.natcorp.ox.ac.uk>, свободный. – Загл. с экрана.