УДК 004.81:004.942

# Сравнение методов машинного обучения в задаче предсказания интенции участника онлайн дискуссии

Карпов Н.В. к.т.н., Демидовский А.В.

**Национальный исследовательский университет Высшая школа экономики**

**Нижний Новгород**

**Аннотация:** Данная работа посвящена исследованию интенции пользователя в социальной сети. Для этого используется оригинальный набор данных, где собраны диалоги пользователей из социальной сети и каждое сообщение соотнесено с одним из 25 типов интенций. Рассмотрены современные методы машинного обучения, которые позволяют анализировать элементы последовательности и предсказывать следующие. На выбранном наборе данных проведен вычислительный эксперимент по предсказанию следующей интенции пользователя в ходе дискурса. Оценена точность работы каждого алгоритма.

**Ключевые слова:** предсказание последовательности, анализ социальных сетей, машинное обучение, искусственные нейронные сети

## 1. Введение

В настоящее время в связи с большой популярностью различного рода социальных сетей большой интерес представляет исследование поведения пользователей. Возможность предсказать их поведения даёт новые инструменты во многих сферах деятельности, начиная с маркетинговых и заканчивая политическими манипуляциями общественного мнения.

Ряд исследований как в психологической области, так и в области машинного обучения показывает, что количество типов мотивов (или интенций) собеседника конечно и может быть определено с помощью синтаксического или словарного анализа сообщения [6]. Однако, открытыми остаются вопросы существования неких паттернов в последовательностях сообщений и возможности выявить явную зависимость мотивации

собеседника ***Б*** от того, с каким мотивом было написано сообщение собеседника ***А***. Можно ли, используя такие средства анализа данных, как машинное обучение и рекуррентные нейронные сети, выявить данные паттерны и научиться предсказывать интенцию собеседника исходя из предыдущих сообщений?

Основными целями работы являются:

1. Представление графа дискурса в виде набора последовательностей интенций.
2. Исследование современных методов машинного обучения, которые позволяют анализировать элементы последовательности и предсказывать следующие.
3. Применение исследованных алгоритмов для предсказания следующей интенции пользователя в ходе дискурса.
4. Оценка точности работы выбранных алгоритмов на созданном наборе данных.

## 2. Постановка задачи

Для того, чтобы перейти к описанию основных подходов к решению задачи, необходимо ввести несколько определений и сформулировать общую концепцию того, что такое проблема предсказания последовательности.

Итак, пусть есть конечный алфавит [4] $I = \{i_1, i_2, \dots, i_m\}$. Каждый элемент этого алфавита может быть как атомарным символом, так и набором некоторых элементов. Главное требование заключается в неповторяемости элементов внутри алфавита. Иными словами, алфавит – это множество из уникальных элементов.

**Определение 1.** *Последовательность.* Последовательностью называют упорядоченный список из элементов алфавита. Пусть есть некоторый набор элементов $S = \langle s_1, s_2, \dots, s_n \rangle$. $S$ будет считаться последовательностью, если выполняется условие, что $\forall s_i \in I,\ i \in [1, n]$.

Для того, чтобы осуществлять предсказание следующего элемента последовательности требуется большое количество последовательностей. Для удобства обозначим такое множество следующим символом: $D = \langle S_1, S_2, \dots, S_k \rangle$. Таким образом, проблема предсказания последовательности заключается в умении предложить следующий элемент $s_{n+1}$ для заданной последовательности $S = \langle s_1, s_2, \dots, s_n \rangle$.

## 3. Обзор существующих методов

В результате анализа предметной области нами было выделено несколько наиболее часто используемых подходов к решению задачи предсказания последовательности, а именно традиционные подходы: CPT (Compact Prediction Tree) [4], CPT+ [3], PPM (Prediction by Partial Matching) [2], DG (Dependency Graph) [8], AKOM (All-Kth-Order-Model) [10], TDAG (Transition Directed Acyclic Graph) [7], CTW (Context Tree Weighting) [14], PST (Probabilistic Suffix Trees) [11]. Отдельного внимания заслуживают подходы, использующие нейронные сети для решения поставленной задачи: MLP (Multi Layer Perceptron), CNN (Convolutional Neural Network), LSTM (Long Short Term Memory) [15], RNN (Recurrent Neural Network) [12], DTRNN (Discrete-Time RNN) [9] and Multi-task LSTM [13]. Ниже подробнее рассмотрим некоторые из отмеченных выше алгоритмов.

**3.1. CPT** (Compact Prediction Tree) [4] – это метод, который на данный момент считается наиболее точным в сравнении с другими аналогами. Существует также его улучшенная версия, так называемая **CPT+** [3], где предложено несколько оптимизаций, но ключевая идея остается неизменной. Метод строится вокруг работы с тремя специфическими структурами данных: Prediction Tree (PT), Inverted Index (II), Lookup Table (LT). На этапе тренировки Prediction Tree представляет собой дерево, которое формируется из набора последовательностей $D$. У дерева есть корень, который не содержит в себе никакого элемента $s_i$, играет служебную роль для образования дерева. Вторая важная структура – Inverted Index, представляет собой таблицу $II = \|ii_{kl}\|$, где на пересечении $k$-той строки, соответствующей элементу алфавита $i_k$, и $l$-того столбца, соответствующего последовательности $S_l$, находится либо 1 либо 0, что означает наличие/отсутствие этого элемента алфавита в заданной последовательности. Наконец, третьим элементом является Lookup Table, который связывает Prediction Tree и Inverted Index и представляет собой таблицу, которая для каждой последовательности хранит ссылку на соответствующий её последнему элементу лист в Prediction Tree. Формирование этих трех структур данных соответствует стадии обучения, за которой следует стадия предсказания.

Стадия предсказания представляет особый интерес. Пусть $x$ – длина префикса (контекста), а $S$ – некоторая последовательность. Первым шагом находим все

последовательности из PT, которые содержат последние $x$ элементов из $S$ в любом порядке. Далее, осуществляется поиск через II, простым пересечением строк, соответствующих этим $x$ элементам. Обозначим набор таких последовательностей как $Y$. Для каждой $y \in Y$, найдем общий префикс $y$ и $S$. Например, $S = \langle i_2, i_3, i_5, i_9, i_1 \rangle, x = 2, y = \langle i_2, i_9, i_1, i_4, i_2 \rangle$. Отбросим общий префикс $\langle i_2 \rangle$ и в оставшейся подпоследовательности $y$ $\langle i_9, i_1, i_4, i_2 \rangle$ для каждого элемента заведем строку в специальной таблице Count Table или прибавим соответствующий счётчик на один. Как только закончились $y$, выбираем строку с максимальным значением счётчика (*support*) – это и есть наиболее вероятный следующий элемент. Если таких строк несколько, вводится дополнительная метрика confidence:

$$confidence(s_i) = \frac{support(s_i)}{|\{y \mid y \in Y, s_i \in y\}|} \quad (1)$$

Далее в качестве предсказанного элемента выбирается тот, которому соответствует большее значение $confidence$.

CPT как оригинальный подход имеет ряд недостатков, которые частично решаются в оптимизированной версии алгоритма [3]: потенциально большая глубина дерева, отсутствие работы с очищением последовательностей от шума, невозможность предсказать символ, который не присутствует в обучающей выборке и т.д. Кроме того, возникает вопрос о правилах выбора максимальной длины префикса, который пока решается эмпирическим подбором. В [3] предлагаются стратегии по уменьшению PT через введение новых элементов алфавита для наиболее повторяющихся ветвей, объединение веток без ветвления (один потомок у каждого родителя, начиная с корня). Однако, и обновленный алгоритм CPT+ [3], решая проблемы базового алгоритма, вводит дополнительные параметры, такие как: минимальная и максимальная длина подпоследовательности как критерий определения частой подпоследовательности, приемлемая доля шума, минимальное количество обновлений PT и т.д. Подбор значений параметров становится самостоятельной проблемой.

**3.2. PPM** (Prediction by Partial Matching) [2] – подход, ставший традиционным в задаче предсказания последовательности. Основная идея заключается в построении Марковской

модели заданного порядка $k$, что означает использование $k$ элементов в контексте для предсказания следующего элемента. Пусть $k = 2$, а контекст, по которому делаем предсказание имеет вид $\langle \#, a \rangle$, где # – это знак пробела. Для каждого элемента $i_l$ подсчитывается количество раз, когда последовательность $\langle \#, a \rangle$ имела продолжение $\langle \#, a, i_l \rangle$, обозначим это как $c(i_l)$. Тогда вероятность того, что следующим будет символ $i_l$ равняется:

$$p(i_l) = \frac{c(i_l)}{1 + C}, \quad (2)$$

где $C$ – количество раз появления заданного контекста в тренировочной выборке. Однако, важнейшим элементом этой модели является расчет вероятности возникновения символа, который еще не встречался в заданных контекстах. Для него также рассчитывается вероятность (так называемая *escape* вероятность), например так (в оригинальной модели Метод *A*):

$$p(i_l) = \frac{1}{1 + C} \times \frac{1}{|I| - q}, \quad (3)$$

где $q$ – количество символов, которые уже появлялись в данном контексте.

Несмотря на простоту метода, существуют экспериментальные доказательства его конкурентоспособности по отношению к другим существующим методам [2, 3, 8, 10].

**3.3. DG** (Dependency graph) [8] – это подход, который изначально был разработан для решения задачи эффективной предварительной загрузки ресурсов по мере пользования клиентом сети Интернет. Цель заключается в предсказании наиболее вероятного следующего ресурса, который захочет посетить пользователь на основе данных о его предыдущих действиях. Традиционно, такие алгоритмы представляют собой обобщенные решения, которые могут быть использованы и для проблемы предсказания последовательностей. Общий алгоритм строится вокруг создания одноименной структуры – графа зависимостей, таким что между двумя узлами $A$ и $B$ существует ребро при условии существования тренировочной последовательности, в которой $A$ и $B$ встречались друг за другом в контексте окна – *lookahead window*. Соответственно, вес ребра – частота возникновения этого условия по всей тренировочной базе. Важно отметить, что вес ребра не является вероятностью, то есть не существует обязательного

условия на равенство суммы весов исходящих ребер единице. Процесс предсказания следующего элемента последовательности прост и представляет собой выбор того элемента, у которого вес ребра перехода из текущего состояния в следующее максимален.

**3.4. AKOM** (All-Kth-Order-Model) [10] является своеобразным последователем PPM [2] и заключается в том, что предсказание строится не на основе Марковской модели заданного порядка $k$, но на ряде Марковских моделей, вплоть до Марковской модели $k$-того порядка. Иными словами, тренировка алгоритма заключается в построении таких сетей, а выбор предсказания каждый раз осуществляется в Марковской модели такого порядка, которая содержит префикс нужной длины (контекст). В [10] было показано, что данный метод хорошо работает на длинных последовательностях и достаточно точен, если в качестве критерия точности выбирать попадание ожидаемого в тренировочной выборке предсказания в первые 10 предсказаний (top-10), которые делаются данным методом.

**3.5. TDAG** (Transition Directed Acyclic Graph) [7] также как и PPM [2] и AKOM [10] строится вокруг Марковских процессов. Однако, в данном случае, речь идет о построении Марковских деревьев, вместо стохастического конечного автомата. При этом построенное дерево является аппроксимацией Марковской сети, что влияет, во-первых, на качество получаемого результата, а, во-вторых, приводит к наличию как минимум одного обязательного параметра – максимальной глубины результирующего дерева.

**3.6.** В последние годы **искусственные нейронные сети** [9, 12, 13, 15] приобретают заметную популярность и появляются подходы, применяющие этот особый вид алгоритмов для решения, в том числе, и задачи предсказания последовательностей. В силу того, что в AKOM [10] было показано, что предсказание последовательностей хорошо работает на длинных последовательностях, а существующие традиционные алгоритмы не учитывают данную особенность, использование LSTM сетей [5] объективно обладает высоким потенциалом с точки зрения точности предсказания. С точки зрения структуры, состояние LSTM сети $C_t$ может быть получено через использование так называемого «вентиля» выхода $o_t$. Изменение состояния происходит

через $i_t$, а очистка предыдущего состояния $C_{t-1}$ (в терминах человеческой памяти – «забывание») – через $f_t$. Такая структура несколько сложнее базовой структуры рекуррентных нейронных сетей, однако, позволяет учитывать долгосрочные взаимосвязи. С точки зрения точности производимых результатов, согласно [15], результаты LSTM сетей оказываются лучше, чем у конкурентов – CNN (конволюционные сети) и MLP (многослойный перцептрон), проигрывая по избранным тренировочным выборкам DG [8].

Таким образом, можно заметить, что существует значительное многообразие методов, основывающихся либо на построении хорошо зарекомендовавших и изученных Марковских цепей или деревьев либо на быстрых и эффективных структурах данных, либо на основе нейронных сетей, преимущественно рекуррентных. Далее рассмотрим подробнее предлагаемый подход и сравним полученные результаты с теми, что получаются имеющимися алгоритмами, описанными выше.

## 4. Набор данных

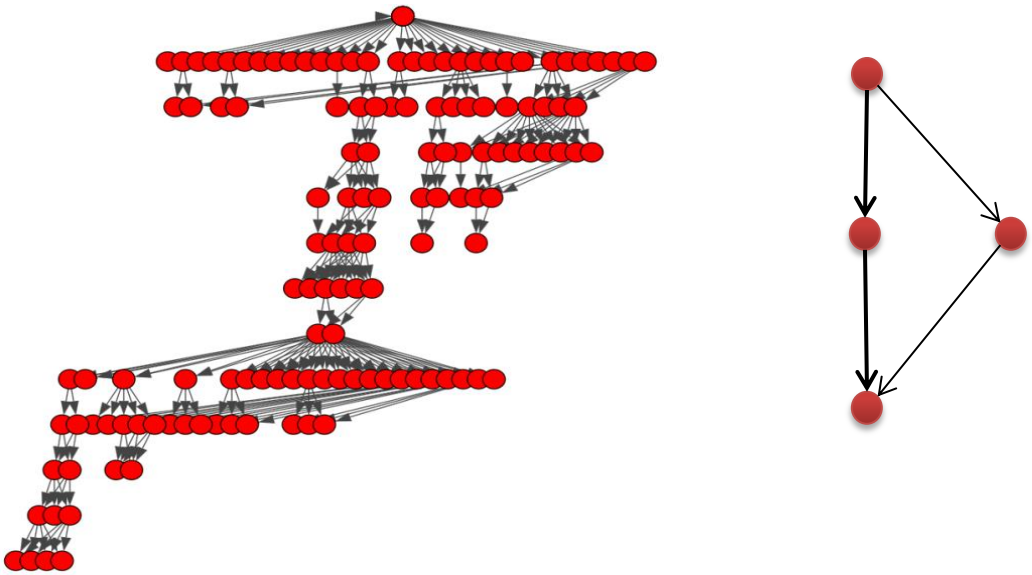
Важную роль в рамках данной задачи играет набор данных и их предварительная обработка. В качестве обучающей выборки были использованы данные, полученные в ходе исследования, описанного в работах [1]. Данные представляют из себя ориентированный GML (Graph Modelling Language) граф, каждый узел которого содержит в себе информацию об интенции автора, текст сообщения, имя автора и направленность сообщения. GML – иерархический формат файлов для описания графов, основанный на ASCII. Интенции представлены в виде 25 букв кириллического алфавита как показано в Таблице 1. Двадцать пять типов интенций объединяются в пять укрупненных групп по столбцам соответственно.

Таблица 1. Классификация типов интенции

| | | | | |
|---|---|---|---|---|
| Информативно-воспроизводящий тип. Воспроизвести в речи наблюдаемое | Эмотивно-консолидирующий тип. Предложение собственной картины мира для кооператив- ного | Манипулятивный тип. Со-циальное доминирование, установление иерархии | Волюнтивно-директивный тип. Побудить адресата к действию, внести изменения в фрагмент | Контрольно-реактивный тип. Выразить оценочную реак- цию на ситуацию |

| | | | | |
|---|---|---|---|---|
| | взаимодействия | | действительности | |
| **(А)** Удивление Вопрос | **(Е)** Самопрезен-тация | **(Л)** Оскорбление | **(Р)** Поощрение к оложительным действиям Рекомендации | **(Х)** Одобрение Похвала |
| **(Б)** Несогласие Сомнение | **(Ж)** Привлечение внимания Риторические вопросы | **(М)** Запугивание, Угрозы | **(С)** Подстрека-тельство к отрицательному | **(Ц)** Сарказм Злорадство |
| **(В)** Согласие Поддержка | **(З)** Успокоение аудитории Подбадривание | **(Н)** Дискредитация (нарушение полномочий) | **(Т)** Обвинение | **(Ч)** Критика |
| **(Г)** Непринятие \ Отказ от общения | **(И)** Прогнозы | **(О)** Демонстра-ция силы (без прямых угроз) | **(У)** Предупреж-дение о последствиях | **(Ш)** Ирония |
| **(Д)** Сострадание Симпатия | **(К)** Обоснование (как само-оправдание, например, без обвинений) | **(П)** Морализация, проповедь | **(Ф)** Отвод обвинений (если обвиняют) | **(Щ)** Разобла-чение |

Каждый граф представляет собой один пост в социальной сети “Вконтакте” и цепочки ответов пользователей на него. Пользователи могут отвечать как на первоначальный пост, так и на уже существующие ответы к посту, порождая тем самым древовидную структуру переписки. Длинные сообщения могут содержать несколько интенций. Визуально подобный дискурс можно представить в виде дерева интенций (Рис. 1а). Каждая вершина графа является интенцией пользователя.



(а) Граф всей дискуссии

(б) Фрагмента графа при наличии двух интенций во втором комментарии.

Рис. 1. Представление онлайн дискуссии в виде ориентированного графа.

Начальным шагом подготовки данных является извлечение из графа всех последовательностей интенций от корня до листьев дерева. В случае, когда сообщение состоит из двух интенций, оно представлено двумя вершинами, а каждая вершина предыдущего поста связана с каждой вершиной следующего (рис 1б). При построении всех возможных последовательностей от корня до листьев количество вариантов перемножается, образуя очень большое число путей. Для сокращения количества было принято решение использовать только первую (главную) интенцию из комментария и не использовать все второстепенные интенции. Таким образом, у нас получилось 13156 последовательностей в наборе данных.

## 5. Вычислительный эксперимент

Для вычислительного эксперимента с алгоритмами PPM, DG, CPT+ была выбрана библиотека SPMF[1], где они все реализованы. Искусственные нейронные сети были реализованы при помощи фреймворка Keras[2]. Исходный код программной реализации на

[1] http://www.philippe-fournier-viger.com/spmf/
[2] https://keras.io

языке Python доступен в открытом репозитории Dialog-Intent-DL[3]. Общая структура нейронных сетей приведена в Таблице 2.

Таблица 2.Структура нейронной сетей

| Структура рекуррентной сети | Структура сверточной сети |
|---|---|
| **Embedding**<br>**LSTM**<br>**Dropout**<br>**LSTM**<br>**Dropout**<br>**Dense**<br>**Dense** | **Embedding**<br>**Convolution1D**<br>**MaxPooling1D**<br>**Convolution1D**<br>**MaxPooling1D**<br>**Dense**<br>**Dense** |

Первый слой в искусственной нейронной сети (Embedding) ставит в соответствие типу интенции его векторное представление небольшой размерности (10). Далее эти вектора поступают либо на сверточную, либо на рекуррентную нейронную сеть. В данной работе не проводился подбор гиперпараметров и оптимальных топологий нейронных сетей, а только использованы их наиболее часто встречаемые реализации.

Для оценки точности работы алгоритмов подготовленные последовательности интенций разделены случайным образом на обучающую и тестовую подвыборки в пропорции 2/8. Максимальная длина последовательности на входе равна 5. При увеличении длины последовательности точность падает. Полученные величины точности приведены в Таблице 3.

Таблица 3.Точность предсказания интенции

| **Алгоритм** | **Точность для 25 классов** | **Точность для 5 классов** |
|---|---|---|
| PPM | 0,26 | 0,36 |
| DG | 0,23 | 0,38 |
| CPT+ | **0,48** | **0,66** |
| LSTM | 0.26 | 0.58 |
| CNN | **0.47** | **0.62** |

Из полученных результатов можно сделать следующие выводы:

[3] https://github.com/demid5111/dialog-intent-dl

- Алгоритмы CPT+ и RNN показывают себя лучше остальных.
- Использование укрупнённых групп ожидаемо даёт результаты, которые превышают точность предсказания не укрупненных типов интенций.
- Долгосрочные зависимости между последовательностью интенций не прослеживаются - гораздо большую точность показали эксперименты с меньшим количеством контекста.

## 6. Заключение

В настоящей работе проведено исследование последующей интенции пользователя в зависимости от интенции предыдущих реплик в дискурсе. Рассмотрены современные методы машинного обучения, которые позволяют анализировать последовательности и предсказывать их элементы. Рассмотренные алгоритмы применяются для предсказания следующей интенции пользователя в ходе дискурса. Оценка точности работы выбранных алгоритмов производится на оригинальном наборе данных, который сформирован из графов дискурса в виде набора последовательностей главных интенций.

Несмотря на то, что нейросетевые алгоритмы показали не самые лучшие результаты, остается ощущение, что подбор топологии сетей и гиперпараметров может дать прирост точности предсказания и незначительно обогнать алгоритм CPT+. Кроме этого они позволяют легко применять дополнительные признаки для улучшения точности, такие как вектор предыдущих комментариев Doc2vec и любые другие.

## 7. Благодарности

Статья подготовлена в результате проведения исследования (№ 17-05-0007) в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2017-2018 гг. и в рамках государственной поддержки ведущих университетов Российской Федерации "5-100".

## 8. Список литературы

1. Радина Н.К. Интент-анализ онлайн-дискуссий (на примере комментирования материалов интернет-портала ИноСМИ.ru ) // Медиаскоп. 2016. № 4.

2. Cleary J., Witten I. Data compression using adaptive coding and partial string matching // IEEE transactions on Communications. 1984. № 4 (32). С. 396–402.

3. Gueniche T. [и др.]. CPT+: Decreasing the time/space complexity of the Compact Prediction Tree Springer, 2015. 625–636 с.

4. Gueniche T., Fournier-Viger P., Tseng V.S. Compact prediction tree: A lossless model for accurate sequence prediction Springer, 2013. 177–188 с.

5. Hochreiter S., Schmidhuber J. Long short-term memory // Neural computation. 1997. № 8 (9). С. 1735–1780.

6. Karpov N., Demidovskij A., Malafeev A. Development of a Model to Predict Intention Using Deep Learning CEUR Workshop Proceedings, 2017. 69–78 с.

7. Laird P., Saul R. Discrete sequence prediction and its applications // Machine learning. 1994. № 1 (15). С. 43–68.

8. Padmanabhan V.N., Mogul J.C. Using predictive prefetching to improve world wide web latency // ACM SIGCOMM Computer Communication Review. 1996. № 3 (26). С. 22–36.

9. Pérez-Ortiz J.A., Calera-Rubio J., Forcada M.L. Online symbolic-sequence prediction with discrete-time recurrent neural networks Springer, 2001. 719–724 с.

10. Pitkow J. Mining longest repeated subsequences to predict World Wide Web surfing 1999.

11. Ron D., Singer Y., Tishby N. The power of amnesia: Learning probabilistic automata with variable memory length // Machine learning. 1996. № 2–3 (25). С. 117–149.

12. Sun R., Giles C.L. Sequence learning: from recognition and prediction to sequential decision making // IEEE Intelligent Systems. 2001. № 4 (16). С. 67–70.

13. Tax, N. Human Activity Prediction in Smart Home Environments with LSTM Neural Networks (статья будет опубликована позднее),.

14. Willems F.M., Shtarkov Y.M., Tjalkens T.J. The context-tree weighting method: basic properties // IEEE Transactions on Information Theory. 1995. № 3 (41). С. 653–664.

15. Zhao, Y. [и др.]. Sequence Prediction Using Neural Network Classifiers 164–169 с.