МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский университет   
«Московский институт электронной техники»

Институт cистемной и программной инженерии и информационных технологий

(СПИНТех)

Постоев Игорь Евгеньевич

Магистерская диссертация   
по направлению 09.04.04 «Программная инженерия»

Исследование и разработка модели и алгоритма формализованного представления историй болезни пациентов на основе неструктурированных данных

Студент Постоев И.Е.

Руководитель,   
к.т.н., доцент Андрианов А.М.

Москва 2021

Содержание

[Введение 3](#_Toc57104675)

[Актуальность исследования 3](#_Toc57104676)

[Объект исследования 4](#_Toc57104677)

[Предмет исследования 4](#_Toc57104678)

[Цели и задачи исследования 4](#_Toc57104679)

[Положения, выносимые на защиту 5](#_Toc57104680)

[Глава 1. Исследование проблемной области 6](#_Toc57104681)

[1.1 Общие положения NLP и NER 6](#_Toc57104682)

[1.2 Оценка качества выполнения задачи NER 8](#_Toc57104683)

[1.3 Анализ существующих способов решения 9](#_Toc57104684)

[Словари 9](#_Toc57104685)

[Грамматические правила 9](#_Toc57104686)

[Методы машинного обучения 10](#_Toc57104687)

Список сокращений

НДЭ – неструктурированные данные эпилепсии

NER – named entity recognition (рус. - распознавание имен сущностей)

NLP – native language processing(рус. – обработка естественного языка)

ИИ – искусственный иснеллект

POS – part-of-speech (рус. – часть речи)

# Введение

## Актуальность исследования

Эпилепсия — это тяжёлое заболевание головного мозга, характеризующееся периодическими нарушениями деятельности мозга, возникающими вследствие неконтролируемой активности нейронов головного мозга. Лечение эпилепсии продолжительностью 5-7 лет и сопровождается подбором методов в зависимости от анализа ЭЭГ и описаний приступов, которые составляются самими пациентами или их родственниками в неструктурированном виде. Частота приступов может составлять до нескольких эпизодов в день, в то время как исследование с помощью ЭЭГ из-за высокой стоимости производится сравнительно редко: раз в 3-4 месяца. Таким образом, описания приступов являются важным источником данных для врача при назначении и корректировки плана лечения. На сегодняшний день врачи должны вручную извлекать из текстов описаний полезные данные для внесения в историю болезни и принятия решений о лечении, что является трудоёмкой операцией.

Время проведения анализа описаний эпизодов было бы существенно уменьшено в случае, если данные описания были бы представлены в формализованном виде, согласно правилам оформления истории болезни и содержащими лишь актуальные для специалиста данные. В них бы фигурировали лишь те полезные данные (далее - сущности), которые являются значимыми согласно заранее определенным критериям. Данную задачу можно переформулировать как задачу извлечения полезных сущностей из неструктурированного текста, написанного на естественном языке. Она является типичным представителем класса задач распознавания имен сущностей(NER), являющееся подклассом более общего свода задач по обработке естественного языка(NLP).

## Объект исследования

Объектом исследования диссертационной работы являются неструктурированные тексты, описывающие протекание эпилепсии у пациентов.

## Предмет исследования

Предметом исследования диссертационной работы является алгоритм извлечения полезных данных из НДЭ.

## Цели и задачи исследования

Целью данной работы является определение способа (создание алгоритма) эффективного извлечения данных из НДЭ.

В соответствии с целью и предметом исследования в диссертации необходимо решить следующие задачи:

* определить основные проблемы и ключевые критерии оценки эффективности решения.
* исследовать существующие методы и алгоритмы NER с целью выявления их характеристик эффективности, преимуществ и недостатков;
* произвести формализованную постановку задачи извлечения данных из НДЭ;
  + разработать функциональную схему алгоритма формализованного представления историй болезни на основе НДЭ;
* провести экспериментальное исследование алгоритма;
  + создание тестовой модели на прежде сформированном корпусе данных для обучения;
  + разработать программное обеспечение для оценки эффективности алгоритма;
  + провести оценку результатов исследования, сравнительный анализ результатов разработки и экспериментов с существующими решениями.

## Положения, выносимые на защиту

1. Формализованное представление задачи статической ОВЗИП на основе вероятностного подхода и графового представления программы определяет выбор алгоритма оценки времени исполнения программ.
2. Алгоритм построения регулярного выражения по графу потока управления программы позволяет сохранить необходимую информацию о по- токе.
3. Оценка математического ожидания и дисперсии программы могут быть получены по регулярному выражению, описывающему все возможные пути из начальной вершины ГПУ этой программы в конечную.
4. Использование транзитивного замыкания и запоминание промежу- точных результатов позволяют ускорить процесс построения РВ по ГПУ в 5,7 и 2155,5 раз соответственно.
5. Экспериментальная проверка показывает, что отклонение получае- мой оценки математического ожидания от эталонного значения составляет от 1,3% до 6,2%. Отклонение же оценки дисперсии составляет 5%-110%.

# Глава 1. Исследование проблемной области

## 1.1 Общие положения NLP и NER

NLP является областью исследований обработки и интерпретации компьютером текстов естественных языков с учетом использования моделей ИИ. NLP охватывает множество задач: синтаксический и семантический анализы текста, определение именованных сущностей, определение эмоциональной окраски текста, его жанра, генерирование текста(вопросно-ответные системы), определение отношения слов друг к другу(интерференция) и т.д.

Как рассчитывается точность?

Задача ПЗС исследована хорошо(привести примеры)…в том числе, для медицинских текстов….

Однако, есть проблемы(нет fine-tuned исследования для эпилепсии + нет корпусов эпилепсии)….. + сложность для русского языка

На основе [,,,] наиболее точными методами являются...

Выбраны методы…

Есть гипотеза, что высокая точность может быть достигнута при …

Данные тексты в исходной форме неструктурированны. В процессе обработки, сначала проводится их сегментация (например, разбиение на предложения - сегменты) и токенизация – выделение отдельных слов – токенов в тексте. Затем проводится лемматизация – приведение слова к нормальной форме, например, для существительного – приведение к именительному падежу и единственному числу. Вместо лемматизации может проводиться стемминг – поиск корня слова. Далее происходит определение части речи токенов (POS-тегирование). В зависимости от части речи, токену присваивается определенный тег, например NN – сокращенно от английского «noun» – для существительного. После этого, выделяются комбинации слов (англ. - chunking) на основе POS-тегов и им также присваиваются теги. Наример, VB(verb phrase) – глагольная фраза. И каждому токену присваивается BIO-тег (beginning – начало фразы, inside – внутри фразы и outside – вне какой-либо комбинации). Таким образом, текст подготавливается для дальнейшего анализа и применения алгоритмов NER.

В большинстве текстовых документов присутствуют группы слов, которые являются наиболее информативными и описывают объекты реального мира. Например, такими объектами могут быть люди, локации, временные промежутки, имеющие конкретные описания - значения. Целью NER является идентификация подобных групп слов и соотнесение им определенного класса-метки объекта реального мира. Видно, что эта задача сводится к задаче классификации – определению класса группы слов. Пример подобной классификации приведен во фрагменте из новостного сайта: «Российский автономный необитаемый подводный аппарат впервые погрузился на дно <Марианской впадины/ЛОКАЦИЯ>. Об этом сообщает <РИА «Новости»/ОРГАНИЗАЦИЯ> со ссылкой на <Фонд перспективных исследований ([ФПИ](https://www.gazeta.ru/tags/organization/fond_perspektivnyh_issledovanii.shtml))/ОРГАНИЗАЦИЯ>. Отмечается, что своей цели устройство достигло в <22.34 мск 8 мая/ВРЕМЯ\_ДАТА>. Датчики зафиксировали глубину <10 028 метров/ПАРАМЕТР>»

К сложностям обработки русского языка относят:

* Анафоры – определение к какому существительному относится местоимение. Например: «Антон поехал в свой дом на машине, потому что он устал. Антон поехал в свой дом на машине, потому что он находится далеко». В первом случае «он» — это «Антон», а во втором случае «он» — это «дом».
* Свободный порядок слов – «Дети играют на улице. Играют на улице дети».
* Неологизмы – новые слова в языке.
* Полисемия – множество значений у одного слова (омонимы, омографы).

Также, имеют место сложности, присущие выбранным методам извлечения именованных сущностей.

## 1.2 Оценка качества выполнения задачи NER

Качество классификации обычно определяется точностью(precision), полнотой(recall), и их производной – f-мерой(f-score). Точность системы в пределах класса – это доля слов, действительно принадлежащих данному классу относительно тех слов, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором слов, принадлежащих классу относительно всех слов этого класса в тестовой выборке.

Пусть TP— истино-положительное решение TN — истино-отрицательное решение, FP— ложно-положительное решение, FN— ложно-отрицательное решение. Тогда формулы для определения точности и полноты выглядят:

F-мера представляет собой гармоническое среднее между полнотой и точностью:

F

## 1.3 Анализ существующих способов решения

### Словари

Классификация слов может происходить на основе заранее составленных словарей. Положительной стороной использования является простота реализации, отрицательными – ограниченность размера словарей. При составлении словаря соответствия метке (классу) слова или групп слов, необходимо учесть также и контекст слов, их сочетания, что является трудоемкой операцией. Однако, для повышения эффективности метода могут быть задействованы эвристические алгоритмы для уменьшения «жесткости» соответствия.

### Грамматические правила

В методах, основанных на грамматических правилах (rule-based) используются написанные специалистами регулярные выражения. Например, имея шаблон типа «^(19|20)\d\d**[**- /.**]**(0**[**1-9**]**|1**[**012**]**)**[**- /.**]**(0**[**1-9**]**|**[**12**][**0-9**]**|3**[**01**]**)» можно извлечь дату с 1900-01-01 до 2099-12-31. Подобный метод, с одной стороны, эффективен, когда извлекаемые сущности имеют явно определяемый шаблон. С другой стороны, метод не требует больших вычислительных ресурсов и времени анализа. Однако, процесс составления регулярных выражений для многих классов сущностей не так прост, как для даты и требует долгой кропотливой работы лингвистов. В настоящий момент метод актуален и применяется в некоторых решениях, в том числе, в совокупности с другими, более современными методами на определенных этапах анализа. Например, в решении для диетических рекомендаций drNER используются словари, регулярные выражения и методы машинного обучения вместе.

### Методы машинного обучения

Методы машинного обучения основаны на использовании размеченного человеком корпуса данных. С помощью него можно обучить модель для того, чтобы в дальнейшем использовать ее для определения меток слов на неразмеченном тексте.

Формально задачу извлечения структурированных данных эпилепсии можно представить следующим образом.

Пусть X – множество нормализованных слов-токенов, полученных из НДЭ, Y – множество классов (типов сущностей), которым необходимо соотнести токены, имеет мощность D. Элементы Y пронумерованы от 1 до D. – объекты обучающей выборки, – целевые значения для объектов множества . Элемент k вектора принимает значения из множества {0,1} и равен 1, если класс под номером k из множества Y присвоен токену .

Необходимо, на основании данных в X и Y, найти решающее правило, которое максимизирует вероятности правильного соотнесения класса токену на некотором распределении p(x,y).

Скрытые марковские цепи (HMM)

Метод основан на использовании модели, которая представлена тройкой параметров:

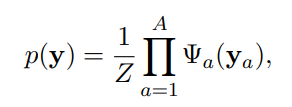
,

где A - матрица вероятностей переходов между состояниями, некоторые из которых являются скрытыми, B - матрица наблюдаемых значений, π - начальное распределение.

Основная цель обучения модели - найти оптимальные параметры модели для достижения максимума вероятности появления ожидаемой последовательности состояний. В применении алгоритма к задаче NER набором возможных состояний являются классы сущностей.

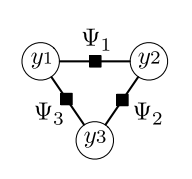
Случайные условные поля

В данном методе происходит факторизация результирующей вероятности распределения посредством функций (факторов). Итоговое распределение может быть выражено:



, где a – порядок фактора в наборе факторов, ψ – фактор, Z – параметр нормализации, обеспечивающий соблюдение границ [0,1] для p(y).

Строится ненаправленный граф, составляющими которого являются факторы и случайные переменные. С факторами соединяются те переменные, которые фактор принимает в качестве параметра.

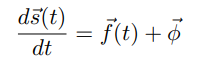


RNN и LSTM

RNN – рекуррентная нейронная сеть – метод обучения с учителем, в нейронной сети которого используется обратная связь. В данной модели, в случае обработки естественного языка, определяются эмбеддинги слов – векторы, которые содержат их значимые характеристики. В RNN они представлены как векторы (hidden state) – скрытое состояние. Операцией, при которой на основе входных слов строятся скрытые состояния модели, является кодированием и модель работает как encoder (кодировщик). В случае RNN, определяется эмбеддинг текущего слова, и скрытое состояние (или его часть) используется для определения состояния следующего слова. Таким образом, в формировании итогового состояния для всего предложения или участка текста, участвуют все его слова и в разной степени.

Рассмотрим математическую модель RNN. Пусть – многомерный вектор состояния модели.

Имеется нелинейное неоднородное дифференциальное уравнение первого порядка:



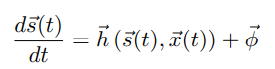
Здесь – векторная функция времени и – многомерный вектор констант.

Каноническая форма функции имеет вид:



– многомерный вектор входного сигнала и () – некоторая вектор-функция принимающая на вход векторные аргументы.

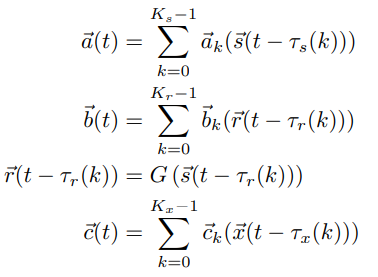
В итоге, получим:



Существует особый вид функции

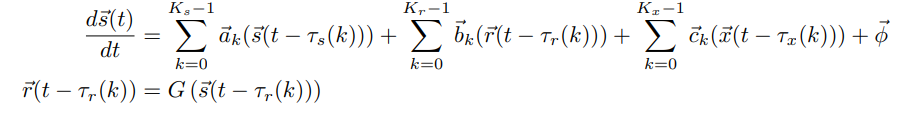


Здесь , – вектор-функции времени и данные компоненты определены как:

**

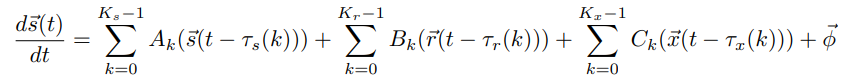
– сигнал выхода – обернутый в функцию сигнал состояния . Обычно такая функция – гиперболический тангенс. Элементы - постоянные дискретные задержки. количества функций соответственно.

В итоге получается система уравнений:



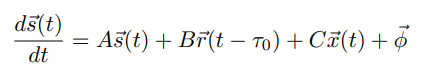
В итоге видно, что изменение состояния зависит от функции состояния (первое слагаемое), функции выходного сигнала (второе слагаемое) и функции над входным сигналом (третье слагаемое) – все это с учетом возможного временного сдвига – и константы «смещения». Компоненты задержек позволяют организовать «память» в системе, то есть мгновенное изменение сигнала состояния подвержено влиянию значений состояния, выхода и входа системы в разные моменты времени относительно текущего момента времени t.

Следует указать, что модель может быть представлена как в дискретном, так и аналоговом виде. Для обучения такой нейронной сети, используется алгоритм обратного распространения по времени(Back propagation through time) для обоих её представлений. Учитывая наличие временных задержек, они, наряду с другими весами нейронной сети, также подвергаются обучению и имеют, таким образом, гибкое представление.  
 Если предположить, что функции - линейны, то предыдущее уравнение можно представить, используя матрицы линейных коэффициентов:



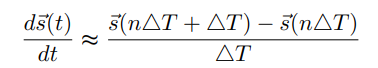
В случае, если , и циклические матрицы, данное уравнение можно представить в виде сверток.

Упрощенная форма уравнения выглядит следующим образом:



Для решения дифференциальных уравнений применяют метод Эйлера.

Тогда левая часть уравнения может быть представлена как:



Правая часть:





Временной интервал между шагами дискретизации обозначен ΔT, а их количество – n.

В итоге получим:

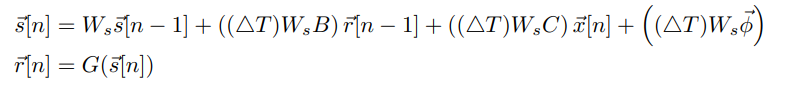


Отбрасывая ΔT и, соответственно, оставляя временную ось безразмерной, дальнейшие вычисления происходят уже с последовательностями сигналов.

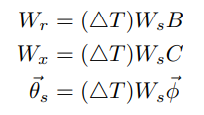




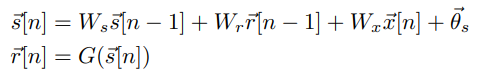
Пусть = , тогда умножая обе части уравнения на и произведя сдвиг последовательности на шаг назад, получается:



Определяются дополнительные матрицы весов:



В итоге, каноническая форма RNN для определения состояния модели на шаге n имеет вид:



Графически модель представлена на рис. 2:

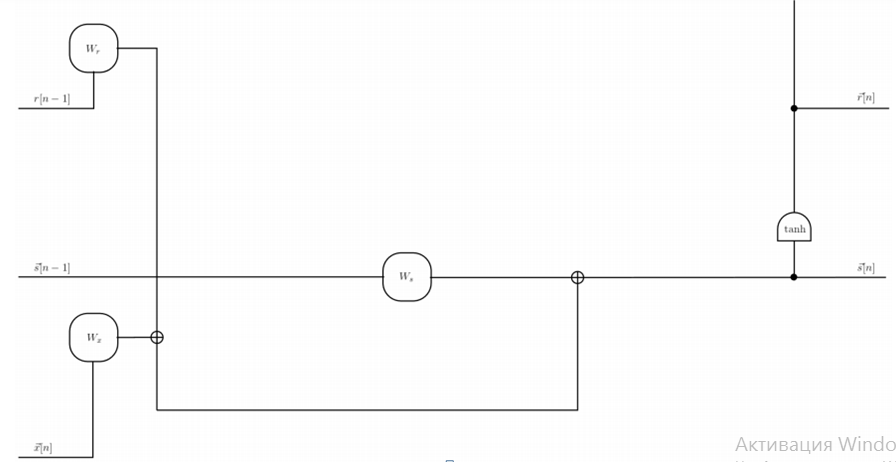


Рис.2 Графическое представление RNN на шаге n

В результате упрощений можно получить равенство, которое является стандартным определением RNN:



Модель можно представить как переиспользуемую ячейку, которая рекуррентно применяется для вычисления сигналов на каждом последующем шаге (рис. 3).

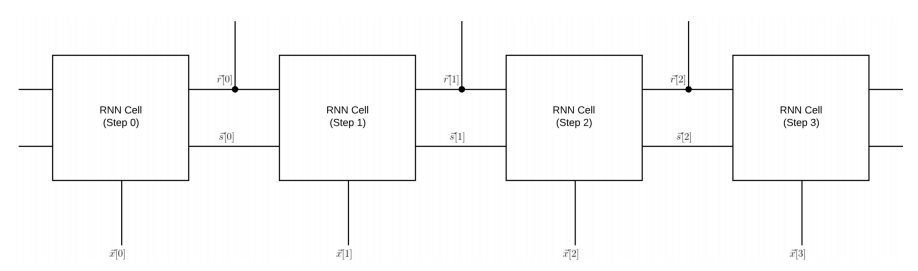


Рис.3 Развернутое представление переиспользуемой ячейки RNN

Таким образом, сигнал на шаге n зависит от всех сигналов, включая нулевой шаг. На практике обычно сколь угодно длинную последовательность выходных сигналов [n] для любого 0≤n≤N−1, где N – длина последовательности, разбивают на M сегментов, где - количество семплов (единиц входных/выходных данных, отсчетов) в пределах одного сегмента (его длина).

Представление параметров RNN обозначается как Θ:



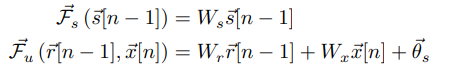
Также, определена функция стоимости L, определяющая отклонение вычисленных выходных значений от эталонных. Единственной переиспользуемой ячейки RNN, развернутой в определенном количестве шагов достаточно, чтобы вычислить неизвестную Θ, оптимизирующую L для обучения или определения скрытых входных значений.

Обучение такой модели проходит при использовании алгоритма обратного распространения по времени (BPTT). Ячейка RNN «разворачивается» во времени, то есть учитываются ее различные состояния на шагах. Для каждой совокупности этих шагов и соответствующих им входных данных имеется набор параметров Θ. При этом, градиент ошибки распространяется по обратному направлению индекса n.

Так как состояние на каждом последующем шаге жестко связано с выходным сигналом текущего шага, в ходе обучения модели градиент целевой функции относительно состояния [n] может затухать или резко увеличиваться. Увеличение размера окна приводит к увеличению чувствительности к стабильности системы, что делает RNN неэффективным при существующих зависимостях на больших интервалах входных данных.

Данная проблема решена в алгоритме LSTM. Пусть сигнал состояния на шаге n представлен равенством:





Здесь обозначает вклад предыдущего состояния, а - вклад предыдущего выходного сигнала и текущего элемента входных данных. Векторы и служат для тонкой настройки этих вкладов соответственно, и также, ∈ [].

Для решения проблемы с градиентами в RNN, в LSTM присутствует вектор , ограничивающий долю выходного с предыдущего шага.

Пусть



Тогда предыдущее уравнение с учетом – ограничителя входного сигнала (придает гибкость модели) преобразуется в



Для простоты, предполагая, что =1, пусть:



Тогда сигнал состояния:



Упрощенная схема ячейки LSTM имеет следующий вид:

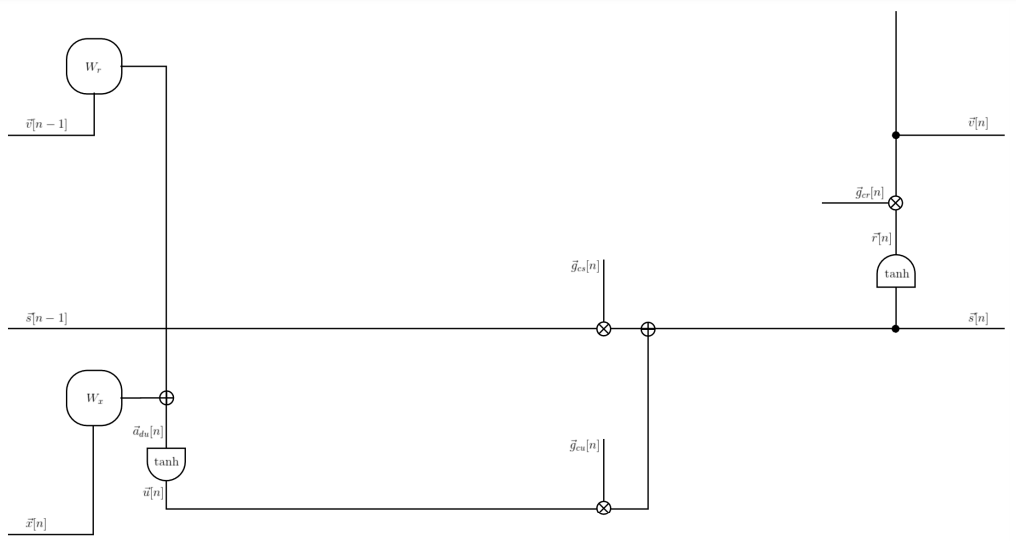


Рис. 4 Структура ячейки LSTM

Всего в LSTM используется 15 параметров. Архитектура LSTM также имеет ряд улучшений. В их числе:

- Внешние контекстные окна входа. В данном случае формируются матрицы коэффициентов, которые участвуют в конволюции с набором входных сигналов. При этом, в набор входных сигналов входят данные не только с предыдущих шагов, но и с будущих, формируя «контекстное окно».

- Рекуррентный проекционный уровень. Данный уровень представлен применительной к выходному сигналу матрицей весов, что позволяет регулировать количество выходных данных и скорость обучения.

Механизм внимания

Word2Vec/Glove

BiLSTM

ELMo

Transformer

BERT

[Определение требований к эффективному решению задачи.](#_heading=h.17dp8vu)

[Синтез идеи решения задачи](#_heading=h.3rdcrjn)

Литература

1. NLP Chowdhury GG. Natural language processing. Annual review of information science and technology. 2003;37(1):51–89.
2. CRF

<https://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf> !!!

1. Павлов <https://github.com/Factlink/pavlov>
2. Наташа <https://natasha.github.io/ner/>
3. <https://pdfs.semanticscholar.org/9528/4b31f27b9b8901fdc18554603610ebbc2752.pdf> <https://www.aclweb.org/anthology/P02-1060.pdf> - HMM in NER
4. Сводка данных о методах NER [http://redforester.com/2019/02/13/сравнение-подходов-к-ner-на-примере](http://redforester.com/2019/02/13/%D1%81%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BF%D0%BE%D0%B4%D1%85%D0%BE%D0%B4%D0%BE%D0%B2-%D0%BA-ner-%D0%BD%D0%B0-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B5)
5. Rule-based <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0179488#pone.0179488.e001>
6. LSTM + Keras <https://towardsdatascience.com/named-entity-recognition-ner-using-keras-bidirectional-lstm-28cd3f301f54>
7. Difference word2vec – Transformer <https://www.quora.com/What-are-the-main-differences-between-the-word-embeddings-of-ELMo-BERT-Word2vec-and-GloVe>
8. RNN + LSTM <https://arxiv.org/pdf/1808.03314.pdf>