Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Факультет управления и прикладной математики
Вычислительный центр им. А. А. Дородницына РАН
Кафедра «Интеллектуальные системы»

Романенко Александр Александрович

Применение условных случайных полей в задачах обработки текстов на естественном языке

010656 — Математические и информационные технологии

Выпускная квалификационная работа магистра

Научный руководитель:
д. ф.-м. н. Воронцов Константин Вячеславович

Москва
2014

Содержание

Аннотация

В работе рассматривается применение аппарата условных случайных (CRF) полей к двум задачам обработки естественного языка: выделению временны́х выражений и нормализации цифровой записи числительных. Для решения этих задач применяется линейная модель CRF.

Также в работе предлагается способ модификации линейной модели CRF для задач разметки последовательностей некоторого специального вида. Проведено сравнение модифицированной и классической моделей на примере задачи нормализации цифровой записи числительного.

Ключевые слова: *условное случайное поле, обработка естественного языка, временно́е выражение, нормализация числительных, conditional random field, linear chain CRF, time expression, temporal expression, number normalization*

1 Введение

Во многих областях обработки данных (обработка естественного языка, биоинформатика, распознавание изображений), часто встречаются задачи, которые можно свести к так называемой задаче сегментации и разметки последовательности (sequence segmentation and labeling problem). К таким задачам относятся, например, определение частей речи слов во фразе (POS-taging) [?], поверхностный синтаксический разбор (chunking) [?], поиск временных выражений [?] и разрешение анафоры [?]. В этих задачах входную последовательность длинны n из алфавита D необходимо отобразить в последовательность той же длины n, но из другого алфавита L.

Применительно к задачам обработки естественного языка, чаще всего, элемент выходной последовательности называют меткой или тэгом (label, tag). Также обычно элемент входной последовательности представляет собой некоторую самостоятельную единицу — слово в предложении, абзац в тексте или текст в коллекции текстов.

Существует большое количество подходов к решению задачи сегментации и разметки последовательности, причем основанных как на правилах [?], так и базирующихся на машинном обучении [?, ?, ?, ?]. В многочисленных работах (например, [?, ?]) была продемонстрирована большая эффективность выбора условных случайных полей на линейной цепи (Linear-chain CRF) в качестве статистической модели.

Линейная модель CRF является дискриминативной моделью [?], и в этом заключается ее сходство с популярной марковской моделью максимальной энтропии (Maximum entropy markov model, MEMM). Тем не менее в работах [?, ?] было показано, что в отличии от CRF, модель максимальной энтропии обладает недостатком, получившим название *смещение метки* (label bias). Суть проблемы состоит в том, что вследствие особенностей обучения, модель максимальной энтропии имеет тенденцию отдавать предпочтение тем скрытым состояниям, которые имеют меньшую энтропию распределения последующих состояний.

Линейная модель CRF успешно применена для решения многих задач обработки естественного языка в большинстве своем для английского языка, например, к задаче POS-тэггинга [?], к задаче поверхностного синтаксического анализа[?] или к задаче разрешения анафоры [?].

Применительно к русскому языку существуют две проблемы связанные с применением статистических методов для задач обработки естественного языка. Во-первых, отсутствие общепринятых и общедоступных размеченных корпусов для мно-

гих задач обработки текста. Во-вторых, из-за богатой морфологии русского языка корпусы должны быть достаточно большими, причем размечать их должны хорошо подготовленные люди, лингвисты. По этим причинам статистические методы применены к малому количеству задач для русского языка. Опыт использования CRF на русскоязычном материале описан в статье [?]. Линейая модель CRF была использована для POS-тэггинга, разрешения анафоры и анализа тональности текста.

В данной работе приводится описание аппарата условных случайных полей, дается подробное описание настройки и применения линейной модели CRF. Более того, в работе предлагается модификация линейной модели CRF, применимая ко многим задачам обработки текстов. В качестве примеров применения моделей рассмотрены две задачи: поиска временных выражений и нормализации цифровой записи числительных. На первой задаче демонстрируется работа классической линейной модели CRF. На примере второй задачи проводится сравнение модифицированной и классической моделей CRF.

Работа организована следующим образом. Сначала будет описан общий аппарат условных случайных полей. Затем будут описаны детали обучения и применения линейной модели. После этого будет предложена модификация линейной модели CRF. Затем будут описаны задачи выделения временных выражений и нормализации цифровой записи числительных и результаты вычислительных экспериментов, связанных с этими задачами.

2 Условные случайные поля

Условное случайное поле (*Conditional Random Field, CRF*) — это статистический метод классификации, характерным отличием которого является возможность учитывать «контекст» классифицируемого объекта. CRF является дискриминативной ненаправленной вероятностной графической моделью [?]. Одним из главных достоинств этой модели является то, что она не требует моделировать вероятностные зависимости между так называемыми наблюдаемыми переменными. Также этот метод, в отличие марковской модели максимальной энтропии, не имеет проблемы смещения метки (*label bias problem*) [?, ?]. CRF и различные его модификации нашли применения в таких областях как обработка естественного языка, компьютерное зрение, биоинформатика и т. д.

Широко распространенным в применении является линейное условное случай-

ное поле (*linear chain CRF*). Эта модель успешно применяется для решения задач машинного обучения, где прецедентом является последовательность случайных величин с поставленными им в соответствие метками. Это так называемые задачи разметки и сегментации последовательностей. Поэтому эта модель популярна в областях, для которых характерны данные в виде последовательности переменных, например, в задачах обработки естественного языка и биоинформатики.

Далее будет дано описание общей и линейной моделей CRF. Также будет описан вариационный вывод и даны комментарии по обучению модели для линейной модели CRF. В заключении раздела будет предложена одна естественная модификация линейной модели CRF, позволяющая уменьшить количество параметров модели и ускоряющая процесс обучения для некоторых задач разметки последовательности.

2.1 Общая модель условного случайного поля

Введем необходимые обозначения.

Пусть $\mathcal{A}$ — это конечное множество индексов. Пусть также $V = \{V_i | i \in \mathcal{A}\}$ — многомерная случайная величина, такая что каждая компонента V_i, являющаяся одномерной случайной величиной, принимает значение v_i и определена в своем вероятностном пространстве. Для удобства будем считать, что $\forall i$ V_i дискретны и множество их значений конечно. Обозначим реализацию многомерной случайной величины V как $v \in \Omega$, где Ω — множество всех возможных конфигураций.

Для удобства рассмотрения можно представлять, что множество индексов $\mathcal{A}$ задает множество точек на плоскости. Соответственно, рассматривается реализация многомерной случайной величины V в этих точках. Введенная таким образом случайная величина V называется случайным полем (сокращенно СП; в англоязычной литературе принято название *Random Field*).

Изобразим с помощью неориентированных ребер зависимости между компонентами случайного поля V. Пусть E — это множество ребер, отражающее все зависимости между компонентами.

Определение 1. Для любой случайной величины V_i его множеством соседей ∂i называется множество смежных с V_i вершин:

$$\partial i = \{j \in \mathcal{A} : j \neq i, (i, j) \in E\}.$$

Таким образом, многомерная случайная величина V и система зависимостей её компонент образуют ненаправленный граф $G = (V, E)$.

Определение 2. Пусть $G = (V, E)$ — неориентированный граф с множевством вершин V и множеством ребер E. Набор случайных величин $V_i, i \in \mathcal{A}$ образует Марковское случайное поле (*Markov Random Field, MRF*) по отношению к G, если выполнены следующие условия

1. $\forall v \in \Omega \; P(V = v) > 0$;

2. $P(V_i = v_i | V_j = v_j, j \in \mathcal{A} \setminus \{i\}) = P(V_i = v_i | V_j = v_j, j \in \partial i)$.

Определение 3. Полный подграф графа G называется кликой.

Пусть c — клика, а v_c — ограничение реализации v на c, то есть $v_c = (v_{i_1}, \ldots, v_{i_{|c|}})$, где $i_j \in c, j = 1, \ldots, |c|$. Пусть $C(G)$ — это множество всех клик графа $G = (V, E)$. Определим *функцию-фактор* $\Psi_c(v_c)$ как некоторую функцию $\Psi_c : C \to \mathbb{R}_+$.

Определение 4. Дискретное распределение называется *распределением Гиббса* [?], если

$$P(V = v) = \frac{1}{Z} \prod_{c \in C(G)} \Psi_c(v_c),$$

где Z — нормирующая константа, называемая статистической суммой, такая что:

$$Z = \sum_{v \in \Omega} \prod_{c \in C(G)} \Psi_c(v_c).$$

Наиболее важной теоремой, связывающей марковские случайные поля и распределение Гиббса, является теорема *Хаммерслея-Клиффорда* [?].

Теорема 1 (Хаммерслей-Клиффорд). V является марковским случайным полем, соответствующим $G = (V, E)$ тогда и только тогда, когда $P(V = v)$ — распределение Гиббса.

Определение 5. Условным случайным полем (УСП) (*Conditional Random Field, CRF*) называется MRF, у которого множество вершин разбито на два непересекающихся множества $V = X \cup Y$, где X и Y — множества наблюдаемых и скрытых переменных соответственно.

Всюду далее мы будем использовать следующее обозначение $\mathbf{x} = \{v \in X\}$ и $\mathbf{y} = \{v \in Y\}$. Также мы будем предполагать, что значения случайных величин из $\mathbf{x}$ и $\mathbf{y}$ принадлежат некоторым конечным пространствам $\mathcal{X}^{|\mathbf{x}|}$ и $\mathcal{Y}^{|\mathbf{y}|}$ соответственно.

Задача предсказания состоит в том, чтобы оптимальным образом восстановить значения $\mathbf{y}$, при условии, что нам даны наблюдаемые $\mathbf{x}$. Таким образом, в соответствии с теоремой Хаммерслея-Клиффорда нужно максимизировать

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Psi_c(\mathbf{x}, \mathbf{y}),$$

где $Z(\mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \prod_{c \in C} \Psi_c(\mathbf{x}, \mathbf{y}')$ — статистическая сумма.

Функции-факторы Ψ_c в подавляющем большинстве случаев является экспонентой от линейной комбинации некоторых признаков с весами, которые нужно определить в процессе обучения: $\Psi_c = \exp\left(\sum_{k=1}^{K} f_k(x_c, y_c) \cdot \theta\right)$.

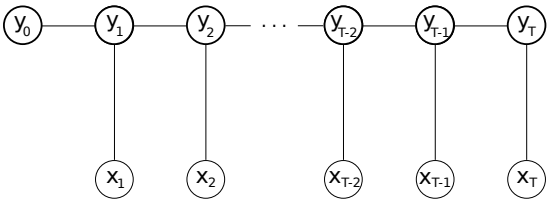
Стоит также отметить, что наиболее трудоемкой в смысле вычислительной сложности является задача оценки статистической суммы $Z(\mathbf{x})$, так как количество слагаемых в ней растет экспоненциально по размеру $\mathbf{x}$. Поэтому в общем случае при вариационном выводе и на этапе обучения $Z(\mathbf{x})$ не вычисляют точно, а лишь приблизительно оценивают. Однако для одного частного, но очень часто встречающегося на практике случая, существует метод точной оценки статистической суммы. В следующем разделе описан процесс обучения и вариационного вывода для этого случая.

2.2 Линейная модель условного случайного поля

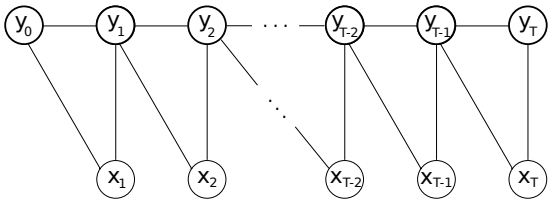
Определение 6. Линейно-цепочечное УСП (*linear-chain CRF*) — это УСП, у которого множество скрытых переменных вытянуто в цепочку.

Примеры линейных моделей CRF изображены на рис. **??**. На нем вершины y_1, $\ldots$, y_T соответствуют скрытым переменным $\mathbf{y}$, а $\mathbf{x}_1$, $\ldots$, $\mathbf{x}_s$ — наблюдаемым переменным. Первая скрытая переменная y_0 носит вспомогательный характер и используется только для того, чтобы все клики в графе были единообразны. В частности, без ограничения общности мы можем положить ее значение равным некоторому выделенному элементу $\mathcal{Y}$ — 'Start'. Отметим также, что наблюдаемые переменные обычно представляют из себя вектор случайных величин:

$$\mathbf{x}_t = \left(x_t^1, x_t^2, \ldots, x_t^{|\mathbf{x}_T|}\right), \; x_t^i \in \tilde{\mathcal{X}}.$$



(a) Линейная модель CRF, с размером максимальной клики 2.



(b) Линейная модель CRF, с размером максимальной клики 3.

Рис. 1: Примеры линейных моделей CRF.

Введем понятие функций признаков. Функции признаков — это индикаторные функции, такие что

$$\forall i, j \in \mathcal{Y}\ \forall o \in \tilde{\mathcal{X}} f_{ijo}(y', y'', x) = [y' = i][y'' = j][x = o].$$

Также иногда рассматривают еще и функции признаков вида $f_{io}(y', y'', x) = [y' = i][x = o]$ и $f_{ij}(y', y'', x) = [y' = i][y'' = j]$. Перенумеруем все функции признаков от 1 до N. Тогда запись $f_n(y_t, y_{t-1}, \mathbf{x}_t)$ будет означать n-ую функцию признаков для клики $(y_t, y_{t-1}, \mathbf{x}_t)$. Отметим, что количество функций признаков $N \geqslant |\mathcal{Y}|^2|\tilde{\mathcal{X}}|$.

Таким образом, функции-факторы $\Psi_c(\cdot)$, рассмотренные выше, можно выбирать равными

$$\Psi_t(y_{t-1}, y_t, \mathbf{x}_t) = \exp\left\{\sum_{n=1}^{N} \theta_n f_n(y_t, y_{t-1}, \mathbf{x}_t)\right\}.$$

Здесь $f_n(y_t, y_{t-1}, \mathbf{x}_t)$ — функции признаков, а θ_n — соответствующие параметры модели, которые нужно оценить в процессе обучения. Тогда вероятность цепочки скрытых переменных при условии цепочки наблюдаемых переменных равна

$$p(\mathbf{y}|\mathbf{x}) = \frac{\prod_{t=1}^{T} \Psi_t(y_{t-1}, y_t, \mathbf{x}_t)}{\sum_{\mathbf{y}' \in \mathcal{Y}^T} \prod_{t=1}^{T} \Psi_t(y'_{t-1}, y'_t, \mathbf{x}_t)}.$$

Оценка параметров модели. Пусть имеется обучающая выборка $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{Q}$. Будем считать, что длина i-ой цепочки из обучающей выборки равна $T(i)$. В дальнейшем, индекс i будем опускать для упрощения записей. Будем также считать, что для каждой клики выделяются N признаков.

Запишем условный логарифм правдоподобия:

$$l(\Theta) = \sum_{i=1}^{Q} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}).$$

Распишем условное распределение $p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$ через произведение функций-факторов и воспользуемся свойством логарифма произведения:

$$l(\Theta) = \sum_{i=1}^{Q}\sum_{t=1}^{T}\sum_{n=1}^{N}\theta_n f_n(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{Q}\log Z(\mathbf{x}^{(i)}).$$

Для того, чтобы уменьшить переобучение модели сделаем регуляризацию модели. Будем считать, что $\Theta \in \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Тогда

$$\tilde{l}(\Theta) = \sum_{i=1}^{Q}\sum_{t=1}^{T}\sum_{n=1}^{N}\theta_n f_n(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{Q}\log Z(\mathbf{x}^{(i)}) - \sum_{n=1}^{N}\frac{\theta_n^2}{2\sigma^2}.$$

Здесь, σ — гиперпараметр, который тоже необходимо оценить. Однако, как показывает практика [**?**], качество модели практически не меняется при увеличении или уменьшении параметра σ в десятки раз. Также иногда используется l_1-регуляризация [**?**], что ведет к разреживанию вектора параметров Θ.

Возьмем частную производную по параметру θ_n :

$$\frac{\partial\tilde{l}(\Theta)}{\partial\theta_n} = \sum_{i=1}^{Q}\sum_{t=1}^{T}\sum_{n=1}^{N} f_n(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{Q}\sum_{t=1}^{T}\sum_{y,y'} f_n(y, y', \mathbf{x}_t^{(i)})p(y, y'|\mathbf{x}_t^{(i)}) - \sum_{n=1}^{N}\frac{\theta_n}{\sigma^2}.$$

Так как функция $g(\mathbf{x}) = \log\sum_i \exp x_i$ выпуклая, то и функция $l(\Theta)$ — выпуклая. Тогда из-за регуляризационного члена функция $\tilde{l}(\Theta)$ будет строго выпуклой, то есть любой ее локальный оптимум будет глобальным оптимумом, при этом она будет иметь ровно один глобальный оптимум.

Самый простой способ оптимизировать $\tilde{l}(\Theta)$ — использование градиентного спуска вдоль градиента $\frac{\partial\tilde{l}(\Theta)}{\partial\theta_n}$, однако это требует слишком много итераций, а потому с трудом реализуемо на практике. Ньютоновские методы оптимизации сходятся намного быстрее, но для их применения необходимо вычислять и хранить гессиан, матрицу всех вторых производных $\tilde{l}(\Theta)$. С учетом того, что в практических задачах количество оцениваемых параметров может достигать миллионов, то ньютоновские методы также становятся не столь применимыми на практике.

По этим причинам чаще всего используют квази-ньютоновские методы оптимизации, такие как BFGS [**?**], которые лишь делают аппроксимацию гессиана. Полная $N \times N$ аппроксимация гессиана также требует квадратичных затрат по памяти, однако есть вариант BFGS с ограниченным использованием памяти [**?**]. Есть также ряд других техник по оптимизации функции $\tilde{l}(\Theta)$ [**?**, **?**, **?**].

Вариационный вывод. В течение процесса обучения при вычислении градиента необходимо знать маргинальное распределение $p(y_t, y_{t-1}|\mathbf{x})$. Также для вычисления самого правдоподобия необходимо знать $Z(\mathbf{x})$. Помимо этого нужно уметь прогнозировать последовательность скрытых переменных при наблюдении последовательности наблюдаемых:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}).$$

В случае линейной модели CRF, все эти задачи можно выполнить используя различные варианты алгоритмов динамического программирования. В данном разделе описываются эти алгоритмы применительно к линейной модели CRF. Классической описание можно найти в [**?**]. В случае же модели CRF общего вида для решения этих задач используют обобщения этих алгоритмов, например, алгоритм «пересылки сообщений» [**?**].

Пусть $\Psi_t(y_t, y_{t-1}, \mathbf{x}_t) = \exp\left\{\sum_{n=1}^{N} \theta_n f_n(y_t, y_{t-1}, \mathbf{x}_t)\right\}$, тогда распишем выражение статистической суммы:

$$\begin{aligned} Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t) \\ = \sum_{y_T} \sum_{y_{T-1}} \Psi_T(y_T, y_{T-1}, \mathbf{x}_T) \cdot \\ \cdot \sum_{y_{T-2}} \Psi_{T-1}(y_{T-1}, y_{T-2}, \mathbf{x}_{T-1}) \sum_{y_{T-3}} \dots \end{aligned}$$

Как видно, каждая внутренняя сумма, будучи вычислена один раз, может быть сохранена и многократно использована для вычисления внешних сумм. Сделаем обозначение

$$\alpha_t(j) = \sum_{\mathbf{y}_{<1\dots t-1>}} \Psi_t(j, y_{t-1}, \mathbf{x}_t) \prod_{\tau=1}^{t-1} \Psi_\tau(y_\tau, y_{\tau-1}, \mathbf{x}_\tau), \text{ для } \forall j \in \mathcal{Y}.$$

то есть $\alpha_t(j) \propto p(\mathbf{x}_{<1\dots t>}, y_t = j)$. Здесь, $\mathbf{x}_{<1\dots t>} = [\mathbf{x}_1, \dots, \mathbf{x}_t]$, и аналогично для $\mathbf{y}_{<1\dots t-1>}$.

Тогда, если $\alpha_1(j) = \Psi_1(j, y_0, \mathbf{x}_1)$, то с помощью рекурсии можно вычислить

$$\alpha_t(j) = \sum_{i \in \mathcal{Y}} \Psi_t(j, i, \mathbf{x}_t)\alpha_{t-1}(i) \text{ и } Z(\mathbf{x}) = \sum_{i \in \mathcal{Y}} \alpha_T(i).$$

Такая рекурсивная процедура называется «Forward».

Аналогично, можно провести произвести процедуру «Backward». Обозначим,

$$\beta_t(j) = \sum_{\mathbf{y}_{<t+1\dots T>}} \prod_{\tau=t+1}^{T} \Psi_\tau(y_\tau, y_{\tau-1}, \mathbf{x}_\tau),$$

то есть $\beta_t(j) \propto p(\mathbf{x}_{<t+1\ldots T>}|y_t = j)$.

Тогда, проинициализировав $\beta_T(i) = 1$, можно вычислить

$$\beta_t(j) = \sum_{i\in\mathcal{Y}} \Psi_t(i, j, \mathbf{x}_{t+1})\beta_{t+1}(i) \text{ и } Z(\mathbf{x}) = \sum_{i\in\mathcal{Y}} \Psi_1(i, y_0, \mathbf{x}_1)\beta_1(i).$$

Итак, с помощью одной из процедур «Forward» или «Backward», можно вычислить статистическую сумму. А для того, чтобы вычислить маргинальное распределение $p(y_t, y_{t-1}|\mathbf{x})$ воспользуемся сразу двумя процедурами:

$$p(y_{t-1}, y_t|\mathbf{x}_t) = \alpha_{t-1}(y_{t-1})\Psi_t(y_t, y_{t-1}, \mathbf{x}_t)\beta_t(y_t).$$

Теперь опишем, как найти оптимальную последовательность скрытых переменных $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$. Пусть $\delta_1(j) = \sum_{n=1}^{N} \theta_n f_n(y_0, j, \mathbf{x}_1)$. Тогда $\forall\tau \in \{2, \ldots, T\}$ и $\forall j \in \mathcal{Y}$:

$$\delta_\tau(j) = \max_{i\in\mathcal{Y}} \left(\delta_{\tau-1}(i) + \sum_{n=1}^{N} \theta_n f_n(i, j, \mathbf{x}_\tau) \right).$$

В итоге,

$$\max_{\mathbf{y}} \sum_{t=1}^{T} \sum_{n=1}^{N} \theta_n f_n(y_t - 1, y_t, \mathbf{x}_t) = \max_{j\in\mathcal{Y}} \delta_T(j).$$

Отметим, что полученная оптимальная последовательность $\mathbf{y}^*$ называется *путем Витерби*, а сам алгоритм ее нахождения — *алгоритмом Витерби*. Отметим также, что алгоритм выполняется за время $O(T|\mathcal{Y}|^2)$.

2.3 Модифицированная модель условного случайного поля

Также как и алгоритм Витерби, процедуры «Forward» и «Backward» требуют $O(T|\mathcal{Y}|^2)$ вычислений. Однако при обучении мы должны запускать эти процедуры для каждого прецедента из обучающей выборки $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{Q}$. Поэтому вся процедура имеет вычислительную сложность $O(T|\mathcal{Y}|^2QG)$, где G — число итераций, необходимых для нахождения максимума правдоподобия $\tilde{l}(\Theta)$. Поэтому для некоторых задач с большой обучающей выборкой и большим значением $|\mathcal{Y}|$ обучение модели может занимать большое время [**?**].

Кроме того, в некоторых задачах есть необходимость для некоторых y_i из цепочки скрытых переменных лишь выбрать метку из некоторого допустимого множества $\mathit{fflcalY}' \subset \mathcal{Y}$, причем $\mathcal{Y} \equiv \mathcal{Y}'_1 \times \mathcal{Y}'_2 \times \cdots \times \mathcal{Y}'_r$. То есть фактически некоторые элементы последовательности нужно классифицировать на несколько классов. Примером такой задачи может быть снятие морфологической неоднозначности в текстах. В этой

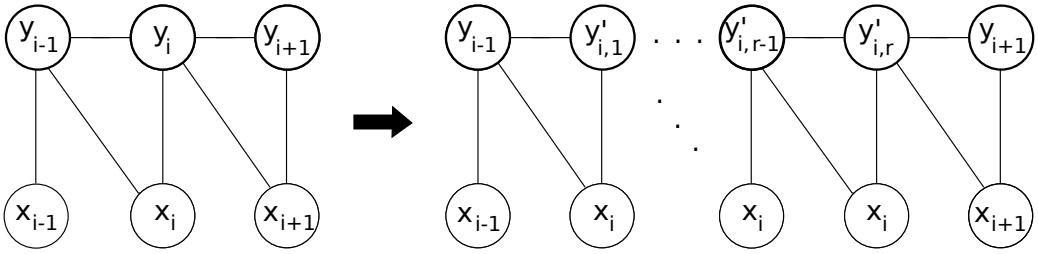


Рис. 2: Модифицированная модель CRF, используемая для определения типа и формы числительного.

задаче прецедентом является предложение, то есть последовательность слов на естественном языке. Каждому слову нужно поставить в соответствие скрытую переменную — морфологический разбор слова. Применимо к этой задаче $\mathcal{Y}$ — это множество всевозможных разборов слов, которые могут быть в языке, $\mathcal{Y}'$ — это разборы слова, которые может иметь конкретное слово по словарю, а $\mathcal{Y}'_j, j = 1, \ldots, r$ — множество значений определенной грамматической характеристики слова (например, часть речи, для прилагательных — падеж, род, для глаголов — склонение, переходность и т. д.).

Для задач такого типа предлагается естественная модификация классической линейной модели CRF. Предлагается вытянуть одну скрытую переменную $y \in \mathcal{Y}'_1 \times \mathcal{Y}'_2 \times \cdots \times \mathcal{Y}'_r$ в цепочку из r скрытых переменных $y'_1, \ldots, y'_r$, где $y'_j \in \mathcal{Y}'_j, j = 1, \ldots, r$. При этом, для каждой новой скрытой переменной y'_j, полученной вытягиванием старой переменной y_i, в качестве наблюдаемой переменной предлагается брать скрытую переменную переменную $\mathbf{x}_i$. Идея модификации изображена на рис. **??**. Применение этой модификации уменьшит размер множества значений скрытых переменных на $\prod_{j=1}^{r} |\mathcal{Y}'_j| - \sum_{j=1}^{r} |\mathcal{Y}'_j|$. Кроме того, если при определенном соотношении значений r и мощностей множеств $\mathcal{Y}'_j$ общее количество параметров уменьшится. Все это говорит о том, что использование модифицированной модели уменьшает эффект переобучения, а также ускоряет вычисление статистической суммы $Z(\mathbf{x})$ и маргинального распределения $p(y_t, y_{t-1}|\mathbf{x})$.

3 Задача выделения временных выражений

В данном разделе описывается задача выделения временных выражений в предложениях на естественном языке (для русского языка), а также применение классической линейной модели CRF для ее решения. Помимо этого проводится сравнение

линейной модели CRF как со статистическим методом классификации, так и с нестатистическим, шаблонным методом выделения временных выражений.

3.1 Описание задачи

Задача выделения временных выражений (в англ. *Temporal Expressions Extraction*) является частным случаем задачи выделения именованных сущностей (*Named Entity Recognition*). Обычно, после решения задачи выделения временных выражений решают задачу их нормализации, то есть привязки найденного выражения к временной оси.

Временным выражением (*temporal expression, timex*) называется выражение естественного языка, несущее временную окраску и обозначающее точку во времени, промежуток времени или периодичность некоторого события. Понятие временного выражения довольно расплывчато. Тем не менее, в рамках ISO был принят стандарт разметки и нормализации временных выражений «*TimeML*» [**?**].

Ниже приведены примеры временных выражений.

Что будут показывать <TIMEX>*сегодня ночью*</TIMEX> по пятому каналу?
<TIMEX>*8 сентября 2013 года*</TIMEX> состоялись выборы на пост мэра Москвы.
<TIMEX>*Через 2 недели*</TIMEX> состоится встреча с руководителем.
Какую телепередачу показывают <TIMEX>*ежедневно в 7 часов вечера*</TIMEX>?

Надо отметить, что слова «*мгновенно*», «*долго*» и т. д. принято не относить к временным выражениям.

Для решения задачи выделения временных выражений разработано два основных подхода. Первый подход основан на поиске в предложении определенных шаблонов[**?**, **?**], второй — на применении методов машинного обучения[**?**]. Для выделения временных выражений в англоязычных текстах успешно применяются оба подхода. Первый подход требует усилий лингвистов и наличия статистики употребления временных выражений в данном языке. Второй подход требует наличия размеченного корпуса текстов. Первый подход, как правило, используется при извлечении информации из текстов ограниченной тематики. Это обоснованно меньшим количеством видов временных выражений, встречающихся в текстах [**?**]. В рамках второго

подхода не требуется специальных лингвистических знаний, а алгоритмы анализа часто оказываются более эффективными.

3.2 Подход к решению

В [?] в рамках спецификации «*TimeML*» предложен метод разметки текстов в XML стиле. Эта разметка громоздка и содержит много лишних тегов, ненужных для выделения временных выражений. Поэтому использовались другие более простые разметки BIO (*Begin Inside Out*) и IO (*Inside Out*) [?].

Для сведения задачи поиска временных выражений в тексте к задаче проставления меток используется следующий подход:

1. если i-й токен входной последовательности является началом некоторого временного выражения, то на i-й позиции выходной последовательности устанавливается метка B;

2. если i-й токен является частью временного выражения и не является его началом, то на i-й позиции выходной последовательности устанавливается метка I;

3. в противном случае на i-й позиции выходной последовательности устанавливается метка O.

В множество допустимых признаков $\mathcal{X}$ входили следующие группы признаков:

1. Все варианты разбора токена по словарю, составленному на основе словаря OpenCorpora (омонимия не снята) [?]. Это, например, такие признаки как «сущ.», «глагол», «кр. прил.», «дат. п.», «сов. вид» и т. д.

2. Признаки, основанные на написании токена: «содержит цифры», «начинается с заглавной буквы», «все буквы прописные» и т. д.

3. Признаки, характеризующие положения токена в предложении: «первый токен», «последний токен» или «токен в середине предложения».

4. Является ли токен словом-триггером: «название месяца», «время суток» и т. д.

5. Перечисленные выше признаки для соседних токенов.

Таким образом, задача выделения временного выражения сводится к задаче разметки последовательности:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}),$$

где $\mathbf{y} = [y_1, \ldots, y_T], y_i \in \mathcal{Y} \equiv \{B, I, O\}$ — скрытая последовательность переменных, $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_T], \mathbf{x}_i \in \mathcal{X}$ — последовательность наблюдаемых переменных.

3.3 Используемые данные

Для русского языка на настоящий момент еще не создано корпуса с разметкой временных выражений, поэтому для тестирования статистической модели была выполнена ручная BIO-разметка около 2000 предложений из корпуса OpenCorpora, содержащих около 500 временных выражений.

Тем не менее, обучение алгоритма требует значительно больше количества размеченных предложений. Поэтому для получения обучающих данных использовалась следующая схема:

1. На первом шаге шло написание регулярных выражений, на основе которых тексты размечались автоматически.

2. На втором шаге полученные данные анализировались, определялся вид временных выражений еще не обрабатываемых правилами.

3. На третьем шаге новые правила добавлялись в список регулярных выражений.

4. Далее процедура повторялась с учетом новых правил.

В результате этого процесса был получен base-line алгоритм для выделения регулярных выражений, а также большой размеченный корпус, используемый для обучения. Идею описываемого процесса описывает рис. **??**.

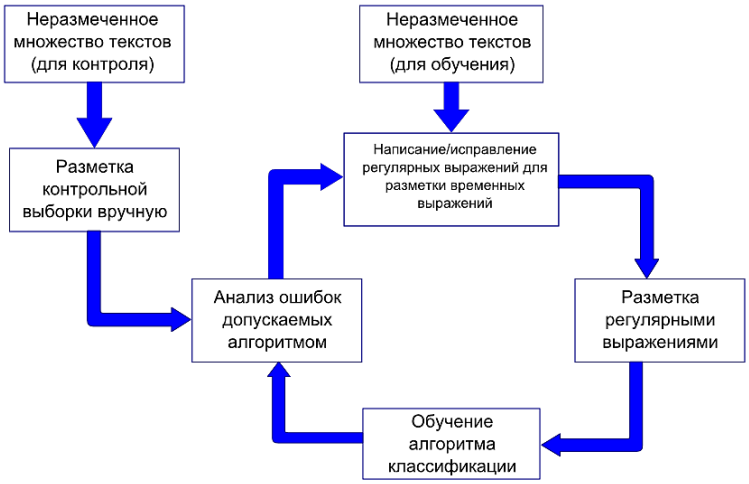


Рис. 3: Общая схема разметки данных в задаче выделения временных выражений

3.4 Отбор признаков

Так как размер допустимого множества признаков $\mathcal{X}$ достаточно большой, то было решено воспользоваться алгоритмом отбора признаков. В качестве такого алгоритма был выбран алгоритм «Random Forest» (случайный лес решающих деревьев) [?]. Это стохастический логический метод машинного обучения. Кроме отбора признаков мы также получаем для сравнения еще один алгоритм выделения временных выражений. К преимуществам этого алгоритма относятся также высокая обобщающая способность, обработка данных в признаковом пространстве высокой размерности и обработка дискретных признаков.

Эксперименты проводились для разных количеств соседей, признаки которых добавлялись к признакам слова, а также при разных количествах признаков оставляемых алгоритмом отбора. В итоге наилучшей конфигурацией оказалась следующая: к признакам слова добавлялись наиболее значимые признаки двух правых и двух левых соседей. В итоге получили 55 признаков.

3.5 Вычислительный эксперимент

Для обучения и тестирования были взяты данные из корпуса OpenCorpora. Данные были размечены автоматически шаблонным base-line алгоритмом и разбиты на обучающую (380000 предложений, 5000 выражений) и контрольную (40000 предложений, 900 выражений) выборки. Также для тестирования использовалась часть тек-

стов размеченная вручную (2000 предложений, 500 выражений).

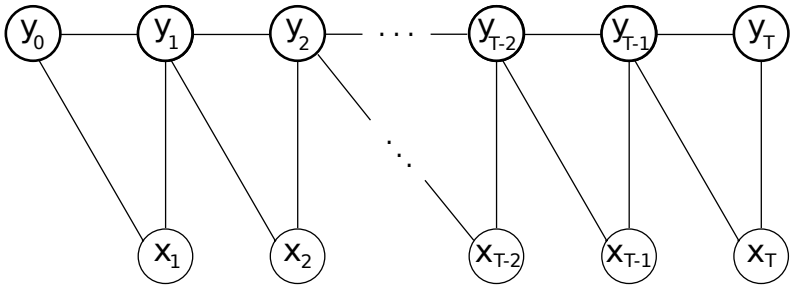


Рис. 4: Общая схема разметки данных

На обучающей выборке были обучены алгоритм Random Forest и линейная модель CRF. Схема применяемой модели изображена на рис. **??**. Результаты работы алгоритмов переводились из «BIO» разметки в «IO» разметку. Затем высчитывались меры качества классификации на 2 класса:

- полнота $R = \frac{tp}{tp+fn}$,
- точность $P = \frac{tp}{tp+fp}$,
- $F_1 -$ мера $F_1 = \frac{2PR}{P+R}$.

Ниже приведены результаты работы шаблонного алгоритма, CRF и Random Forest на тестовой выборке, размеченной вручную. Стоит отметить, что результаты указаны при для наилучшей конфигурации признаков, то есть на наиболее важных признаках, отобранных с помощью Random Forest.

Алгоритм	P	R	F_1
Base-line	95,7	85,7	90,4
RF	96,4	87,1	91,5
CRF	96,3	89,9	93,05

Анализ результатов. Из результатов разметки предложений можно отметить следующие факты.

- CRF показывает лучшие результаты, за счет разметки последовательности целиком, а не отдельно по одному слову, как в случае Random Forest.
- Random Forest и CRF перенастраиваются на специальные слова: отсюда ложная классификаций фраз «комендантский час», «торговая неделя», «мальчик восьми лет» и т. д.

- RF и CRF верно отрабатывают на фразах «в прежние годы», в отличие от base-line алгоритма

- Ошибки на нетипичных выражениях: «на рубеже XIX века», «в 8 по Гринвичу», «за сто двадцать восемь дней и ночей»

- Не малочисленны ошибочные пропуски слов во временных выражениях («не раньше, чем через 7 часов»), то есть качество можно повысить, за счет постобработки.

4 Задача нормализации цифровой записи числительных

4.1 Описание задачи

Одной из частых задач, с которыми сталкиваются на этапе синтеза речи, является правильное прочтение машиной так называемых «нестандартных» слов, то есть аббревиатур, сокращений, а также числительных, представленных в предложении цифрами [**?**, **?**]. Для того чтобы TTS система (*Text-To-Speech System*) правильно произнесла фразу с «нестандартным» словом, нужно расшифровать и поставить его в правильную грамматическую форму.

Под грамматической формой числительного, записанного цифрами, понимается его тип (порядковое или количественное), падеж, число, род и одушевленность. Затем на основе этой формы и цифровой записи числительного можно однозначно восстановить его словесную форму, например, с помощью конечного автомата.

В данной части приведен пример применения различных вариаций линейных моделей CRF для определения грамматической формы числительного, записанного цифрами. В [**?**] описывается применение других методов для решения этой задачи, опираясь только на частотные характеристики данных и не используя грамматических свойств контекста употребления числительного. Также в [**?**] рассматривалась задача только для количественных числительных. В работе же предлагается учитывать грамматические характеристики слов из контекста для определения типа числительного (порядковое или количественное) и его грамматической формы.

4.2 Подход к решению

Множество $\mathcal{X}$ всех признаков наблюдаемых переменных в этой задаче распадается на несколько групп:

$$\mathcal{X} = \mathrm{GRAM} \cup \mathrm{SPEL} \cup \mathrm{SPEC} \cup \mathrm{NEAR}.$$

- GRAM — грамматические метки слов («существительное», «предлог» и т. д.).
- Метки особенностей написания числительного (например, длина в символах) были отнесены в группы SPEL.
- В связи с тем, что некоторые слова (например, названия валют, названия месяцев, и т. д.) употребляются в подавляющем большинстве случаев с определенным типом числительных, были введены два дополнительных признака: «характерно с количественным числительным» и «характерно с порядковым числительным». Эта группа признаков SPEC.
- NEAR — это признаки соседних слов из групп, описанных выше.

Множество значений $\mathcal{Y}'$ скрытых переменных, соответствующих числительным, можно представить в декартова произведения множеств:

$$\mathcal{Y}' = \mathrm{TYPE} \times \mathrm{CASE} \times \mathrm{GEND} \times \mathrm{SNGL} \times \mathrm{ANIM}.$$

Здесь

- TYPE — тип числительного (количественное или порядковое)
- CASE — падеж числительного (именительный, родительный, ...)
- GEND — род числительного (мужской, женский, средний, неизвестно)
- SNGL — число числительного (единственное, множественное, неизвестно)
- ANIM — одушевленность числительного (одушевленное, неодушевленное, неизвестно)

То есть, для каждого числительного нужно определить его тип, падеж, род, число и одушевленность.

Также предполагается, что грамматическая форма числительного зависит от предыдущего и следующего слов. Поэтому будем определять еще и часть речи для соседних слов числительных, а оставшиеся слова можно пометить меткой O.

Таким образом, задача определения грамматической формы числительного сводится к задаче разметки последовательности:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}),$$

где $\mathbf{y} = [y_1, \ldots, y_T], y_i \in \mathcal{Y} = \{O\} \cup \text{POS} \cup \text{TYPE} \times \text{CASE} \times \text{GEND} \times \text{SNGL} \times \text{ANIM}$ — скрытая последовательность переменных, $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_T], \mathbf{x}_i \in \mathcal{X} = \text{GRAM} \cup \text{SPEL} \cup \text{SPEC} \cup \text{NEAR}$ — последовательность наблюдаемых переменных. Схема линейной модели CRF, применяемая для решения этой задачи, изображена на рис. **??**.

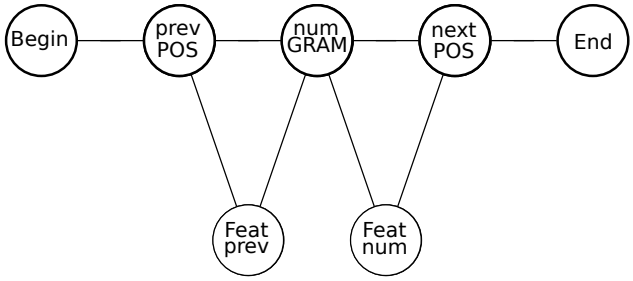


Рис. 5: Модифицированная модель CRF, используемая для определения типа и формы числительного.

Заметим, однако, что при таком множестве $\mathcal{Y}'$ количество возможных грамматических меток числительного велико: $2 \cdot 6 \cdot 4 \cdot 3 \cdot 3 = 432$. Чтобы уменьшить количество меток, «растянем» одну грамматическую метку числительного из пространства TYPE × CASE × GEND × SNGL × ANIM в цепочку из пяти меток из пяти пространств TYPE, CASE, GEND, SNGL и ANIM, то есть воспользуемся модифицированной линейной моделью CRF. Тогда, множество значений скрытых переменных будет выглядеть следующим образом:

$$\tilde{\mathcal{Y}} = \{O\} \cup \text{POS} \cup \text{TYPE} \cup \text{CASE} \cup \text{GEND} \cup \text{SNGL} \cup \text{ANIM}.$$

Схема этой модифицированной линейной модели CRF, изображенна на рис. **??**.

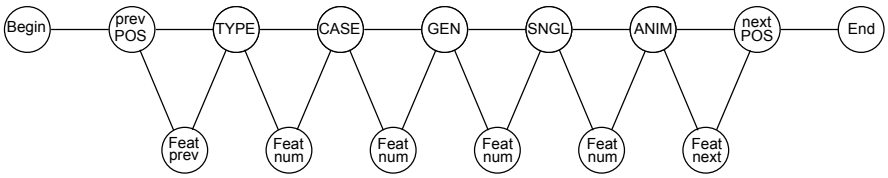


Рис. 6: Модифицированная модель CRF, используемая для определения типа и формы числительного.

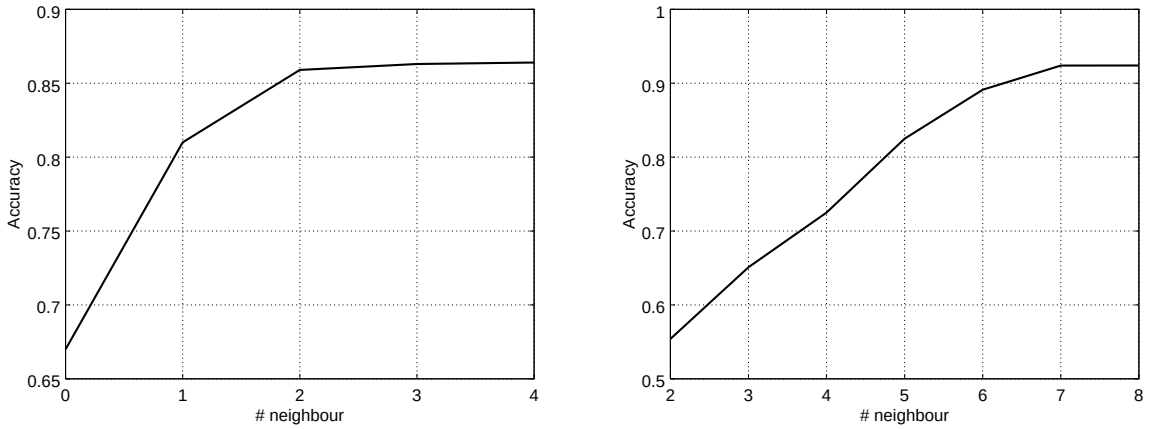
4.3 Вычислительный эксперимент

Данные. Для получения данных для контроля и обучения использовалась часть Национального корпуса русского языка. Из него выбирались предложения, содержащие числительные, имеющие в описании грамматические характеристики. Далее все эти числительные представлялись в виде цифровой записи. В итоге всего получилось 10268 фраз, содержащих числительные. 8251 фраза использовалась для обучения, 2017 — для контроля.

Меры качества. Для измерения качества использовались те же меры, что и в задаче выделения временных выражений: полнота R, точность P и F_1-мера. Также измерялась аккуратность Acc определения меток, то есть общее количество правильно определенных меток, деленное на количество данных ответов. Стоит также отметить, что рассматривались только ответы данные для грамматических меток числительных.

Результаты На рис. **??** изображена зависимость качества работы моделей от количества соседних слов q, включаемых в признаковое описание данного слова. Например, при $q = 3$ к признакам любого слова дописывались признаки 3-х левых и 3-х правых соседних слов. Как видно из рисунка, в определении грамматической формы «помогают» лишь слова из ближайшего контекста числительного. И увеличение параметра q после некоторого значения прироста качества практически не дает. Таким пороговым значением для классической модели является $q = 2$, а для модифицированной модели — $q = 7$.

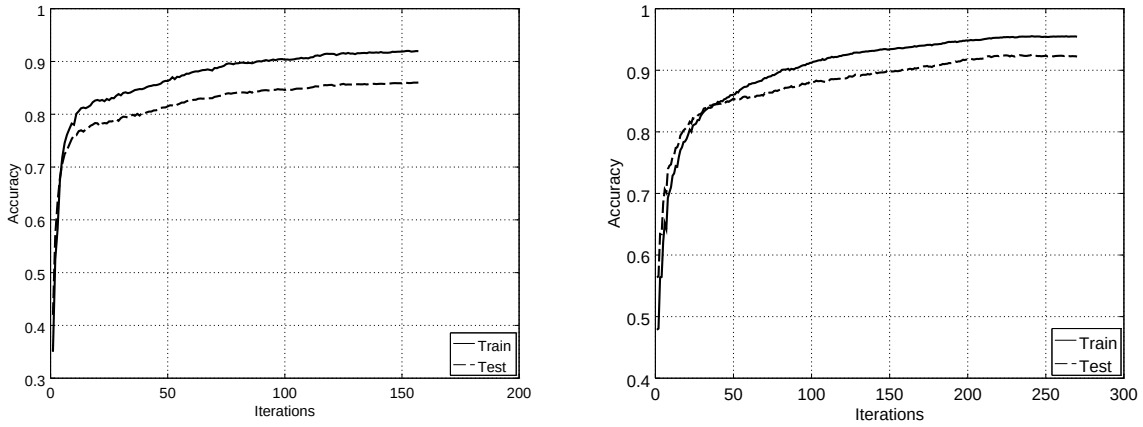
На рис. **??** изображена зависимость качества работы моделей на обучающем и контрольном множествах в течение настройки параметров моделей. Как видно из графиков, эффект переобучения имеется у обеих моделей, причем у модифицированной модели разница качества работы на обучении и контроле меньше. Отметим,



(a) Классическая модель (b) Модифицированная модель

Рис. 7: Аккуратность модели CRF в зависимости от количества соседей, включенных в признаковое описание слова

что на графиках приведены результаты при оптимальных значениях для моделей параметра q. Вообще говоря, из рис. **??**, **??** видно, что качество работы модифи-



(a) Классическая модель (b) Модифицированная модель

Рис. 8: Качество работы моделей на контрольном и обучающем множествах при настройке параметров моделей

цированной модели выше: для модифицированной модели на контрольной выборке $Acc = 92{,}39\%$, для классической модели — $Acc = 85{,}91\%$.

Также на этих же данных для модифицированной модели CRF была проведена процедура кросс-валидации (5-fold CV). Оценка скользящего контроля равна $CV = 92{,}21\%$, что говорит о высокой обобщающей способности используемого метода.

Ниже приведены значения точности P, полноты R и F_1-меры, усредненные по категориям:

Мера качества	TYPE	CASE	GEN	SNGL	ANIM
P	97,21	91,33	89,77	82,39	87,66
R	97,21	92,93	90,74	85,97	95,05
F_1	97,21	92,10	90,24	84,05	91,11

4.4 Анализ результатов

Нужно отметить, что действительное качество системы нормализации числительных будет выше. Во-первых, словесная запись некоторых числительных в разных грамматических формах совпадает, и модель CRF чаще всего ошибается именно в таких случаях (например, путает родительный и винительный падежи числительных); во-вторых, отнесение числительного к определенному роду, числу или одушевленности, когда эта информация не имеет значения (стоит метка UNKNOWN), ошибкой можно не считать. С учетом второго фактора качество работы алгоритма повышается до 94,53%.

С другой стороны, что употребление в тексте числительного в цифровом или словесном представлении зависит от многих факторов: тип числительного, вид текста, манера изложения автора и т. д. Мы же для получения обучающих и тестовых данных использовали преимущественно числительные, записанные словами. Поэтому вопрос о качестве обучающих данных, а значит и качестве полученной модели, остается открытым.

5 Заключение

В данной работе были описаны два примера задач обработки естественного языка: выделение временных выражений и нормализация цифровой записи числительных. Эти задачи сводятся к задаче сегментации и разметки последовательности. Широко распространенным статистическим методом решения таких задач является линейная модель CRF. В работе дано описание общей модели CRF, описаны детали использования и настройки параметров линейной модели CRF, а также предложена модификация линейной модели CRF, подходящая для некоторых задач разметки последовательности.

На примере задачи выделения временных выражений показано, что линейная модель CRF показывает лучшее качество работы других методов за счет учета контекста употребления временных выражений.

На примере же задачи нормализации цифровой записи числительных проведено сравнение предлагаемой модифицированной модели и классической модели CRF. Показано, что модифицированная модель меньше переобучается и показывает лучшее качество работы на тестовых данных.

Список литературы

[1] *Sutton C., McCallum A.* An Introduction to Conditional Random Fields for Relational Learning // MIT Press, (2006)

[2] *Lafferty J., McCallum A. and Pereira F.* Conditional random fields: Probabilistic models for segmenting and labeling sequence data //Proceedings of the 18th International Conference on Machine Learning, Williamstown, Massachusetts, 2001. — Pp. 282–289.

[3] *Sha F., Pereira F.* Shallow parsing with conditional random fields //In Proceedings of HLT/NAACL, 2003. — Pp. 213–220.

[4] *Poveda J., Surdeanu M., Turmo J.* A comparison of statistical and rule-induction learners for automatic tagging of time expressions in english // In Proc. of the 14th International Symposium on Temporal Representation and Reasoning (TIME 2007, IEEE, 2007. — Pp. 141–149.

[5] *Culotta A., Wick M., Hall R., McCallum A.*First-Order Probabilistic Models for Coreference Resolution In Proc. of HLT-NAACL, 2007.

[6] *Abney S.* Parsing by chunks //In Berwick R., Abney S., and Tenny C., editors, Principle-based Parsing,Kluwer Academic Publishers, 1991. — Pp. 257–279.

[7] *Sutton C., McCallum A.* Introduction to Conditional Random Fields for Relational Learning /Introduction to Statistical Relational Learning / Ed. by L. Getoor, B. Taskar. — MIT Press, 2006

[8] *Pustejovsky J., Ingria R., Sauri R. et al.* Timeml: Robust specification of event and temporal expressions in text // in Fifth International Workshop on Computational Semantics (IWCS-5) — 2003.

[9] *Hammersley J. M., Clifford P.* Markov fields on finite graphs and lattices // Unpublished, 1971.

[10] *Goodman J.* Exponential priors for maximum entropy models. // In Proc. of the HLT conference, NAACL, 2004.

[11] *Bertsekas D. P.* Nonlinear programming // Athena Scientific, 2nd edition, 1999.

[12] *Byrd R. H., Nocedal J., Schnabel R. B.* Representations of quasi-Newton matrices and their use in limited memory methods // Math. Program., 63(2):129-156, 1994.

[13] *Rabiner L. R.* A tutorial on hidden Markov models and selected applications in speech recognition // Proceedings of the IEEE., 77(2):257-286, 1989.

[14] *Koller D., Friedman N.* Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning // The MIT Press, 2009.

[15] *Ramshaw L. A., Marcus M. P.* Text chunking using transformation-based learning // The Third Workshop on Very Large Corpora, Pp. 82–94, 1995.

[16] *Reeves R. M, Ong F. R, Matheny M. E.* Detecting temporal expressions in medical narratives. // I. J. Medical Informatics. — 2013. — Vol. 82, no. 2, Pp. 118–127.

[17] *Breiman L.* Random forests //Mach. Learn. — 2001, oct. — Vol. 45, no. 1. — Pp. 5–32.

[18] *Bottou, L.* Une approche theorique de l'apprentissage connexionniste: Applications a la reconnaissance de la parole. Doctoral dissertation, Universite de Paris XI, 1991

[19] *Грановский Д. В., Бочаров В. В., Бичинева С. В.* Открытый корпус: принципы работы и перспективы, 2010

[20] *Антонова А. Ю., Соловьев А. Н.* Использование метода условных случайных полей для обработки текстов на русском языке //Компьютерная лингвистика и интеллектуальные технологии, 2013.

[21] *Muller T., Schmid H., Schutze H.* Efficient Higher-Order CRFs for Morphological Tagging. // In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, p. 322–332.

[22] *Sproat R. et al* Normalization of Non-Standard Words // WS'99 Final Report, 1999

[23] *Olinsky C., Black A. W.* Non-standard word and homograph resolution for asian language text analysis. // INTERSPEECH, ISCA, p. 733–736, 1999

[24] *Sproat R.* Lightly supervised learning of text normalization: Russian number names // SLT, IEEE, p. 436–441, 2010