Использование марковской модели максимальной энтропии для задачи извлечения собственных имен из текста

© М.А. Глазова

Санкт-Петербургский государственный университет
fruindic@inbox.ru

Аннотация

Рассматривается решение задачи извлечения собственных имен из английских текстов. Для решения задачи выбрана марковская модель максимальной энтропии. Этот документ включает в себя описание характеристических функций, сформулированных для используемого метода. Приведены результаты экспериментов по использованию различных словарей и правил для автоматической разметки обучающего множества.

1 Введение

В настоящее время основная масса информации хранится и обрабатывается в электронном виде. Практика показывает, что большинство деловых поисковых задач в интернете в той или иной степени связано с поиском имен собственных: имен людей, названий организаций и географических объектов и т. п. Правильно выделять и распознавать собственные имена необходимо и при компьютерном анализе текстов, решении аналитических задач из области компьютерной разведки. К тому же, задача извлечения собственных имен из текста является критически важной технологией для создания вопросно-ответных систем, систем информационного поиска и понимания документов.

Можно выделить два основных подхода, которые применяются при решении задачи извлечения собственных имен из текста:

- методы с использованием словарей и правил (метод регулярных выражений; метод опорных векторов (SVM));
- методы, основанные на статистике встречаемости искомых слов (скрытые марковские модели (HMM); условные случайные поля; марковские модели максимальной энтропии (MEMM)).

2 Постановка задачи

Задача состоит в следующем: изучить возможность применения метода марковской модели максимальной энтропии к задаче извлечения собственных имен из текста, а также сформулировать набор характеристических функций для рассматриваемой задачи. Кроме того, необходимо получить результаты применения данной методики на некотором множестве английских текстов и исследовать значимость каждой из сформулированных функций. Так как исследуемый метод относится к так называемым методам машинного обучения, то возникает вспомогательная задача – разметка обучающего множества. В рамках данной работы также ставилась задача исследования возможности проведения автоматической разметки обучающего множества с использованием различных словарей.

3 Обзор литературы

Упомянутые выше методы подробно рассмотрены в литературе. Приведем здесь лишь краткий обзор.

В качестве примера применения метода регулярных выражений к поставленной задаче можно рассмотреть программную систему Inex [6]. В этой статье описывается способ построения регулярных выражений для извлечения информации из текстов. Для этого используются знания о структуре текста и особенностях построения в нем предложений. Результаты, приведенные в указанной статье, показывают, что применение данного метода неэффективно для текстов, структура которых не известна заранее. Также метод неприменим в случае, когда нет четко поставленной задачи (например, найти победителя соревнований по легкой атлетике). В такой ситуации трудно составить регулярное выражение, которое охватит все возможные варианты решений.

В то же время, очевидно, что данная модель весьма эффективна при условии, что структура построения текстов известна и задача четко сформулирована: при поиске решений мы четко представляем себе, какие данные нам необходимы.

В случае, когда нет данных о структуре текста или запрос сформулирован недостаточно четко, используют метод опорных векторов. Это метод машинного обучения, который в отличие от метода регулярных выражений требует предварительного обучения на базе некоторого тренировочного набора размеченных текстов. И именно это делает метод

Труды 12[ой] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2010, Казань, Россия, 2010

более гибким, настраиваемым на различные приложения. В статье [7] описаны алгоритм применения данного метода для выделения определений из текстов, а также способ нахождения текстов, релевантных запросу.

Для нашей задачи интересен случай выделения определений из текста. Как и в предыдущей статье, в которой приведен метод регулярных выражений, в основе метода опорных векторов лежит набор правил для анализа текста. Только в этом случае правила менее зависимы от структуры текстов из рассматриваемого набора и более ориентированы на особенности языка, на котором данные тексты написаны. Это свойство делает метод переносимым между коллекциями.

Кроме того, возможно повышение точности работы метода опорных векторов с помощью использования алгоритма leave one out cross-validation (LOO CV) [8]. Для этого проводятся повторное рассмотрение тестового множества и проверка уже найденных определений (релевантных запросу текстов) с учетом уточненных данных о структуре текста.

Полу-условные случайные поля рассматриваются в [9]. В данной статье исследуется задача сегментирования текста с учетом строго заданного порядка следования сегментов. Сегментирование – это разбиение текста на сегменты (отдельные части текста, характеризующиеся каким-либо определенным свойством). Самый простой пример сегментирования текста – выделение в тексте сегментов: Заголовок, Автор (Авторы), Введение, Основная часть, Заключение. Если в качестве исходного текста рассматривать каждое предложение в отдельности, а выделяемыми сегментами выбрать «Собственное имя» и «не Собственное имя», то полу-условные случайные поля можно использовать для решения задачи извлечения собственных имен из текста. Это возможно при наличии строго заданного алгоритма построения предложений в тексте, например, если нам известно, что после СИ обязательно должен следовать глагол, и т. п.

Это вероятностная модель, основанная на цепях Маркова (математический аппарат, позволяющий с учетом текущего состояния системы вычислить вероятность попадания системы на следующем шаге в одно из возможных состояний). В статье состояниями являются сегменты, вся система – текст, а состояние системы на текущем шаге – токен. Вероятность попадания токена в сегмент вычисляется на основе всех имеющихся данных о парах токен – сегмент. При этом последовательность смены сегментов строго задана, что облегчает алгоритм, так как возможные конечные состояния строго заданы.

Модель эффективна для задачи сегментирования текстов, но не всегда пригодна для поиска данных в тексте. Например, необходимо найти такие данные, которые могут встречаться в любой части текста, и невозможно задать точную схему предложений, в которых эти данные могут встретиться.

Скрытые марковские модели, описанные в [10], используют текущее состояние системы (в нашей задаче возможны всего 2 состояния: «собственное имя» и «несобственное имя») и рассматриваемый токен для построения матрицы переходов в следующее состояние. Для этого используется набор заданных заранее характеристических функций, которые считаются взаимно независимыми. Характеристические функции – функции, зависящие от рассматриваемого токена, а также, возможно, от нескольких соседних токенов. Характеристические функции задаются экспертом на основе анализа обучающего текста с учетом его лингвистических и стилистических особенностей. Например, «наблюдаемое слово начинается с большой буквы», «перед наблюдаемым словом расположено слово из специального набора: mr., sir, president, doctor и т. д.». На основе построенной матрицы переходов прогнозируют вероятное состояние, в которое попадет система на следующем шаге. При наличии большого числа входных токенов размерность матрицы переходов значительно возрастает, что усложняет вычисления. Кроме того, этот метод не всегда точен. Он показывает хорошие результаты при выявлении наиболее вероятного состояния на основе известного предыдущего состояния. В случае же, когда нам неизвестна вся последовательность состояний, упомянутый метод не гарантирует хороших результатов.

Для последнего описанного случая (последовательность смены состояний неизвестна заранее) используется марковская модель максимальной энтропии [1]. Для определения наиболее вероятной последовательности состояний по входным токенам используется метод максимальной энтропии информации. Кроме того, учитывается возможная зависимость между характеристическими функциями. А в остальном эта модель очень схожа со скрытыми марковскими цепями. В упомянутой выше статье авторы приводят некоторые результаты по сравнению эффективности работы метода с другими существующими методами.

Условные случайные поля [11] представляют собой улучшенный вариант марковских моделей максимальной энтропии, в которых учитывается зависимость вероятности следующего состояния не только от предыдущего входящего токена, но и от всех рассматриваемых токенов, а также, по аналогии с марковскими моделями максимальной энтропии, рассматривается возможная зависимость характеристических функций между собой. Эта модель улучшает результаты, но при этом значительно возрастает сложность вычислений, следовательно, падает скорость обработки обучающего множества.

Из приведенных выше описаний методов становится ясно, что значимое влияние на сложность алгоритма оказывает число токенов, встречающихся в рассматриваемых текстах. Для некоторых задач уменьшение вычислительной нагрузки достигается за счет использования групп – объединение разных слов, похожих по смыслу. Например, в качестве

группы можно рассматривать слова «машина» и «автомобиль». Объединение в группы может применяться при кластеризации набора текстов по ключевым словам. Но, в то же время, метод неприменим в тех задачах, когда часть речи токена играет ключевую роль. Так как собственные имена относятся к именам существительным, этот метод сокращения числа токенов является непригодным для нашей задачи.

В статье [12] приводится алгоритм использования двоичных деревьев для того, чтобы слова с одной основой определялись как одинаковые или похожие слова. Это также позволяет уменьшить количество рассматриваемых токенов. Очевидно, что при больших объемах анализируемых текстов это значительно снижает вычислительные нагрузки.

Эффективный способ уменьшения количества рассматриваемых токенов – использование групп слов. Слова-синонимы объединяются в группы слов. Но иногда смысл слова можно определить только с учетом контекста, следовательно, одно и то же слово может быть отнесено к разным группам. В таких случаях необходимо провести анализ текста.

Для построения анализатора текстов представляет интерес метод стохастических грамматик. В [13] этот метод применяется для распознавания речи. Данную идею можно спроецировать на задачу распознавания смысла текста или на задачу нахождения в тексте какой-либо интересующей нас информации.

4 Описание метода

Теперь рассмотрим подробнее марковскую модель максимальной энтропии, адаптированную автором к поставленной задаче, аналогично тому, как это было сделано в [5].

Рассмотрим случайную величину X, принимающую значения x_i, $i = 1,...,n$. Пусть заданы m характеристических функций $f_k(x_i)$, $k = 1,...,m$. И пусть ограничения на функции распределения вероятностей заданы выражениями вида

$$\sum_{i=1}^{n} P(x_i \mid I)\, f_k(x_i) = F_k,\ k = 1,...,m, \qquad (1)$$

$$\sum_{i=1}^{n} P(x_i \mid I) = 1,$$

где $P(x_i \mid I)$ – вероятность попадания в состояние x_i при условии I, $f_k(x_i)$ – значение k-й характеристической функции для состояния x_i, а F_k, $k = 1,...,m$, – правая часть уравнений-ограничений, не зависящая от характеристической функции $f_k(x_i)$.

Тогда вероятностным распределением с максимальной энтропией является распределение Гиббса

$$P(x_j \mid I) = \frac{1}{Z(\lambda_1,...,\lambda_m)} \exp\left[\sum_{i=1}^{m} \lambda_i\, f_i(x_j)\right], \qquad (2)$$

где λ_i, $i = 1,...,m$, – коэффициенты при характеристических функциях, зависящие от правых частей F_k уравнения (1), а $Z(\lambda_1,...,\lambda_m)$ – нормализующий множитель.

Из заданной коллекции документов произвольным образом выделяется обучающее множество (некоторое небольшое, относительно размеров всей коллекции, количество документов, которые размечаются вручную; на основе этих данных подбираются коэффициенты для характеристических функций в функциях распределения).

Обучающее множество разбивается на токены, o_j, с которыми ведется дальнейшая работа. В нашей задаче токены – слова. Для каждого токена определяется состояние, к которому он (токен) относится. В рассматриваемой задаче используется всего 2 состояния – «собственное имя» и «не собственное имя». Таким образом, множество состояний принимает вид $S = \{0,1\}$. В некоторых случаях могут быть заданы функции запрета переходов между состояниями на каких-либо шагах. На начальном этапе также выбираются характеристические функции $f_i(s_j, o_j)$. Параметр o_j дописывается в качестве аргумента характеристической функции, чтобы дополнительно подчеркнуть зависимость характеристической функции от текущего токена.

Эти характеристические функции входят в уравнения (1), задающие ограничения на функции распределения вероятностей. Для составления этих уравнений с помощью эксперта проводятся следующие операции:

- определяется последовательность смены состояний $s_1,...,s_n$ на основе обучающего множества – для каждого токена o_i из обучающего множества эксперт определяет соответствующее ему состояние s_i;
- по принципу максимальной энтропии строятся матрицы вероятностей перехода между состояниями $P_{s'}(s \mid o)$, где $s' \in S$ – предшествующее начальное состояние системы, $s \in S$ – следующее состояние системы, o – рассматриваемый токен. В качестве ограничений (1) используется предположение, что математические ожидания значений характеристических функций на множестве всех рассматриваемых текстов будут равны средним значениям этих функций на обучающем множестве (аналогично тому, как предложено в [1]):

$$\frac{1}{m'_s}\sum_{k=1}^{m'_s} f_a(s_{t_k}, o_{t_k}) = \frac{1}{m'_s}\sum_{k=1}^{m'_s}\sum_{s \in S} P'_s(s \mid o_{t_k})\, f_a(s, o_{t_k}),$$

где m'_s – число переходов из состояния s', $t_1,...,t_{m'_s}$ – моменты времени перехода из состояния s' для обучающего множества.

Как уже упоминалось, при таких ограничениях функции распределения вероятностей имеют вид распределения Гиббса (2). Оценка параметров λ_a производится по алгоритму GIS (General Iterative Scalling) [14]. С помощью алгоритма можно итеративно определить приближенное значение параметров, входящих в вероятностную функцию перехода, заданную в выражении (2).

Необходимым условием корректной работы алгоритма является постоянное для всех рассматриваемых токенов значение суммы вероятностных функций. Для достижения этого, как правило, вводится дополнительная характеристическая функция, значение которой равно разности между максимальной суммой значений характеристических функций среди всех рассматриваемых токенов из обучающего множества и значением суммы характеристических функций для текущего токена.

Подставив полученные значения параметров в уравнение (2), получаем набор вероятностей переходов между состояниями для заданных токенов. Тогда на новом множестве токенов, т. е. на оставшемся множестве исследуемых текстов, вероятность попадания системы в состояние s на шаге t определяется по алгоритму Витерби [15], рекурсивный шаг которого задается выражением

$$\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') * P_{s'}(s \mid o_{t+1}),$$

где $\alpha_{t+1}(s)$ – вероятность того, что $(t+1)$-й токен попадает в состояние s'. Это значение вычисляется для всех возможных состояний $s \in S$ и выбирается максимальное значение.

Следует отметить, что в приведенной формуле учитываются не только предыдущее или текущее состояние системы, но и вся предыдущая цепочка состояний, так как для вычисления вероятности последующего состояния мы используем вероятность текущего состояния, которая вычислена по этой же формуле.

5 Описание характеристических функций

В работе рассматривается применение марковской модели максимальной энтропии к задаче извлечения собственных имен из текста. На основе анализа специфики построения английских текстов для решения рассматриваемой задачи было сформулировано 10 характеристических функций. Для пяти из них в [5] автором уже была проанализирована эффективность применения выбранной модели. Используемые характеристические функции таковы.

Необходимое условие. Слово начинается с большой буквы и является существительным. Это наиболее очевидная характеристическая функция, которая, за исключением опечаток, является необходимым условием того, что рассматриваемое слово является собственным именем.

Наличие аббревиатуры. Предыдущее слово содержит аббревиатуру и рассматриваемое слово начинается с большой буквы. В данном случае под аббревиатурой понимается несколько заглавных букв подряд, разделенных точкой. Такая комбинация символов встречается при наличии инициалов и фамилии в тексте, когда инициалы предшествуют фамилии. При этом наличие точек между заглавными буквами инициалов обязательно. Это исключает возможность отнесения к собственным именам общепринятых сокращений, употребляемых для уточнения типа юридического лица и т. п.

Последователь собственного имени. Предшествующее слово относится к классу собственных имен и после него следует запятая. Если рассматриваемое слово начинается с большой буквы, то, вероятнее всего, оно будет также относиться к классу собственных имен.

Наличие апострофа. Рассматриваемое слово начинается с большой буквы и в нем содержится символ «’». Исходя из особенностей грамматики английского языка, апостроф в конце слова означает принадлежность одушевленному предмету. Это позволяет предположить, что рассматриваемое слово – собственное имя. Если же апостроф содержится в начале слова, то его наличие также, зачастую, является признаком собственного имени.

Функция частоты. Частота встречи слова во множестве документов менее 20. Данная функция используется для того, чтобы избежать рассмотрения возможных ложных собственных имен и отсеять наиболее часто встречающиеся слова, так как это, скорее всего, распространенные глаголы и предлоги.

Специальные предшественники. У рассматриваемого слова есть предшественник из специального списка предшественников (например, mr., sir., president, chairman и т. д.) и само слово начинается с большой буквы. Данный список был сформирован на основе анализа некоторых текстов из коллекции и может быть дополнен при более детальном рассмотрении данного вопроса.

Перечисление собственных имен. Предыдущие два слова – слово с большой буквы и «and». Если рассматриваемое слово также начинается с большой буквы, то оно может рассматриваться как кандидат в собственные имена. Эта характеристическая функция схожа с той, где вместо слова «and» рассматривалось наличие символа « , ». Ее важная особенность заключается в том, что для анализа необходимо использовать не только одно слово-предшественник, а сразу два. Это повышает точность исследования.

Функция длины слова. Длина рассматриваемого слова находится в пределах [3;20] символов. Границы установлены автором условно, так как при длине слова, меньшей 3, велика вероятность того, что это предлог или вспомогательное слово. Если длина превосходит 20 символов, то рассматриваемое слово, скорее всего, является сложным – со-

ставным, следовательно, маловероятно, что оно относится к собственным именам.

Комбинация с несобственным существительным. Предыдущие слова – существительное или местоимение и слово «and». Если рассматриваемое слово начинается с большой буквы, то его относим к кандидатам в собственные имена. В данном случае рассматривается не только два предшествующих слова, что уже было отражено в характеристических функциях данного набора. Здесь учтены части речи рассматриваемых слов, а также устранено требование наличия заглавной буквы в слове-предшественнике.

Последующий глагол. Рассматриваемое слово начинается с большой буквы, а следом за ним в предложении (тексте) расположены глагол или какая-нибудь отглагольная форма, отражающая действие. Исходя из особенностей английского языка, глагол, как правило, следует за существительным, и если это существительное начинается с большой буквы, то это может быть и собственное имя. Особенность этой характеристической функции в том, что она использует не предыдущие слова, а последующее, а также принимает во внимание не только само слово, но и его часть речи.

В сформулированных характеристических функциях учитываются два слова предшествующие исследуемому токену и одно слово следующее за непосредственно за ним. На значение характеристических функций влияют части речи этих слов, а также результат их сравнения со специальным набором слов-предшественников. Кроме того, учитываются длина рассматриваемого слова, частота его встреч во всей коллекции, а также наличие знаков препинания между словами.

6 Описание экспериментов

Для проведения экспериментов использовалась коллекция английских текстов Reuters21578 [2]. Это архив новостных публикаций, распространенных агенством Reuters в 1987 году.

В качестве обучающего множества выбиралась часть документов из вышеназванной коллекции. Первоначально разметка обучающего множества проводилась вручную. Остальные документы из рассматриваемого набора использовались в качестве тестового множества. Полученные результаты сравнивались с данными, полученными с помощью ручного анализа текстов.

При анализе текстов использовался анализатор грамматики treeTagger [3]. С его помощью для каждого слова определялись часть речи, к которой оно относится, и вероятная основа. В упомянутой программе для выделения основы слова и определения его части речи используется двоичное дерево.

При увеличении объема обучающего множества временные затраты на ручную разметку значительно увеличились. В связи с этим возникла вспомогательная задача – автоматической разметки текстов. Для ее решения используются словари заведомо известных собственных имен и лингвистические правила, характерные для языка, на котором написаны тексты коллекции. В данном случае – это английский язык. В качестве словарей использовались множества собственных имен из Wikipedia [4], а также список фамилий, содержащийся в самой коллекции Reuters21578 [2]. Так как в коллекции содержится лишь файл с фамилиями, а нас интересовали также и имена персон, то имена в данном случае мы взяли из первого списка. Таким образом, данный словарь можно считать синтезированным.

Введем некоторые обозначения.

Обозначим через CNE количество слов, верно выделенных из текста в результате разметки собственных имен, ENE – количество «кандидатов» – слов, помеченных как собственные имена при использовании метода, TNE – действительное количество собственных имен в тексте.

Качество полученных результатов будем оценивать следующими характеристиками:

- точность определения класса (NEP – name entity precision) – отношение числа правильно выделенных элементов класса к общему числу выделенных элементов этого класса

$$NEP = \frac{CNE}{ENE};$$

- полнота определения класса (NER – name entity recall) – отношение числа правильно определенных элементов класса к общему числу элементов данного класса в тексте

$$NER = \frac{CNE}{TNE}.$$

Кроме того, на основе значений точности и полноты определения собственных имен вычисляется дополнительная метрика – F-мера :

$$F\text{-мера} = \frac{2 \cdot NEP \cdot NER}{NEP + NER}$$

для сравнения значений, полученных для разных множеств.

7 Результаты экспериментов

Как уже упоминалось ранее, в [5] были проведены исследования коэффициентов, полученных для пяти из приведенных здесь характеристических функций (необходимое условие, функция частоты, функция длины слова, наличие аббревиатуры и специальные предшественники). При анализе полученных коэффициентов было замечено, что коэффициенты в формуле распределения вероятностей для функций «необходимое условие» и «наличие аббревиатуры» в 1,5 раза больше, чем для других характеристических функций, что говорит о большей зависимости вероятности от этих функций по сравнению с остальными. В исследованном частном случае наименьший коэффициент соответствовал функции длины слова. Но из этого не следует, что данная функция будет слабо влиять на вероятность токена быть собственным именем в другом наборе

текстов или для другого множества характеристических функций.

При проведении аналогичных исследований, но уже для набора из 10 характеристических функций, были получены результаты, представленные в табл. 1. В ней для краткости наборы содержат порядковые номера входящих в них характеристических функций.

Величина F-меры является показателем качества метода: чем больше ее значение, тем точнее определяются классы.

Из табл. 1 видно, что F-мера принимает относительно большие значения для наборов характеристических функций (1-9); (1-6, 8-10); (1-3,5-10); (1-2,4-10). Наибольшие значения мера достигает для наборов из девяти функций, причем значения для этих наборов близки между собой. Из этого можно сделать вывод о том, что девять из десяти сформулированных функций являются значимыми для решения поставленной задачи.

Таблица 1. Значение точности, полноты и F-меры для разных наборов характеристических функций. Объем обучающего множества составляет 10 % от исследуемого

Наборы функций\ характеристики	F-мера	*NEP*	*NER*
1-10	0,11702	0,11	0,125
1-9	0,2	0,5	0,125
1-8, 10	0,0001	0,0001	0,0001
1-7, 9-10	0,06140	0,035	0,25
1-6, 8-10	0,22222	1	0,125
1-5, 7-10	0,08333	0,05	0,25
1-4, 6-10	0,05801	0,03	0,875
1-3, 5-10	0,2	0,5	0,125
1-2, 4-10	0,22222	1	0,125
1, 3-10	0,11758	0,111	0,125
2-10	0,08333	0,05	0,25
1,5,9	0,05468	0,035	0,125

Для рассматриваемых текстов и введенных характеристических функций наиболее точно классы определяются для наборов из девяти функций, полученных исключением из полного набора 3, 4, 7 или 10 функции. Это показывает, что при определении классов выделенные функции на данном наборе текстов не несут новой полезной информации, а лишь «загрязняют» полученные результаты – понижают точность при неизменном значении полноты, т. е. при их использовании выделяется большее количество токенов в качестве собственных имен, при этом относительное количество верно определенных представителей класса собственных имен уменьшается.

С точки зрения поставленной задачи – извлечения собственных имен из текста (их в текстах, как правило, гораздо меньше, чем несобственных) – интерес представляют точность и полнота определения не всех рассматриваемых классов, а именно класса собственных имен, причем интерес представляет не отдельная характеристика (точность или полнота) сама по себе, а их совокупность.

Наибольшие значения точность выделения «собственных имен» достигает для наборов из девяти функций, полученных исключением из полного набора 3, 4, 7 или 10 функции. А полнота определения «собственных имен» достигает наибольшего значения для набора без 5 функций.

Все приведенные выше данные относятся к тому случаю, когда вычисления проводились для начальных коэффициентов алгоритма GIS, выбранных специальным образом: сначала вычислялись приближенные значения коэффициентов для одинаковых начальных значений $\lambda_i^{(0)} = 1$. Новое начальное значение коэффициента $\lambda_i^{(0)}$ полагается равным вычисленному приближению, разделенному на 10.

Данные результаты были получены для обучающего множества, состоящего из 10% от рассматриваемого множества текстовых документов. (для этого исследования были взяты первые 1000 документов коллекции, обучающее множество в данном случае состояло из 100 документов). Исследования также проводились для обучающих множеств, состоящих из 30% и 50% от всего множества текстов (обучающее множество состояло из 300 и 500 документов соответственно). Приведем значения рассматриваемых характеристик для найденных «оптимальных» наборов характеристических функций (см. таблицы 2 и 3).

Таблица 2. Значение точности, полноты и F-меры для разных наборов характеристических функций. Объем обучающего множества составляет 30 % от исследуемого

Наборы функций\ характеристики	F-мера	*NEP*	*NER*
1-10	0,05797	0,0327	0,25
1-9	0,09999	0,0833	0,125
1-6, 8-10	0,10526	0,0909	0,125
1-3, 5-10	0,10526	0,0909	0,125
1-2, 4-10	0,04651	0,0285	0,125

Таблица 3. Значение точности, полноты и F-меры для разных наборов характеристических функций. Объем обучающего множества составляет 50 % от исследуемого

Наборы функций\ характеристики	F-мера	*NEP*	*NER*
1-10	0,04419	0,0233	0,4
1-9	0,05128	0,0280	0,3
1-6, 8-10	0,08510	0,0540	0,2
1-3, 5-10	0,08333	0,0526	0,2
1-2, 4-10	0,05172	0,0283	0,3

Для таких начальных данных наибольшее значение F-меры принимает для наборов характеристических функций (1-6, 8-10) и (1-3, 5-10), причем для этих же наборов достигает свои наибольшие значения и точность определения собственных имен.

Значит, можно считать, что для точности класса наименее значимыми являются функции 4 и 7.

Легко заметить, что с увеличением обучающего множества растет полнота определения собственных имен, но при этом падает точность. Это происходит в том случае, когда в качестве собственных имен выбирается большое количество токенов. Значит, рассматриваемый метод не является устойчивым для сформулированных характеристических функций. Из приведенных таблиц можно сделать вывод, что для исследуемого набора характеристических функций и рассматриваемого множества текстов при увеличении размера обучающего множества уменьшаются точность определения «собственных имен» и значение F-меры. Этот факт расходится с ожидаемыми результатами и требует дополнительного исследования.

Проанализировав полученные значения, можем сделать следующие выводы:

- при специфическом выборе начальных значений для коэффициентов в упомянутом алгоритме значимыми на описанном наборе текстов являются девять характеристических функций;
- существенными для достижения точности определения класса собственных имен являются функции 1, 2, 3, 5, 6, 8, 9, 10;
- с увеличением обучающего множества метод расходится; необходимо найти такую характеристическую функцию для уже имеющегося набора, которая обеспечит сходимость метода, а также подобрать такое оптимальное по размеру обучающее множество, чтобы исключить возможность возникновения переобучения метода;
- при увеличении размера обучающего множества уменьшаются значение точности определения «собственных имен» и значение F-меры, следовательно, необходимо исследовать возможное переобучение метода и проследить за динамикой соответствующих показателей при ограничении размеров словаря;
- в то же время, полнота выделения «собственных имен» возрастает с увеличением тестового множества.

Теперь рассмотрим вспомогательную задачу – автоматическую разметку текстов с помощью различных словарей. Для этого исследования также были взяты первые 1000 документов коллекции. В качестве словарей использовались множества собственных имен из Wikipedia [4], а также список фамилий, содержащийся в самой коллекции Reuters21578 [2]. Так как в коллекции содержится лишь файл с фамилиями, а нас интересовали также и имена персон, то имена в данном случае мы взяли из первого списка. Таким образом, данный словарь можно считать синтезированным.

Кроме того, интересно посмотреть на результаты, полученные только с использованием списка фамилий из коллекции Reuters21578 [2].

В ходе проведения эксперимента ставились задачи:

- оценить эффективность применения каждого из используемых словарей;
- сравнить результаты, полученные при использовании выбранных множеств собственных имен.

Результаты, полученные во время исследования, приведены в таблицах 4 и 5.

Анализируя полученные данные, можно сделать следующие выводы:

- с точки зрения абсолютных показателей большую эффективность дает применение словаря Wikipedia [4], но следует отметить, что при этом количество выделенных кандидатов практически в два раза больше;
- из сравнения характеристических параметров можно сделать вывод, что применение словаря коллекции Reuters21578 [2] дает несколько более хорошие результаты, чем применение словаря Wikipedia [4], что, в принципе, совпадает с ожидаемым результатом;
- среди невыделенных собственных имен следует отметить наличие арабских и японских имен, которые не содержались в используемых словарях, что ухудшило показатели;
- кроме того, добиться улучшения качества разметки можно, учитывая при разметке некоторые лингвистические правила английского языка (например, добавление 's);
- было выделено много «лишних» слов: артикли, предлоги, заглавные буквы – инициалы из словаря; это объясняется тем, что словарь из Wikipedia [4] содержит в себе такие имена, как Alexander of Makedonia; каждое слово из имени словаря считалось как отдельное собственное имя.

Таблица 4. Абсолютные значения, полученные из эксперимента

	ENE	CNE	Unique CNE	Not found NE	Unique not found NE
Список имен из Wikipedia	969	174	96	161	79
Список имен из Reuters21578	572	128	71	207	104

Таблица 5. Значения характеристических параметров

	NEP	*NER*	F-мера
CNE Wikipedia	0,17957	0,5223	0,26725
Unique CNE Wikipedia	0,09907	0,5485	0,16782
CNE Reuters21578	0,22377	0,3820	0,2822
Unique CNE Reuters21578	0,1241	0,4057	0,1900

Интересные результаты получены для разметки документов только с использованием фамилий из

коллекции Reuters21578 [2]. Для такого словаря выделено всего 44 CNE, что, конечно же, мало в сравнении с общим числом собственных имен в тексте.

Следует отметить, что среди выделенных «кандидатов» не содержится ни одного ложного «кандидата». Если вычислять значения характеристических параметров, рассматривая в качестве искомых собственных имен только фамилии (их всего 119), то получаются следующие результаты: NEP=1, NER= 0,4033, F-мера = 0,57478.

Далее были предприняты попытки по очистке словарей от «лишних» слов. Также добавлено правило с апострофом. Полученные результаты представлены в таблицах 6 и 7.

Таблица 6. Абсолютные значения после обработки словаря

	ENE	CNE	Unique CNE	Not found NE	Unique not found NE
Список имен из Wikipedia	592	181	97	154	78

Таблица 7. Значения характеристических параметров

	NEP		*F*-мера
CNE Wikipedia	0,30574	0,54029	0,3905
Unique CNE Wikipedia	0,16385	0,5543	0,25293

Сравнивая результаты, полученные в рассмотренных случаях автоматической разметки текстов, замечаем, что:

- сравнение только со словарем дает не очень хорошие результаты, эффективнее проводить такое сравнение с дополнительным использованием лексических правил соответствующего языка;
- плохо выделяются арабские и японские имена.

Необходимо пополнить используемый словарь списком специфических имен. Основные категории ошибочных кандидатов:

- названия месяцев (May, August, April);
- сторон света (West, North);
- стран (German, France), городов (London, New York) и пр. географических объектов (Victoria);
- цвета (Black, White).

Литература

[1] McCallum A., Freitag D., Pereira F. Maximum entropy Markov models for information extraction and Segmentation. – http://www.ai.mit.edu/courses/6.891-nlp/READINGS/maxent.pdf.

[2] LANGREITER.COM plain, simple.

[3] www.ims.uni-stuttgart.de/projekte/corplex/ TreeTagger.

[4] Wikipedia. The free encyclopedia. – http://en. wikipedia.org.

[5] Глазова М.А. Извлечение собственных имен из текста с помощью метода максимальной энтропии, основанного на цепях Маркова // Процессы управления и устойчивость: Труды 40-й межд. науч. конф. аспирантов и студентов / Под ред. Н.В. Смирнова, Г.Ш. Тамасяна. – СПб.: Изд. Дом С.-Петерб. гос. ун-та, 2009. – С. 402-407.

[6] Программная система извлечения информации из текстов (ПС INEX). – http://www.skif.pereslavl.ru/psi-info/airec/airec-ppt.rus/inex.ppt

[7] Li Hang. Learning to rank: a new technology for text processing. – http://www-tsujii.is.s.u-tokyo.ac.jp/T-FaNT/T-FaNT.files/Slides/Li.pdf.

[8] Yu Shui, Song Hui, Ma FanYuan. Novel SVM performance estimators for information retrieval systems. – Department of Computer Science and Technology, Shanghai Jiaotong University, Shanghai, 2000.

[9] Wu Xiaofeng, Zong Chengqing. A new approach to automatic document summarization//Proc. of the Third Int. Joint Conf. on Natural Language Processing, 2008.

[10] Blunsom Ph. Hidden Markov models. – University of Melbourne, Faculty of Engineering, Human Language Technology, August 2004. – P. 433-460

[11] Lafferty J., McCallum A., Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. – www.cis.upenn.edu/pereira/papers/crf.pdf.

[12] Haffari G., Whye Y. The hierarchical Dirichlet trees for information retrieval. – www.aclweb.org/anthology/N/N09/N09-1020.pdf.

[13] De Mori R., Kuhn R. Some results on stochastic language modelling. – www.aclweb.org/anthology/H/H91/H91-1043.pdf.

[14] Wang Shaojun, Schuurmans D., Peng F., Zhao Y. Combining statistical language models via the latent maximum entropy principle. – http://www.springerlink.com/content/m7n3w402270613t5/ fulltext.pdf.

[15] Компьютерное распознавание и порождение речи. – http://speech-text.narod.ru/chap4_2_2.html.

Using of maximum entropy Markov model for the problem of extracting name entities from English texts

M. Glazova

In the article is presented the problem of extracting name entities from English texts. Markov model of maximum entropy is selected to solve the problem. This document includes a description of set of characteristic functions, which were formulated to use in the method. Also experiments for the automatic partitioning of the training set are described. In the paper you can find results and conclusions from experiments based on the use of different vocabularies and rules.