САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра системного программирования

Метод выделения именованных сущностей на основе Википедии

Дипломная работа студента 545 группы
Ткаченко Максима Владиславовича

Научный руководитель ..... д.ф.-м.н., проф. Новиков Б.А.
/подпись/

Рецензент ..... к.ф.-м.н., доц. Барашев Д.В.
/подпись/

“Допустить к защите” ..... д.ф.-м.н., проф. Терехов А.Н.
заведующий кафедрой /подпись/

Санкт-Петербург
2011

SAINT PETERSBURG STATE UNIVERSITY
Mathematics and Mechanics Faculty

Software Engineering Department

Named Entity Recognition Technique Based on Wikipedia Corpus

Graduate paper by
Maksim Tkachenko

Scientific advisor ..... Professor B.A. Novikov

Reviewer ..... Associated Professor D.V. Barashev

"Approved by" ..... Professor A.N. Terekhov
Head of Department

Saint Petersburg
2011

Содержание

Введение

Постановка задачи

Выделение именованных сущностей (named entity recognition) - одна из ключевых задач извлечения информации, извлечения структурированных данных из неструктурированных документов. Ее суть - найти в тексте названия, идентификаторы объектов определенного типа, это могут быть фестивали, продукты, биологические объекты, протеины и т.д. Впервые задача была сформулирована еще в 1996 году на Message Understanding Conference, где в качестве сущностей рассматривались: организации, места, люди и некоторые числовые выражения. Минуло 15 лет, а интерес к проблеме не угасает и по сей день - как с академической стороны, так и со стороны индустрии, появляются новые классы, новые домены и т.д.

Под термином *именованная сущность*[1] (named entity) мы будем понимать объект определенного типа, имеющий имя, название или идентификатор. Какие типы выделяет система, определяется в рамках конкретной задачи. Для новостного домена обычно это *люди* (PER), *места* (LOC), *организации* (ORG) и *разное*, объекты широкого спектра: события, слоганы и т.д. В общем случае на вход системе поступает текст, на выходе система сообщает информацию о положении имен в тексте и информацию о классах, которые им соответствуют. Заметим, что набор классов мы фиксируем заранее. Приведем пример размеченного текста:

Born in [LOC Moscow], [LOC Russia], [PER Pushkin] published...

К настоящему моменту существует множество стратегий и подходов к решению поставленной задачи. Особенно популярны методы машинного обучения с учителем. Такая система изначально тренируется на множестве размеченных примеров, натренированную модель теперь можно использовать на произвольных данных. Так как тренировочная коллекция, обычно, создается вручную и имеет ограниченный объем, в некоторой степени справедливо, что успех обучающихся систем выделения сущностей зависит от наличия дополнительных ресурсов: словарей сущностей, дополнительных тренировочных наборов, коллекций неразмеченных текстов и т.д. В данной работе в качестве такого ресурса исследуется Википедия[2].

Википедия - это свободная общедоступная мультиязычная интернет-энциклопедия. Ее статьи покрывают огромное множество предметных областей и, как правило, создаются и постоянно дорабатываются вручную добровольцами со всего мира. По сравнению

[1] Далее, мы будем часто опускать слово “именованная” и иногда отождествлять сущность с ее именем.
[2] http://www.wikipedia.org/

с простыми текстами Википедия имеет хорошо определенную структуру, что делает ее привлекательной для исследователей в таких областях, как обработка естественных языков, информационный поиск и т.д.

Данная работа нацелена на создание нового подхода к использованию Википедии в задаче выделения именованных сущностей, который позволил бы существенно повысить качество подобных систем. В рамках работы предлагается решить следующие задачи:

- Предложить и проанализировать метод использования Википедии в системах выделения сущностей.
- Создать и оценить систему выделения сущностей по классам *люди*, *места*, *организации* и *разное*.

Мотивация

Ответим на вопрос: “Зачем выделять именованные сущности?” Как уже упоминалось ранее, выделение сущностей - один из ключевых этапов предобработки текста для построения более сложных приложений извлечения информации.

Рассмотрим задачу идентификации событий. Задача состоит в извлечении сущностей, описывающих время, в согласовании их с другими сущностями. Рассмотрим в качестве примера событие “конференция”. Его можно описать четырьмя сущностями: название конференции (International Conference on Data Engineering), место проведения (Hannover), дата начала конференции (April 11, 2011) и дата окончания (April 16, 2011). Описание события можно усложнять, добавив, например, названия секций, имена председателей и т.д.

Часто компонента выделения сущностей входит в состав вопросно-ответных систем. Было замечено, что большая часть вопросов подразумевает именованную сущность определенного типа в качестве ответа; например, вопрос “кто?” часто предполагает имя человека, “где?” - определенное место.

Выделение сущностей может быть полезно и в информационном поиске. Известно, что слово “Jaguar” может трактоваться в различных контекстах как название компании, музыкальной группы, марки машин или просто название живого существа. Если соответствующая система информационного поиска знает, о каком из значений слова “Jaguar” идет речь в том или ином документе, то в зависимости от запроса выдача страниц может быть ограничена только определенным типом сущностей.

Предварительное выделение сущностей из огромной коллекции документов позволяет улучшить их организацию и дать первое представление о содержании документа: о каких компаниях идет речь в том или ином договоре, какие протеины упомянуты в той или иной биологической статье.

Все перечисленное делает задачу о выделении именованных сущностей достаточно полезной и интересной не только с точки зрения исследователей, но и с точки зрения компаний, занимающихся интеллектуальным анализом данных, информационным поиском и т.д.

1 Обзор предметной области

1.1 Выделение именованных сущностей

Одной из первых работ в данной области принято считать статью Лизы Рау [32] (1991). Она предложила использовать эвристические подходы и набор правил[1] для выделения названий компаний в тексте. С тех пор за 20 лет исследований было предложено огромное количество решений и стратегий по извлечению имен. Задача была представлена не секциях различных конференций: Message Understanding Conference (MUC) [17], Conference on Natural Language Learning (CoNLL) [38], International Conference on Language Resources and Evaluation[2] (LERC).

Хорошая обзорная работа была проделана Дэвидом Надю и Сатоши Секином [29]. Авторы провели подробный обзор методов, используемых в области выделения и классификации сущностей, за период с 1991 по 2006 года. В данной же работе мы ограничимся основными моментами, необходимыми для общего понимания подходов к решению задачи, и постараемся дополнить упомянутый обзор.

1.1.1 Влияние предметной области. Классы сущностей

Предметная область и жанр текстов имеют сильное влияние на систему выделения сущностей в целом. Проектирование методов, устойчивых к смене домена обрабатываемых текстов, остается трудной задачей [29, 2]. В особенности это влияет на системы, основанные на правилах. Изменение предметной области, как правило, влечет ухудшение от 10% до 40% в точности и полноте систем. Однако, Дауме Хал III в своей работе [11] предложил метод, который повышает устойчивость статистического аннотатора к изменению домена, и продемонстрировал его работоспособность в частности на задаче выделения сущностей.

Для разных предметных областей свойственны различные классы именованных сущностей. Для биологической литературы это могут быть названия протеинов, для государственный отчетов - названия актов и различных учреждений. Часто в качестве сущностей рассматривают имена людей, названия организаций и мест [17, 38]. Однако, этим набор классов не ограничивается; тип *разное* (MISC) был введен на CoNLL и включил в себя широкий набор сущностей: наименования национальностей (Russian, Chinese), названия событий (Language Resources and Evaluation Conference) и т.п. Дополнительное дробление вышеуказанных классов привело к созданию обширных иерархий именованных сущностей. В работе [6] рассматривается набор из 29 классов. Секин и др. [35]

[1] Под правилами мы будем подразумевать обычно те из них, которые создаются вручную.

[2] http://www.lrec-conf.org/

разработали иерархию, включающую в себя порядка 200 категорий сущностей[1], таких как продукты, книги, события, животные, растения, аэропорты и т.д.

1.1.2 Тренировочные данные

Методы машинного обучения с учителем наиболее популярны при выделении сущностей [29]. Системы такого типа обучаются на некотором размеченном сущностями тексте; обученная система может принимать на вход уже произвольный текст. Наличие тренировочного набора - довольно узкое место у такого типа методов. Как правило, тексты размечаются вручную специалистами в конкретной предметной области, что дорого и отнимает много времени. Полезность таких текстов с годами уменьшается; например, в новостном тренировочном наборе 1996 года вряд ли можно встретить упоминание компании, которая была образована, скажем, году в 2009, а следовательно возникает вопрос: сможет ли система корректно распознать сущность. В 2008 году были опубликованы несколько работ, направленных на автоматическое получение тренировочных данных [41, 33, 19]. Отметим работу [33], в которой было продемонстрировано получение тренировочного корпуса для *нескольких языков* при помощи Википедии.

1.1.3 Оценка систем выделения сущностей

Оценка систем выделения сущностей является индикатором прогресса данной области а также проверкой работоспособности новых методов. Как правило, оценка систем проводится на корпусах, размеченных вручную, техники измерения варьируются.

На серии конференций CoNLL был предложен простой способ оценки: именованная сущность выделена системой правильно, если ее класс и границы, обозначенные системой, совпадают с классом и границами, размеченными в корпусе; иначе сущность выделена неправильно. Назовем такой способ оценки оценка методом точного соответствия. Точность (P), полнота (R) и F-мера в данном случае определяются следующим образом[2]:

$$P = \frac{\text{кол-во верно выделенных сущностей}}{\text{кол-во всех выделенных сущностей}},$$

$$R = \frac{\text{кол-во верно выделенных сущностей}}{\text{кол-во сущностей в корпусе}},$$

$$F = \frac{2PR}{P+R}.$$

Данный метод оценки широко распространен, однако подвергается критике. Оценка точным соответствием не позволяет снисходительно относится к ошибкам в границе

[1] http://nlp.cs.nyu.edu/ene/

[2] Указанные формулы можно легко изменить, чтобы измерять качество выделения сущностей определенного класса.

сущности или в ее классе, которые вполне могут быть совершены и людьми при разметке текста. Кристофер Маннинг [23] предложил способ подсчета сегментов, который бы учитывал 3 дополнительных типа ошибки: сущность выделена, но есть неточность в границе, есть ошибка в классе сущности, но граница верна, ошибка есть как в классе, так и в границе сущности. Однако, предложенный способ не нашел широкого распространения.

Существуют и другие способы оценки, применявшиеся в разное время. На серии MUC [17] конференций система оценивалась с точки зрения двух способностей: правильно определять класс и правильно определять границы. Значимость одних типов ошибок над другими рассматривалась на серии конференций ACE[1].

1.1.4 Методы выделения именованных сущностей

В литературе упоминаются примеры использования большого числа обучающихся алгоритмов: метод опорных векторов [25], скрытые марковские цепи [15], метод максимальной энтропии [12, 9] и т.д. В серии работ в качестве базового метода выделяют систему, основанную на модели условных случайных полей (УСП) [22, 14, 21]. Модель УСП была специально разработана для разметки и сегментации последовательностей и впервые для выделения сущностей была продемонстрирована в работе [26].

Выбор набора признаков является более важным этапом при построении системы выделения сущностей, чем выбор модели аннотатора. Условно, используемые признаки можно разделить на несколько групп:

- **Признаки уровня слов.** Данная группа признаков наиболее общая для систем выделения сущностей. Она включает в себя слова, символьные n-граммы, префиксы, суффиксы, части речи, орфографические признаки, а также соответствующие биграммы и триграммы признаков.

- **Признаки уровня документа.** Данный набор признаков кодирует информацию о корпусе в целом: можно ли найти акроним для данного термина, позиция слова в предложении, пытаемся ли мы выделять сущность в определенной зоне (заголовке, тексте статьи) и т.д.

- **Признаки дополнительных источников информации.** Часто используются списки именованных сущностей, слов-указателей (для организаций - это Inc., Corp.), стоп-слов, слов, начинающихся на заглавную букву, но не являющихся именами (President, January, ...) и т.д.

[1] http://www.nist.gov/speech/tests/ace/ace05/doc/ace05-evalplan.v3.pdf

Базовый набор признаков обычно составлен из признаков первой группы для слов, находящихся в скользящем по тексту окне размера до 5 токенов. Под токеном подразумеваются не только слова, но и символы пунктуации.

1.1.5 Тенденции последних лет

Некоторые тенденции последних лет не нашли отражения в обзоре [29]. Без претензии на полноту попробуем выделить некоторые из них.

Рассмотрим следующий пример:

> William Gates III is an American business magnate... During his career at Microsoft, Gates held the positions of CEO...

В пределах одного параграфа встречаются два вхождения слова Gates, и системе следует классифицировать эти вхождения как имена людей. Однако, используя в качестве признаков только контекст текущего слова, зависимости такого плана невозможно моделировать. Базовая идея, что в пределах одного документа одинаковые названия скорее всего будут иметь одинаковые метки. МакКаллум и Саттон [36] предложили использовать усложненную модель УСП, которая дополнительно моделирует нелокальные зависимости между одинаковыми словами, написанными с заглавной буквы. Финкель и др. [14] определили модель штрафов, которая уменьшает вероятность последовательностей меток, разрешающих несогласованности. Ратинов и др. [31] использовали совмещение локальных признаков для токенов одинакового типа.

Серия работ последних лет посвящена использованию неразмеченных данных с целью повышения качества выделения сущностей. В работе [31] в качестве признака для токена использовались номера кластеров, в которые он попадает. Кластеры слов были построены с использованием алгоритма, описанного в работе Брауна [5]. Авторы [22] использовали признаки, полученные с помощью покластеризованных фраз поисковых запросов. Им удалось достичь наилучшего результата на тестовом наборе CoNLL-2003 ($F = 0.909$).

1.2 Википедия

1.2.1 Обзор Википедии

В последнее время в исследованиях, связанных с обработкой естественных языков, все более популярным становиться использование свободного вэб-ресурса Википедия. По сравнению с произвольными текстами в энциклопедии присутствует вполне определенная богатая структура, упрощающая извлечение из нее полезной информации. Википедия используется в различных приложениях: разрешение лексической многозначности,

построение онтологии и тезауруса, классификация текстов, сегментация запросов и т.д. [28, 16]

Энциклопедические статьи описывают определенный объект и обычно начинаются с некоторой аннотации, содержащей краткое описание всей статьи. Выделив отдельно первое предложение первого абзаца, которое обычно пишется на манер определения, мы будем называть его определяющим предложением. Также статья может содержать некоторую полу-структурированную информацию: информационные таблицы (infoboxes), таблицы описания видов (taxoboxes) и т.д. Они содержат некоторый стандартно определенный набор фактов об объекте. Все статьи принадлежат хотя бы одной категории, которые перечислены на странице в нижем колонтитуле, например, “Bill Gates” принадлежит категориями: “American computer programmers”, “1955 births”. Через ссылки статьи могут быть связаны с соответствующими переводами на другие языки.

В Википедии можно выделить еще несколько типов статей:

- **Страницы многозначных терминов.** Это специальный тип статей, содержащий несколько возможных трактовок некоего термина. Отражают явление омонимии терминов, например, “Apple (disambiguation)” содержит следующие возможные трактовки слова “Apple”: “Apple Inc.”, “The Apple (1980 film)”, “Apple Bank”, ...

- **Страницы-списки.** Этот тип страниц выполнят функцию, схожую с категориями. Страница-список содержит ссылки на страницы определенного класса (например, “List of monarchs of Korea” содержит список монархов Кореи). Однако в отличие от категорий статья не обязана содержаться в каком-либо списке.

- **Страницы-перенаправления.** Данный тип страниц не несет информации сам по себе, а предназначен для автоматического перенаправления пользователей на другие страницы. Страницы-перенаправления часто отражают пример синонимии терминов (‘IBM PC Company’, ‘I.B.M.’, ‘International Business Machines’, ... $\rightarrow$ ‘IBM’, слева от стрелки содержатся заголовки страниц-перенаправлений, справа - статья, на которую они ссылаются.)

1.2.2 Википедия и выделение сущностей

Использование Википедии может быть полезным и при выделении именованных сущностей. Большинство энциклопедических статей описывают объекты, которые попадают под стандартные категории имен[1].

[1]Около 72%.

Если сущность A упоминается на странице B, то, как правило, со страницы B есть ссылка на A. Данное наблюдение было использовано в работах [33, 19] для автоматического создания аннотированного корпуса. Идея - разметить заголовки внутренних ссылок, присутствующих в тексте, классом соответствующей статьи. В данном случае отдельно решалась задача о том, какой класс необходимо приписать статье. В работе [33] был сделан акцент на использование энциклопедии как тренировочного корпуса для различных языков. Классификация страниц производилась с использованием простых правил, основанных на категориях Википедии. Правила создавались вручную. Работа [19], проведенная в австралийском университете, описывает процесс преобразования английской Википедии в тренировочный корпус. Авторами был предложен итеративный процесс классификации энциклопедических статей по категориям сущностей, с $F = 0.92$ (только по сущностям). Для увеличения размера корпуса были предложены различные эвристики. Авторы работы также исследовали, как влияет добавление полученного корпуса к существующим тренировочным множествам (MUC-7, CoNLL-2003, BNN). Полученный таким образом тренировочный набор, конечно, полезен сам по себе, но его использование с целью существенно повысить качество систем выделения сущностей может быть сомнительным из-за неполного соответствия классов или границ имен, из-за специфики лексикона, используемого в энциклопедии. Тесты [19] показали ухудшение качества на двух из трех тестовых коллекциях при использовании добавочного корпуса.

Информация, предоставляемая Википедией, может быть напрямую добавлена при обучении системы. Рассмотрим определяющее предложение:

William Henry “Bill” Gates III is an American business magnate...

Как было отмечено в работах [1, 20], главное существительное фразы, стоящей после глагола *to be*, является хорошим индикатором класса сущности, (мы будем называть данное существительное определяющим; в данном случае “magnate” явно указывает на то, что в статье описывается некоторая персона). Авторы [20] используют определяющее существительное как признак при обучении классификатора, основанного на модели условных случайных полей: каждому немногозначному вхождению википедийного заголовка в тексте в качестве признака приписывалось соответствующее существительное. Так как определяющие существительные в рамках всей энциклопедии составляют большое множество, то способность системы обучиться на данных признаках сильно зависит от тренировочной коллекции, что в конечном счете может сказаться на полноте. Предлагаемый метод позволил выиграть около 1.6% F-меры на тестовой коллекции CoNLL-2003 ($85.0\% \rightarrow 86.6\%$). Следует заметить, что данный подход является достаточно гибким, так как не требует предварительной классификации Википедии и может

быть использован при выделении сущностей на произвольное число классов при наличии подходящих тренировочных коллекций.

Ратинов [31] предложил метод генерации списков сущностей из Википедии, основываясь на ее категориях, и использовал их для получения признаков. К сожалению, авторы не приводят оценку качества полученным спискам и не указывают, как добавление соответствующих признаков влияет на систему.

В отдельную группу можно выделить работы, направленные на выделение и классификацию сущностей в самой Википедии. Чтобы определить, идет ли в статье речь об именованной сущности или нет, Бунеску и др. [7] руководствовались соглашениями о написании статей в Википедии: все слова, входящие в имена собственные, должны быть написаны с большой буквы. В случае однословного заголовка неоднозначность разрешалась подглядыванием написания в тексте статьи. Данная идея с некоторыми ее дополнениями использовалась также в работе [4]. Зирн и др. [42] предложили более изощренный набор эвристик для классификации Википедии на концепции и сущности. Например, статья с заголовком, написанным во множественном числе, описывает концепцию; википедийные категории, определенные как именованные сущности, указывают на то, что и статья описывает именованную сущность и т.д. Оценка предложенных эвристик была проведена с использованием ResearchCyc, точность (accuracy) метода составила $A = 0.845$. Торал [1] для определения, описывает ли страница именованную сущность или нет, использовал статистику написания заголовка на различных языках, опираясь на идею Бунеску.

Для решения задачи классификации Википедии по классам именованных сущностей в литературе обычно используются подходы машинного обучения. Бол [3] предложил использовать правила, использующие внутреннюю структуру Википедии (информационные таблицы, координаты), и метод опорных векторов. В качестве классов были взяты: люди, организации и места. Как и ожидалось, правила показали лучшую точность, но низкую полноту в сравнении с методом машинного обучения. В качестве признаков для машинного обучения использовался текст статьи.

Салех и др. [34] использовали метод опорных векторов в сочетании с $\beta - \gamma$ пороговой настройкой. В качестве признаков для классификации были использованы слова, составляющие аннотации статей, категории, имена атрибутов информационных таблиц и шаблона “persondata”. В работе было показано, что аналогичные признаки, собранные с соответствующих страниц, написанных на других языках, могут быть полезны при классификации, но в основном не для английской Википедии.

Дакка и Кукерзан [10] в своей работе показали, что классификация с использованием наивного байесовского классификатора хуже, чем классификация с использованием метода опорных векторов, натренированного на тех же признаках. Также они исследовали различные комбинации признаков, такие как текст статьи, заголовки таблиц,

заголовки ссылок и т.д.

Большое число признаков, специфических для Википедии (шаблоны, таблицы описания видов, ...), было рассмотрено в работе [37]. Построенный классификатор показал хорошие результаты на наборе самых популярных страниц Википедии. Однако из-за того, что тестовый набор по сути не отражает реальное распределение признаков в энциклопедии, не совсем корректно обобщать результаты классификации на все множество статей.

Ватанаб и др. [40] использовали модель условных случайных полей, построенную на структуре Википедии, для классификации внутренних ссылок. Классификация проводилась по 12-ти классам сущностей. Результаты получились хуже в сравнении с методами, предлагающими классификацию статей по отдельности.

Бон и Норваг [4] использовали категории для генерации списков именованных сущностей. Статьи с категориями, удовлетворяющими определенным регулярным выражениям (например, "* companies"), классифицировались по трем классам: компании, организации и люди. Эффективность предложенного решения была протестирована на 585 страницах с результатами точности (accuracy) 98%, 97%, 99.8%.

Торал [39] анализировал определяющее предложение и иерархию существительных в WordNet[1], чтобы предсказать класс соответствующей страницы.

Таблица 1 обобщает результаты, описанные выше. Заметим, однако, что указанные значения не сравнимы строго, так как проводились при разных условиях и на разных тестовых коллекциях. Но они дают общее представление о сложности задачи и о возможных результатах. Следует отметить, что метод опорных векторов в качестве классификатора показывает лучшие результаты в сравнении с другими методами. Классификация на основе правил показывает высокую точность, но низкую полноту и требует ручного анализа текста.

Статья	PER	ORG	LOC	MISC	COMM	DAB
Бол [3]	72.7	41.6	70.5	-	-	-
Дакка [10]	95.1	93.4	95.4	92.4	87.8	-
Тардиф [37] (I)	96	95	99	94	-	-
Тардиф [37] (II)	95	93	99	89	93	98
Салех [34]	95.7	82.2	92.4	-	-	-
Нотман [19]	97	85	95	80	79	96
Торал [39]	78.3	22.2	68.1	-	-	-

Таблица 1: Классификация страниц английской Википедии (F-мера). Условные обозначения: PER - человек, ORG - организация, LOC - место, MISC - разное, COMM - не именованная сущность, DAB - страница многозначных терминов

[1]http://wordnet.princeton.edu/

2 Идея метода

Ключевая идея нашего метода заключается в том, что из Википедии можно извлечь списки именованных сущностей, причем с обширной дополнительной информацией о синонимии или полисемии соответствующих терминов. Полученные списки можно использовать для извлечения дополнительных признаков при выделении сущностей, но уже из текста. Википедия содержит порядка $3,5$ млн статей, большинство из которых попадают под категории именованных сущностей, и порядка $4,5$ млн. страниц-перенаправлений. Предлагается создать словарь, содержащий пары “заголовок страницы” и “класс соответствующей статьи” (классы страниц-перенаправлений отождествляются с классами статей, на которые они ссылаются). Схема потока данных указана на рис. 1.

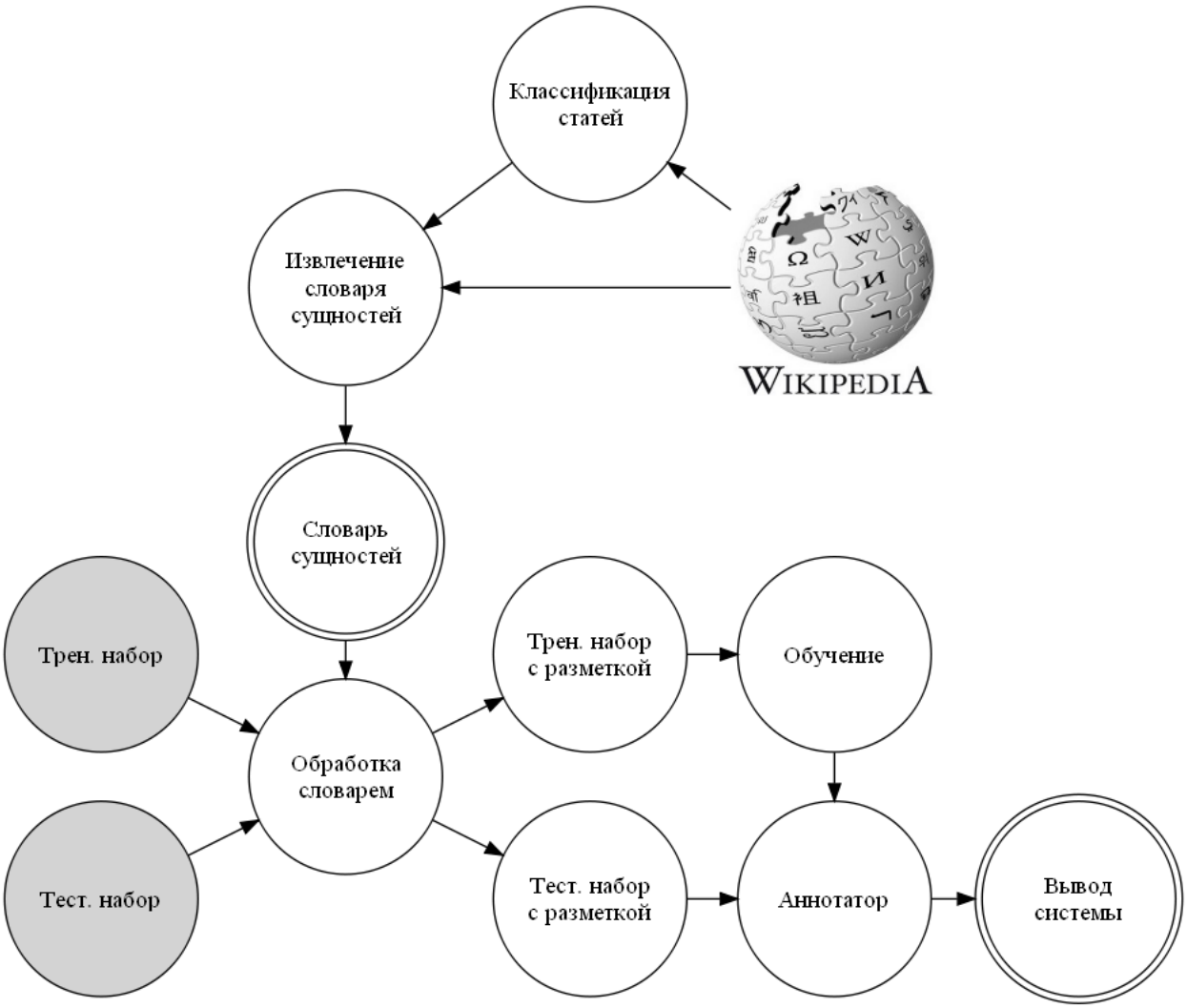

Рис. 1: Общая идея метода. Схема потока данных.

Рассмотрим далее подробно следующие этапы работы:

- Классификация статей Википедии по классам именованных сущностей и извлечение списков именованных сущностей из энициклопедии;
- Добыча дополнительных признаков из текста с использованием Википедии и выделение сущностей.

3 Извлечение сущностей из Википедии

Прежде чем приступить к рассмотрению задачи извлечения сущностей из Википедии, дадим некоторую справку об используемых методах.

3.1 Наивный байесовский классификатор

Наивный байесовский классификатор (Naive Bayes classifier) основан на применении теоремы Байеса с предположением, что признаки документа независимы между собой. Алгоритм рассматривает вероятность документа D с данным набором признаков $\{x_1, x_2, ..., x_n\}$ принадлежать классу C:

$$p(C|D) = p(C|x_1, x_2, ..., x_n).$$

Используя теорему Байеса, перепишем:

$$p(C|x_1, x_2, ..., x_n) = \frac{p(C)p(x_1, x_2, ..., x_n|C)}{p(x_1, x_2, ..., x_n)}.$$

Знаменатель $p(x_1, x_2, ..., x_n) = Z$ - не зависит от класса и поэтому не представляет особой ценности. Числитель, используя предположение о независимости признаков, можно переписать следующим образом:

$$p(C)p(x_1, x_2, ..., x_n|C) = p(C)\prod_{i=1}^{n} p(x_i|C)$$

Подведем итог: условная вероятность принадлежать классу C при данном наборе признаков может быть выражено следующим образом:

$$p(C|D) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(x_i|C)$$

Оценка параметров данной модели может быть выполнена подсчетом соответствующих относительных частот в тренировочной коллекции. Для принятия решения о классе выбирается наиболее вероятная гипотеза:

$$\underset{C}{\operatorname{argmax}}\, p(C|D)$$

3.2 Метод опорных векторов

Представим наши текстовые документы как точки в некотором пространстве. Основная идея метода опорных векторов (support vector machines) - поиск гиперплоскости с мак-

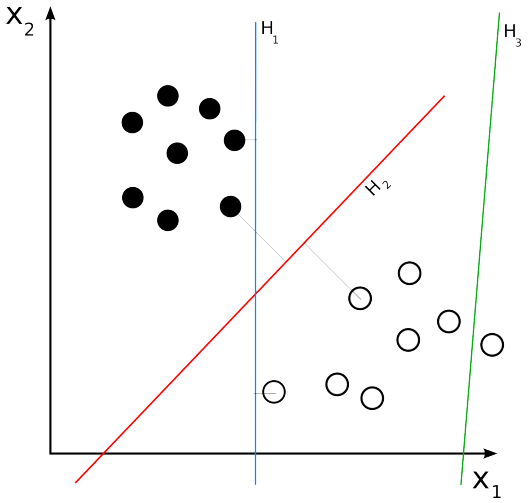

Рис. 2: Метод опорных векторов. H_2 - гиперплоскость с максимальным зазором.

симальным зазором, разделяющей между собой вектора разных классов (рис. 2). Задача поиска параметров для такой гиперплоскости $w \cdot x = b$ сводится к задаче квадратичного программирования. Рассматриваются случаи отделимости и неотделимости множества точек (во втором случае появляется параметр C, описывающий степень отделимости множеств). Примечательно, что вектора документов можно отобразить в пространство более высокой размерности, в котором они станут более отделимыми, но, как правило, для классификации текстов хватает линейного случая.

Предположим, что мы нашли подходящие параметры w и b на тренировочном наборе; для классификации нового примера нам всего лишь требуется определить, с какой стороны от гиперплоскости находится соответствующий вектор: $\text{sign}(w \cdot x - b)$.

Для классификации более чем на два класса мы использовали технику один-против-одного (one-against-one). Для каждой пары классов строится разделяющая гиперплоскость, в качестве итогового класса для документа выбирается тот, за который проголосовало большее число классификаторов.

3.3 Классификация Википедии

Разберемся, какие типы именованных сущностей будет полезно извлекать. Построенная в итоге система выделения сущностей будет тестироваться на тестовом множестве CoNLL-2003. Это значит, что по крайней мере необходимы представители статей для четырех классов: люди (PER), места (LOC), организации (ORG), разное (MISC). Чтобы сохранить естественное распределение страниц Википедии, необходимо рассматривать дополнительный класс OTHER, в который попадали бы статьи, не являющиеся сущностями. Для увеличения полезности нашей разметки мы добавили дополнительные

классы, заимствованные из таксономии [6]. Для этого была вручную проанализирована и размечена случайная выборка из 1500 статей Википедии, использовались классы, которые получили в данной выборке хотя бы 7 представителей. Вместе с примерами классы перечислены в таблице 2. Статьи, описывающие сущности, но не получившие достаточное число представителей, были определены как OTHER.

Класс	Пример
PERSON → [PER]	Barak Obama
GPE → [LOC]	Paris
GEO REGION → [LOC]	Nile (river)
ASTRAL BODY	12 Victoria
FACILITY → [LOC]	Lomonosov Bridge
ORGANIZATION → [ORG]	Apple Computers
VEHICLE → [MISC]	BMW 30
WORK OF ART → [MISC]	Eugene Onegin (novel)
GAME	Go
SUBSTANCE	Deiodinase
SOFTWARE	Visual Studio 2010
EVENT → [MISC]	Kerrville Folk Festival
PLANT	Matricaria chamomilla
INSECT	Scarabaeus
ANIMAL	Megalodon
OTHER	sleep, ball, ...

Таблица 2: Классы именованных сущностей, примеры. В квадратных скобках указан наиболее подходящий класс из набора CoNLL-2003.

Простыми эвристиками из полученной коллекции были отфильтрованы страницы-списки и страницы многозначных терминов. Как правило, страницу-список можно распознать по началу заголовка: “List of *”, “Table of *”. Для распознавания страниц многозначных терминов был предложен ряд правил, исходя из следующий наблюдений:

- Такие страницы созданы с использованием шаблонов *Template:disambig*, *Template:surname*, *Template:given name* или других шаблонов, раскрывающихся в данные;
- Они помечены хотя бы одной из категорий, входящих в иерархию *Category:Disambiguation pages*;
- Заголовок такой страницы содержит уточняющее слово *disambiguation* или *name*;

Данным набором эвристик удалось с высокой степенью точности отфильтровать страницы многозначных терминов (с точностью $A = 98\%$). Страницы, которые отфильтровать не удалось, были переразмечены классом OTHER. Ложно-позитивных примеров данная фильтрация не породила.

На этапе получения классов для статей Википедии мы использовали и сравнивали наивную байесовскую классификацию с методом опорных векторов, являющихся наиболее распространенными и показывающими лучшие результаты по сравнению с другими методами [24, 10, 37].

Мы провели анализ различных признаков, используемых для классификации статей, в том числе и указанных в работах [10, 37], и остановились на наборе признаков, связанных со структурой энциклопедии, которые позволили достичь наилучших результатов. Перечислим их.

Заголовок. Около 19% статей в Википедии содержат уточняющие фразы в заголовке: “Apple (band)”, “Mort (novel)”. В принципе они используются авторами статей, чтобы различать омонимичные термины. Фраза в скобках служит емким определением для термина, описываемого в статье. Для некоторых городов через запятую в заголовке указан регион, штат, к которому он относится. Сам по себе заголовок может быть говорящим: “Carnegie Mellon University.”

Категории. Созданные для упрощения навигации по страницам Википедии, они оказались самыми значимыми признаками при классификации, т.е. классификатор, натренированный только на данной группе признаков, показал лучший результат в сравнении с другими группами. Использовались слова, входящие в категории данной статьи, фильтрованные с использованием списка стоп-слов и приведенные к нижнему регистру. Мы провели дополнительную фильтрацию, чтобы исключить категории, имеющие служебное для Википедии назначение: “Category:Articles needing cleanup”, “Category:Articles that need to be wikified” и т.д.

Шаблоны. Шаблоны используются редакторами Википедии для упрощения процесса создания статей. Наиболее используемые элементы конструкции языка разметки выносятся в шаблон и могут быть использованы повторно. Так, например, существуют следующие шаблоны: “Template:Persondata”, разворачивающийся в таблицу, содержащую краткую биографическую справку о человеке; “Template:Coord”, хранящий информацию о географическом положении объекта. В качестве признаков использовались названия шаблонов, с помощью которых была создана данная страница.

Страницы-списки. Как уже упоминалось, с помощью простой эвристики возможно извлечь из Википедии набор страниц списков. Данный тип страниц имеет для энциклопедии значение, схожее со значением категорий. Но следует заметить, что если каждая статья должна быть помечена хотя бы одной категорией, то для страниц-списков это условие не выполняется. Тем не менее, если статья упомянута на странице “List of sovereign states”, то это будет хорошим показателем того, что речь в ней идет о государстве. В отличие от категорий обрабатывать страницы-списки сложнее, они имеют привязку к исследуемой статье только посредством внутренней ссылки, которая в принципе может указывать куда угодно. Обработка производилась следующим образом: со

страниц-списков (рассматривались страницы только с префиксом “List of”) извлекались все HTML списки, задаваемые тегами ul; для каждого li элемента извлеченного списка определялась первая внутренняя ссылка, соответствующая ссылке страница считалась принадлежащей обрабатываемой странице-списку, а заголовок списка (без префикса), обработанный аналогично категории, дополнил набор признаков.

Определяющее существительное. Как отмечалось ранее, первое предложение энциклопедической статьи часто носит определяющий характер, например, “London is the capital of England”. А как отмечалось в работе [20] определяющее существительное может служить хорошим индикатором класса сущности. Данный признак использовался для классификации. Извлечение определяющего существительного производилось аналогично алгоритму, описанному в [19]:

1. Из текста статьи извлекалось первое предложение;
2. Из него удалялись конструкции в скобках;
3. В предложении находилось первое вхождение глагола “to be”.
4. Предложение делилось на словосочетания с помощью OpenNLP[1];
5. Извлекалось первое словосочетание после “to be”, если его главное слово - существительное; иначе обрываем обработку;
6. Если главное существительное данного словосочетания одно из следующих: “from”, “kind”, “one”, “sort”, “type”, “variety” - извлекаем следующие словосочетание;
7. Если главное существительное имеет притяжательный падеж - извлекаем следующее словосочетание.

Данный набор шагов позволяет достаточно точно извлекать определяющие существительное. Отметим тем не менее, что первое предложение первого абзаца не всегда является определяющими, и по подсчетам авторов [19] для приблизительно 19% статей алгоритм ничего не извлекает.

Таблица описания видов (Template:Taxobox). Чтобы повысить отделимость классов PLANT, ANIMAL, INSECT, в качестве признака дополнительно использовались параметры шаблона Taxobox. Шаблон описывает биологическую классификацию видов живых существ.

Отметим, что в набор признаков итоговой системы не был включен текст статьи. Базовый набор признаков был составлен из текста и заголовка статьи.

[1] http://incubator.apache.org/opennlp/

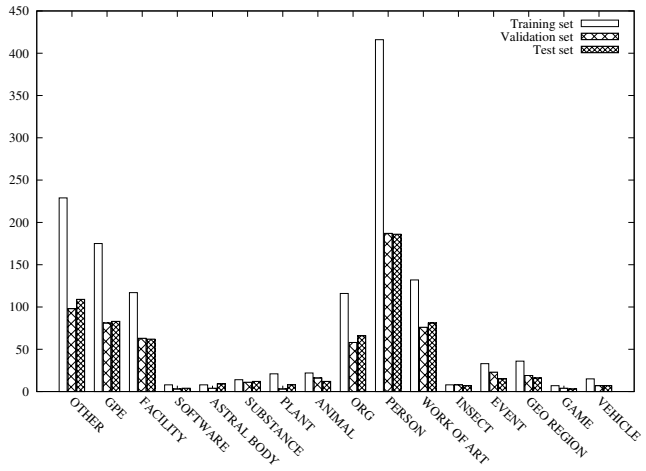

Рис. 3: Распределение по классам.

На вход классификаторам поступал вектор документа из пространства признаков (в компоненте - 1 если соответствующий признак представлен в статье, 0 - иначе). Для метода опорных векторов проводилось перевзвешивание вектора с учетом TF-IDF.

3.4 Эксперименты

В качестве тренировочного набора была использована случайная выборка из 1500 статей Википедии, отфильтрованная от страниц-списков и страниц многозначных терминов, т.е. итоговое множество составило 1357 страниц (train). Для настройки параметров и проверки качества классификаторов были дополнительно размечены два набора по 750 статей. После автоматической фильтрации эвристиками наборы составили 661 (testa) и 680 (testb) соответственно. Распределение страниц по классам для всех трех наборов указано на рис. 3.

Как видно, распределение страниц по классам не сбалансировано: некоторые классы содержат порядка 10 примеров. Тем не менее набор, необходимый для задачи выделения сущностей в тексте (PER, GPE, ORG), представлен достаточно хорошо.

Для оценки качества классификаторов мы использовали стандартные меры: точность (P), полнота (R), микро и макро F-меры:

$$P_c = \frac{\text{кол-во верно классифицированных страниц данного класса}}{\text{кол-во всех страниц попавших в данный класс}},$$

$$R_c = \frac{\text{кол-во верно классифицированных страниц данного класса}}{\text{кол-во страниц данного класса в коллекции}},$$

$$P_{macro} = \frac{1}{|C|}\sum_{c \in C} P_c, \quad R_{macro} = \frac{1}{|C|}\sum_{c \in C} R_c, \quad F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}},$$

	Базовые признаки		Итоговые признаки	
	NB	SVM	NB	SVM
F_{micro}	67.2	70.7	69.7	81.1
F_{macro}	55.9	52.9	68.7	70.7
P_{NE}	68.7	81.1	81.6	95.7

Таблица 3: Результаты тестирования на классификаторов на множестве testb. NB - наивный байесовский классификатр, SVM - метод опорных векторов.

$$F_{micro} = \frac{\text{кол-во верно классифицированных страниц}}{\text{кол-во всех страниц в коллекции}}.$$

Чтобы отдельно отслеживать насколько хорошо классификатор выделяет классы сущностей в Википедии, была введена дополнительная мера: точность по именованным сущностям (P_{NE}).

$$P_{NE} = \frac{T}{A - D - K},$$

где T - количество правильно распознанных классификатором именованных сущностей, A - число страниц в коллекции, D - правильно отклассифицированные страницы класса OTHER, и K - именованные сущности, попавшие в класс OTHER. Данная мера характеризует, насколько точны будут списки именованных сущностей, если использовать ответы классификатора как аннотации.

Для проведения тестов использовались наивный байесовский классификатор в реализации Weka [18] и метод опорных векторов в реализации LibSVM [8] с интеграцией в Weka [13]. Исходные коды тестов были реализованы на Java. Чтобы настроить C-параметр для метода опорный векторов мы предварительно запустили набор тестов на testa и оптимизировали ответы классификатора по P_{NE}. Итоговое значение C было выбрано равным 2, отметим однако, что даже большое варьирование данного параметра приводит лишь к незначительным изменениям в P_{NE} и в микро F-мере по всей коллекции.

Было проведено несколько экспериментов. Первый - сравнение базового и итогового набора признаков для двух классификаторов. В этом случае для тренировки использовался набор train, для тестирования - testb. Результаты представлены в таблице №3.

Метод опорных векторов показал наилучший результат в сравнении с байесовским классификатором, а набор структурных признаков зарекомендовал себя лучше, чем простой текст. Более подробная информация о точности и полноте для метода опорных векторов представлена на рис. 4.

Как видно точность для большинства классов составляет около 90%. Страдает полнота, но в большинстве случаев ошибки происходят из-за того, что классы сущностей путаются с классом OTHER. Высокий показатель P_{NE} позволяет надеяться, что вы-

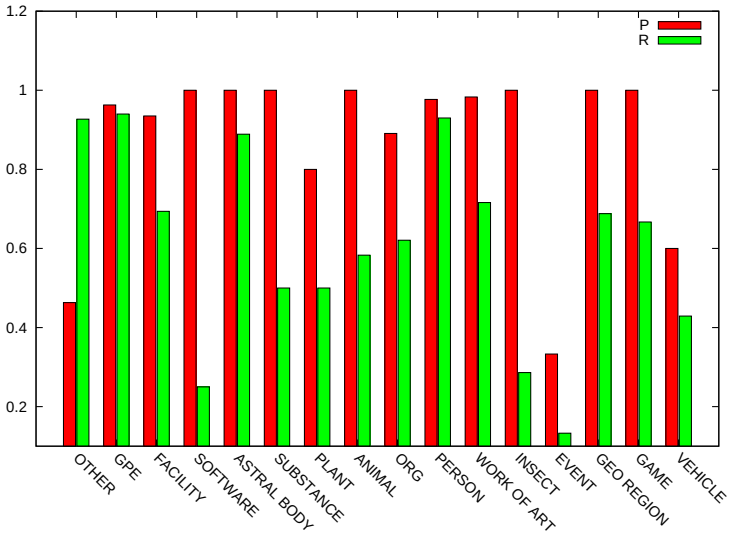

Рис. 4: Точность и полнота для метода опорных векторов на наборе testb.

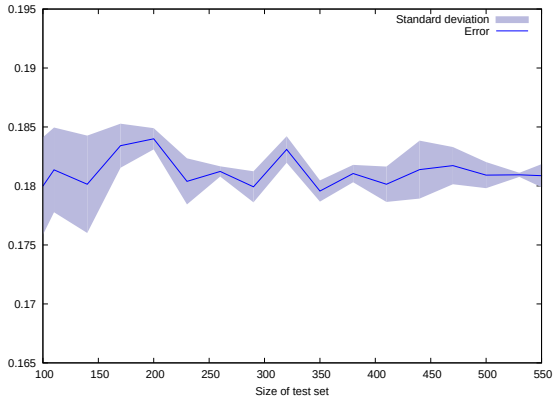

Рис. 5: График зависимости ошибки классификации от размера тестового множества.

деление именованных сущностей из всей Википедии пройдет с высоким показателем точности.

Следующий тест - проверка стабильности системы. Мы построили график зависимости ошибки классификатора от размера тестового множества (рис. 5). В каждой точке измерения проводилось усреднение полученных результатов по 100 случайным выборкам из testb. Отметим, что график ведет себя достаточно стабильно, колебания ошибки производятся относительно точки 0.18, максимум стандартного отклонения составил около 0.005.

Чтобы проверить степень натренированности классификатора, дополнительно были нарисованы два графика: график зависимости микро F-меры от количества признаков и график зависимости ошибки от размера тренировочного множества. Тестирование проводилось на множестве testb. В первом тесте признаки были упорядочены с помощью

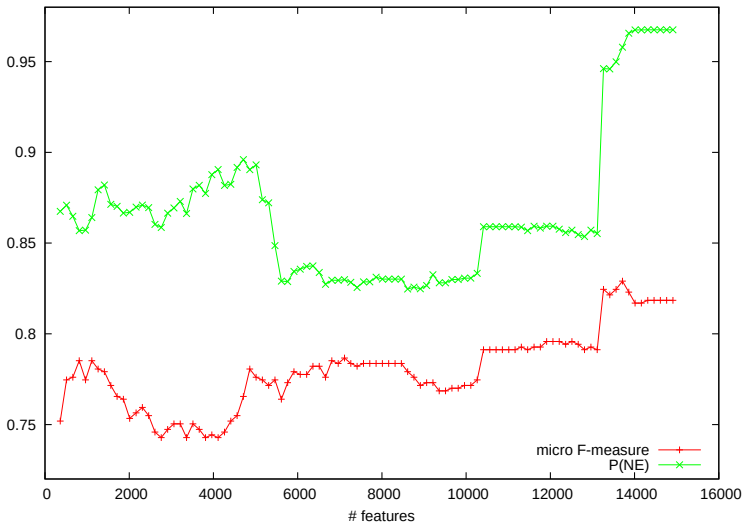

Рис. 6: График зависимости микро F-меры и P_{NE} от количества признаков.

χ^2 теста [24]. Проводилось постепенное отбрасывание признаков с худшими значениями данной статистики. График на рис. 6 имеет глобальный максимум для микро F-меры, однако для P_{NE} глобального максимума не достигается. Что можно воспринимать как недотренированность статистической системы.

Второй график подтвердил указанную тенденцию (рис. 7). Измерение ошибки проводилось с усреднением результатов по 100 случайным выборкам из тренировочного набора в каждой точке измерения. Поведение графика позволяет предположить, что при добавлении дополнительных примеров величина ошибки будет уменьшаться. Мы попробовали обучить классификатор на наборах train + testa - поведение графика не изменилось, количество неправильно классифицированных примеров чуть сократилось, но существенных изменений не произошло. Возможно, что случайная выборка оказалась не самым хорошим для классификатора тренировочным набором, и использование других подходов к выбору тренировочных примеров (например [37]) может привести к улучшению результатов.

Проведенные тесты показали превосходство метода опорных векторов над байесовским классификатором в данной задаче. Использование структурных признаков позволило получить высокую точность выделения страниц именованных сущностей из Википедии ($P_{NE} = 95.7\%$) на 15 классах. Эксперименты показали стабильность данного решения.

Классификатор, используемый для разметки всей Википедии, был натренирован на множествах train и testa. С помощью χ^2 теста был выбран оптимальный набор признаков.

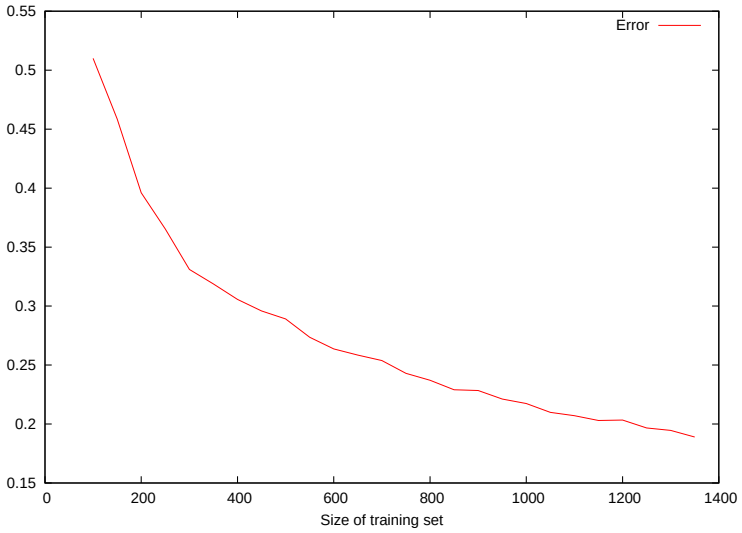

Рис. 7: График зависимости ошибки от размера тренировочного набора.

4 Выделение именованных сущностей в тексте

4.1 Модель условных случайных полей

В качестве модели для выделения сущностей мы использовали модель условных случайных полей (УСП). УСП - ненаправленная графическая модель, используемая для оценки условных вероятностей событий, соответствующих выходным вершинам некоторого графа, при наступлении некоторых событий, соответствующих входным вершинам. В линейной цепи УСП цепь образуют выходные вершины (см. рис. 8).

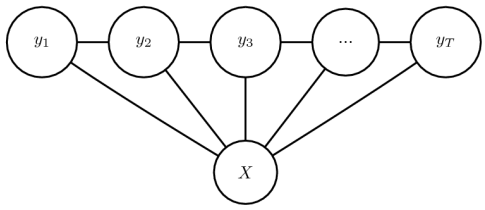

Рис. 8: Линейная цепь условных случайных полей.

Пусть $X = \langle x_1, x_2, x_3, ...x_T \rangle$ - последовательность наблюдаемых данных, например последовательность слов в тексте. $Y = \langle y_1, y_2, y_3, ...y_T \rangle$ - последовательность меток, взятых из заранее фиксированного множества (в данном случае имя человека, название организации, ...). Тогда можно доказать, что условная вероятность $P(Y|X)$ может быть

вычислена следующим образом:

$$P(Y|X) = \frac{1}{Z}\exp(\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(y_{t-1}, y_t, X, t)),$$

где Z - нормализующий множитель, f_k - произвольная признак-функция, λ_k - вес для соответствующей признак-функции, полученный в результате обучения системы.

Веса $\Lambda = \{\lambda_1, \lambda_2, ...\}$ выбираются таким образом, чтобы максимизировать логарифмическую функцию правдоподобия на наборе тренировочных примеров (Δ):

$$\Delta = \{\langle X, Y\rangle^1, \langle X, Y\rangle^2, ...\langle X, Y\rangle^N\}$$

$$L_\Lambda = \sum_{\langle X,Y\rangle\in\Delta}\log(P(Y|X)) - \sum_k \frac{\lambda_k^2}{2\sigma^2},$$

где вторая сумма - сглаживание по λ_k для предотвращения переобучения системы. Максимум функции правдоподобия может быть найден с использованием стандартных техник теории выпуклой оптимизации. Более подробно УСП описана в работе [36]. Используемая нами реализация УСП - MALLET [27].

4.2 Признаки

В принципе, выбор признаков при выделении сущностей имеет большее значение, чем выбор модели. В качестве базовых признаков мы использовали достаточно стандартный набор [22, 14]:

- Токены контекста и их биграммы: $w_{-2}, w_{-1}, w_0, w_1, w_2, w_{-1} : w_0, w_0 : w_1$;
- Части речи: $(t_{-1}, t_0, t_1, t_{-1} : t_0, t_0 : t_1)$;
- Написание слов: $(s_{-2}, s_{-1}, s_0, s_1, s_2, s_{-1} : s_0, s_0 : s_1)$;
- Префиксы и суффиксы текущего слова длиной от двух до пяти символов;
- Бинарный признак: является ли слово началом предложения.

Здесь индекс 0 - означает текущее слово, -1 - предыдущее и т.д., через двоеточие обозначаются биграммы соответствующих признаков.

Отдельного упоминания, пожалуй, требует признак написания слов. Это результат следующего отображения, действующего на слово: замена букв, чисел и всех остальных символов на “x” (“X” в случае, если буква заглавная), “0” и “-” соответственно и склеивание однотипных подпоследовательностей длиной больше двух. Например, iPhone $\rightarrow$ xXxx, 12-month $\rightarrow$ 00-xx.

Для кодирования разметки именованных сущностей применялась схема BILOU, показавшая свое преимущество перед BIO [31]. Поясним, так как классификатор просматривает одно слово за единицу времени, ему необходимо уметь моделировать разметку последовательностей большей длинны. В данном случае метка U - обозначает однословную сущность, B - начало сущности, L - последнее слово, I - слово внутри сущности и O - не сущность. Приведем пример текста с разметкой:

> William/B-PER Henry/I-PER "Bill"/I-PER Gates/I-PER III/I-PER is/O an/O American/U-MISC business/O magnate/O ./O

В качестве дополнительного признака, который вошел в состав итоговой системы, использовалась разметка текста, полученная при помощи Википедии. Заголовки страниц-перенаправлений и статей, которые были найдены в тексте, размечались с помощью BILOU схемы с значением соответствующего класса, полученного в результате классификации Википедии. Пример:

> Real/B-W-ORG Madrid/(L-W-ORG | U-W-GPE)'s Balkan strike force of Davor(B-W-PER | U-W-GPE) Suker/L-W-PER...

Мы использовали только подходящие для задачи классы: PERSON, GPE, GEO REGION, FACILITY, ORGANIZATION, EVENT. Набор признаков, полученных с помощью Википедии, был составлен следующим образом:

$$d_{-2}, d_{-1}, d_0, d_1, d_2, d_{-1} \ : \ d_0, d_0 \ : \ d_1.$$

4.3 Эксперименты

Тестирование производилось на эталонном тестовом множестве CoNLL-2003. Набор представляет из себя выборку из новостной ленты Reuters, размеченную четырьмя классами: имена людей (PER), названия организаций (ORG), местоположение (LOC) и разное (MISC). Коллекция разбита на тренировочный набор (train), проверочный набор (testa) и тестовый набор (testb). Набор train содержит 945 документов и приблизительно 203 тыс. токенов, testa - 216 документов и около 51 тыс. токенов, testb - 231 документов и 46 тыс. токенов. Корпус также размечен частями речи.

Чтобы наши результаты были сравнимы с предыдущими, оценка результатов проводилась методом полного соответствия. В таблице 4 представлено сравнение базовой системы и итоговой системы. Использование Википедии как словаря именованных сущностей позволяет поднять качество разметки на 2.9%.

	Базовые п.			Итоговые п.		
	P	R	F	P	R	F
PER	86.6	86.9	86.7	92.3	89.1	90.7
LOC	88.5	87.5	88.0	89.4	90.1	89.8
ORG	78.8	76.5	77.6	81.6	80.8	81.2
MISC	78.1	75.2	76.6	80.2	76.3	78.2
Общее	83.9	82.6	83.2	86.8	85.4	86.1

Таблица 4: Результаты тестирования на CoNLL-2003. P - точность, R - полнота, F - F-мера.

Используя testa, мы провели анализ разметки, полученной при помощи размеченной энциклопедии. Не все многозначные термины имеют в Википедии полное отражение, например, страницы с заголовками “Petofi” и “Arrive” содержат информацию о человеке и компании соответственно, хотя в тексте корпуса данные слова имеют разметку как организация и не именованная сущность. Отметим, также, некоторую неоднозначность страниц-перенаправлений: перенаправление с заголовком “Borodino” ведет на страницу “Battle of Borodino”, которая описывает событие - битву. Отсюда возникает ошибка при разметке текста, если термин “Borodino” в нем подразумевает населенный пункт.

Отметим также, что 4538 терминов в testa попали в категорию многозначных. Чтобы улучшить википедийную разметку текста, мы встроили в систему простой механизм снятия омонимии. Предположим, что мы нашли в тексте документа заголовок страницы с многозанчными терминами, например, “Washington”. Если в данном документе присутствует однозначный термин, например “George Washington”, соответствующий странице с омонимами, то для разметки многозначного термина используется класс найденного в тексте однозначного, т.е. PERSON в данном случае. Такой механизм позволил поднять общую F-меру на 0.2%. Также нами был опробован алгоритм Леска [30] для разметки неоднозначных терминов, но он не дал положительного результаты из-за слишком большой погрешности метода.

5 Технические решения

Предложенная система будет работать только в пределах множества CoNLL, если в нее не встроить дополнительные компоненты: компоненту определения границ токенов и компоненту разметки частей речи. Для этих целей использовалась открытая библиотека Apache OpenNLP[1]. Данный выбор был продиктован тем, что библиотека имеет хорошую производительность и реализована на языке Java. Общая архитектура системы изображена на рис. 9. Общий язык реализации - Java.

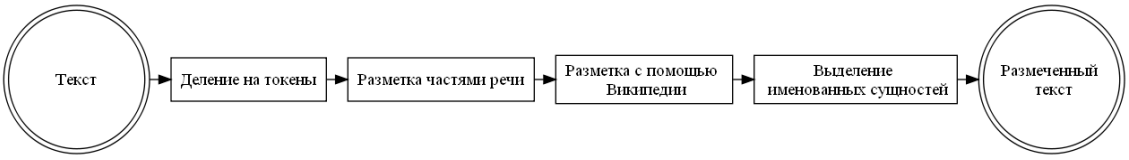

Рис. 9: Архитектура системы выделения сущностей.

[1]http://incubator.apache.org/opennlp/

Заключение

В качестве результатов данной работы можно отметить следующее:

- В рамках данной работы был предложен и проанализирован метод использования Википедии в целях повышения качества систем выделения сущностей. Мы продемонстрировали работоспособность метода для английского языка (полученный результат может быть обобщен для языков, которые в достаточной степени представлены в энциклопедии).
- Предложен и реализован метод для выделения сущностей из Википедии, который показал хорошее качество и стабильность. Получена разметка энциклопедии по 15 типам именованных сущностей.
- Была реализована система выделения именованных сущностей для четырех классов.

Мы надеемся, что результаты данной работы могут быть использованы для проектирования системы выделения сущностей, наилучшего качества.

Список литературы

[1] Rafael Muñoz Antonio Toral and Monica Monachini. Named entity wordnet. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.

[2] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. Named entity recognition in wikipedia. In *Proceedings of the Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, 2009.

[3] Abhijit Bhole, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. Extracting named entities and relating them over time based on wikipedia. *Informatica (Slovenia)*, 31(4):463–468, 2007.

[4] Christian Bohn and Kjetil Norvåg. Extracting named entities and synonyms from wikipedia. In *AINA*, pages 1300–1307. IEEE Computer Society, 2010.

[5] Peter F. Brown, Vincent J. Della Pietra, Peter V. Desouza, Jennifer C. Lai, and Robert L. Mercer. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, 1992.

[6] Ada Brunstein. Annotation guidelines for answer types, 2002.

[7] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics, 2006.

[8] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[9] Hai L. Chieu and Hwee T. Ng. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[10] Wisam Dakka and Silviu-Petru Cucerzan. Augmenting Wikipedia with Named Entity Tags, 2008.

[11] Hal Daume, III. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[12] Asif Ekbal, Eva Sourjikova, Anette Frank, and Simone P. Ponzetto. Assessing the challenge of fine-grained named entity recognition and classification. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 93–101, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[13] Yasser EL-Manzalawy and Vasant Honavar. *WLSVM: Integrating LibSVM into Weka Environment*, 2005. Software available at http://www.cs.iastate.edu/ yasser/wlsvm.

[14] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[15] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.

[16] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 1301–1306, Boston, MA, 2006.

[17] Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[18] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[19] James R. Joel Nothman. Transforming Wikipedia into Named Entity Training Data. pages 124–132, 2008.

[20] Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[21] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. 18th*

International Conf. on Machine Learning, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[22] Dekang Lin and Xiaoyun Wu. Phrase Clustering for Discriminative Learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[23] Christopher Manning. Doing Named Entity Recognition? Don't optimize for F1. August 2006.

[24] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[25] James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 184–187. Edmonton, Canada, 2003.

[26] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields. 2003.

[27] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[28] Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. Mining meaning from wikipedia. *Int. J. Hum.-Comput. Stud.*, 67:716–754, September 2009.

[29] David Nadeau and Satoshi Sekine. A Survey of Named Entity Recognition and Classification, 2007.

[30] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41:10:1–10:69, February 2009.

[31] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[32] L. F. Rau. Extracting company names from text. In *Proc. of the Seventh Conference on Artificial Intelligence Applications CAIA-92 (Volume I: Technical Papers)*, pages 29–32, Miami Beach, FL, 1991.

[33] Alexander E. Richman and Patrick Schone. Mining Wiki Resources for Multilingual Named Entity Recognition," ACL'08, 2008.

[34] Iman Saleh, Kareem Darwish, and Aly Fahmy. Classifying wikipedia articles into ne's using svm's with threshold adjustment. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 85–92, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[35] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[36] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields for Relational Learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[37] Sam Tardif, James R. Curran, and Tara Murphy. Improved Text Categorisation for Wikipedia Named Entities. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 104–108, Sydney, Australia, December 2009.

[38] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[39] A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. *EACL 2006*, 2006.

[40] Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 649–657.

[41] Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. Web-scale named entity recognition. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 123–132, New York, NY, USA, 2008. ACM.

[42] Cäcilia Zirn, Vivi Nastase, and Michael Strube. Distinguishing between Instances and Classes in the Wikipedia Taxonomy. In *ESWC*, pages 376–387, 2008.