

Михаил Хачай
Ольга Барина
Дмитрий Игнатов (редакторы)

Анализ изображений, сетей и текстов

Всероссийская научная конференция АИСТ'13
Екатеринбург, Россия, апрель 2013
Доклады

ДКНИТ 02



Доклады по компьютерным наукам и информационным технологиям

02

Издается с 2012 года

Основатели и первые редакторы серии:

Д. И. Игнатов, Р. Э. Яворский

Редакционный совет

Александр Авдеев,

Intel, Россия, Москва

Сергей Белов,

IBM, Россия, Москва

Александр Гаврилов,

Microsoft, Россия, Москва

Виктор Гергель

*НИУ Нижегородский Государственный Университет
им. Н.И. Лобачевского, Россия Нижний Новгород*

Александр Гиглавый

Лицей информационных технологий, Россия, Москва

Дмитрий Игнатов

НИУ Высшая Школа Экономики, Россия, Москва

Михаил Лаврентьев

*Новосибирский Государственный Университет, Россия,
Новосибирск*

Виктор Иванников

Институт системного программирования РАН, Россия, Москва

Александр Олейник

*Высшая школа бизнес-информатики, НИУ Высшая Школа
Экономики, Россия, Москва*

Александр Петренко

Институт системного программирования РАН, Россия, Москва

Андрей Терехов

*Санкт-Петербургский государственный университет, Россия,
Санкт-Петербург*

Олег Спиридонов

*Московский государственный технический университет им. Н. Э.
Баумана, Россия, Москва*

Павел Христов

Издательство «Открытые системы», Россия, Москва

Анатолий Шкред

Национальный Открытый Университет, Россия, Москва

Ростислав Яворский

Фонд «Сколково», Россия, Москва

Михаил Хачай
Ольга Барина
Дмитрий Игнатов (Редакторы)

Доклады всероссийской научной конференции АИСТ'2013

Модели, алгоритмы и инструменты анализа
данных; результаты и возможности для ана-
лиза изображений, сетей и текстов



Екатеринбург, 4 – 6 апреля 2013 года



Учредитель: Национальный Открытый Университет
«ИНТУИТ»

Редакторы тома

Михаил Хачай
Ольга Барина
Дмитрий Игнатов

Ответственный редактор

Екатерина Черняк

УДК [004.738.5+004.9](063)

ББК 32.973.202я431(2Рос)+32.973.26-018я431(2Рос)

Д63

ISBN 978-5-9556-0148-9

Доклады Всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ, Екатеринбург, 2013). Рассматриваются проблемы в области компьютерного зрения, анализа изображений и видео, анализа форумов, блогов и социальных сетей, анализ сетевых (графовых) и потоковых данных, компьютерной обработки текстов, гео-информационных систем, математических моделей и методов анализа данных, машинного обучения и разработки данных (Data Mining), рекомендательных систем и алгоритмов, Semantic Web, онтологии и их приложений.

Для студентов, аспирантов и специалистов в области машинного зрения, анализа изображений, текстов, социальных сетей и других неструктурированных данных.

© ИНТУИТ

Предисловие

В сборнике представлены работы участников Всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ 2012). Конференция АИСТ предоставляет площадку для обсуждения всего многообразия как теоретических задач, так и практического использования методов анализа данных, в том числе, и больших данных (Big Data). Здесь исследователи получают возможность рассказать о своих работах, поделиться опытом и обсудить свои задачи с коллегами и потребности отрасли с представителями коммерческих фирм.

Конференция проводилась 4 по 6 апреля 2013 года в столице Урала – Екатеринбурге. Все статьи можно условно разбить на несколько групп по темам:

- Математические модели и методы анализа данных,
- Машинное обучение и разработка данных (data mining),
- Анализ форумов, блогов и социальных сетей,
- Рекомендательные системы и алгоритмы рейтингования,
- Semantic Web, онтологии и их приложения,
- Анализ изображений и видео,
- Компьютерная обработка текстов,
- Геоинформационные системы,
- Анализ социально-экономических данных.

Всего было получено 40 заявок, каждая из которых была оценена минимум двумя рецензентами. По итогам рецензирования 22 работы были отобраны для секционных докладов и 9 для постерных сессий. В программу конференции включены два миникурса с практикумами и 9 лекций, прочитанных приглашёнными

докладчиками, а также презентации компаний организаторов и спонсоров конференции. В рамках конференции прошел круглый стол «Наукоемкий бизнес».

Пользуясь этой возможностью, мы выражаем признательность всем организаторам, членам программного комитета, рецензентам, докладчикам, спонсорам и партнёрам конференции, благодаря которым эта конференция состоялась. Мы благодарны Национальному Открытому Университету «ИНТУИТ» за помощь в издании тома трудов конференции.

Апрель 2013

Ольга Барина
Дмитрий Игнатов
Михаил Хачай
Ростислав Яворский

Организаторы

Программный комитет конференции

Сопредседатели

Ольга Барина (МГУ)

Дмитрий Игнатов (НИУ ВШЭ)

Михаил Хачай (ИММ УрО РАН и УрФУ)

Координаторы

Ростислав Яворский (Фонд Сколково)

Члены

Наталия Байгарова (Яндекс)

Виктор Бочаров (OpenCorpora)

Павел Браславский (Kontur Labs и УрФУ)

Константин Воронцов (Форексис и ВЦ РАН)

Александр Гальперин (УрФУ)

Владимир Горшенин (ЧелГУ)

Леонид Дворянский (НИУ ВШЭ)

Алексей Друца (МГУ и Витология)

Максим Дубинин (NextGIS)

Виктор Ерухимов (Itseez)

Леонид Жуков (НИУ ВШЭ)

Кирилл Корняков (Itseez)

Александр Крайнов (Яндекс)

Сергей Кузнецов (НИУ ВШЭ)

Борис Миркин (НИУ ВШЭ)

Ксения Найдёнова (ВМедА)

Александр Панченко (Université Catholique de Louvain)

Евгений Переводчиков (ТУСУР)

Александра Савельева (НИУ ВШЭ)

Павел Сердюков (Яндекс)

Никита Спиринов (University of Illinois at Urbana-Champaign)

Rustam Tagiew (Qlaym GmbH)
Екатерина Черняк (НИУ ВШЭ)
Александр Чигорин (Яндекс)

Организационный комитет конференции

Ирина Войчитская (Яндекс)
Мария Степанова (СПбГУ)
Дмитрий Усталов (ИММ УрО РАН)
Екатерина Щербакова (УрФУ)
Ростислав Яворский (Фонд Сколково)

Секретарь организационного комитета

Екатерина Черняк (НИУ ВШЭ)

Спонсоры и партнеры конференции

Институт математики и механики УрО РАН
Уральский федеральный университет
Национальный исследовательский университет Высшая школа экономики (НИУ-ВШЭ)
Уральский ИТ-кластер
IT-People
СКБ Контур
Издательство Открытые Системы
Российская венчурная компания
Лаборатория Цифрового Общества

Приглашенные докладчики

Виктор Бочаров (OpenCorpora)
Сергей Горшков (Бизнес Семантика)
Владимир Горшенин (ЧелГУ)
Дмитрий Ильвовский (НИУ ВШЭ)
Дмитрий Калаев (RedButton Venture Capital)
Юрий Катков (WikiVote)
Кирилл Корняков (Itseez)
Андрей Купавский (Яндекс)
Дмитрий Людмирский (РВК)

Наталья Остапук (Яндекс)

Александр Панченко (Université Catholique de Louvain)

Михаил Хачай (ИММ УрО РАН и УрФУ)

Оглавление

Секционные доклады

Прогнозирование мощности ветряных электростанций на основе непараметрического алгоритма к ближайших соседей <i>Е. Мангалова, И. Петрунькина</i>	1
Интеграция информационных систем с применением семантических технологий <i>С. Горшков</i>	9
Генерация сниппетов в поисковых системах как задача автоматического квазиреферирования <i>Л. Ермакова</i>	17
Об одном подходе к локализации антропометрических точек <i>А. Шушарин, К. Черенков, А. Гаврилюк, А. Валик</i>	26
Использование ресурсов Интернета для построения таксономии <i>Е. Черняк, Б. Миркин</i>	36
Аннотированные суффиксные деревья: особенности реализации <i>М. Дубов, Е. Черняк</i>	49
Серелекс: поиск и визуализация семантически связанных слов <i>А. Панченко, П. Романов, А. Романов, А. Филиппович, Ю. Филиппович, О. Морозова</i>	58
Методология создания программного комплекса «Интерактивная информационная доска» <i>Ю. Дроздова</i>	69
Применение методов машинного перевода для анализа древнерусских музыкальных рукописей <i>М. Даньшина, А. Филиппович</i>	77
Автоматическое извлечение правил для снятия морфологической неоднозначности <i>Е. Протопопова, В. Бочаров</i>	85

Применение метода комитета большинства для принятия решения по выдаче кредита <i>Ф. Чернавин</i>	93
Оценка сниппетов в поиске Mail.ru: корреляция автоматических и ассессорских оценок <i>А. Кутузов</i>	99
Разработка системы видеодетектирования транспортных средств <i>В. Кустикова, Н. Золотых, И. Мееров, Е. Козинов, А. Половинкин</i>	115
Фильтрация ложных соответствий описателей особых точек изображений <i>С. Белоусов, А. Шишков</i>	123
Некоторые аспекты задачи исследования распространения информации в социальной сети ВКонтакте <i>Е. Рабчевский, А. Цукерман</i>	133
Методы распараллеливания алгоритма сравнения дактилоскопических изображений <i>Д. Лепихова, В. Гудков</i>	142
Анализ тональности текста на русском языке при помощи графовых моделей <i>И. Меньшиков</i>	151
Ошибки первого и второго рода для простановки частных признаков на изображении отпечатка пальца <i>К. Дорофеев</i>	156
Проблемы применения классических методов распознавания для фотографических изображений пыльцевых зерен <i>А. Черных, Е. Замятина</i>	160
Метод классификации объектов различных классов на изображениях <i>Р. Захаров</i>	169
«Бизнес Семантика»: практика интеграции информационных систем с использованием семантических технологий <i>С. Горшков</i>	179

Анализ статистических алгоритмов снятия морфологической омонимии в русском языке	184
<i>Е. Лакомкин, И. Пузыревский, Д. Рыжова</i>	
Синтаксический анализ музыкальных текстов	196
<i>И. Голубева, А. Филиппович</i>	
Постерные доклады	
Распознавание и классификация актантов в русском языке	205
<i>И. Кузнецов</i>	
Применение автоассоциаторов к распознаванию последовательностей аккордов в цифровых звукозаписях	211
<i>Н. Глазырин</i>	
Использование связанных пространственных данных в геоинформационных системах	216
<i>С. Кузьмин</i>	
Применение модели Бокса-Дженкинса для прогнозирования объемов инвестирования в факторы производства	221
<i>Д. Насридинова, Е. Касаткина</i>	
Совершенствование одноязычных, двуязычных и мультязычных словарей: автоматизация процесса сбора материала	225
<i>М. Кюсева, Т. Резникова, Д. Рыжова</i>	
Дедупликация почтовых адресов с помощью методов обработки естественного языка и машинного обучения	233
<i>А. Филипов, А. Семенов</i>	
Построение системы распознавания и определения типа галактик	239
<i>А. Михайлов, В. Волкова</i>	
Идентификация подписи с помощью радиальных функций	244
<i>Э. Анисимова</i>	
Сравнение онлайн-сообществ на основе лексического анализа ленты новостей	254
<i>Д. Усталов, Ф. Краснов, Р. Яворский</i>	

Прогнозирование мощности ветряных электростанций на основе непараметрического алгоритма k ближайших соседей

Екатерина Мангалова¹, Ирина Петрунькина²

¹СибГАУ имени ак. М.Ф. Решетнева, Красноярск, Россия. mangalova@sibsau.ru

²СФУ, Красноярск, Россия. Laki4-ever@yandex.ru

Аннотация. Статья посвящена вопросу предсказания относительной выходной мощности ветряных электростанций и содержит описание следующих этапов решения практической задачи анализа данных: выбор значимых факторов, предварительная обработка данных, построение модели, ее проверка и интерпретация результатов. В качестве модели предсказания мощности выбрана непараметрическая модель k ближайших соседей.

Ключевые слова: анализ данных, алгоритм k ближайших соседей, прогнозирование, моделирование.

Введение

Энергоэффективность и энергосбережение входят в пятерку приоритетных направлений технологического развития в России. С помощью развивающихся технологий использования альтернативных источников энергии возможно повысить энергоэффективность, способствовать рациональному использованию ресурсов и сокращению выбросов парниковых газов [1]. Одним из активно развивающихся направлений в энергетике в настоящее время являются ветряные электрические установки.

Эффективная эксплуатация ветряных электростанций требует решения проблем, связанных с необходимостью оптимизации режимов их работы в рамках единой энергетической системы. В частности возникает необходимость прогнозировать мощность, генерируемую ветряной электростанцией, с целью минимизации затрат электростанций системы, использующих не возобновляемые источники энергии.

Постановка задачи

Постановка задачи и исходные данные взяты из открытого конкурса Global Energy Forecasting Competition 2012 [2]. Для долгосрочных прогнозов мощности семи ветряных электростанций используется следующий набор факторов: метеорологический прогноз (меридиональная и зональная компоненты скорости ветра, направление ветра, скорость ветра) и соответствующие прогнозу дата и время. Дискретность измерений – 1 час. Составляющие метеорологического прогноза проиллюстрированы на рис. 1.

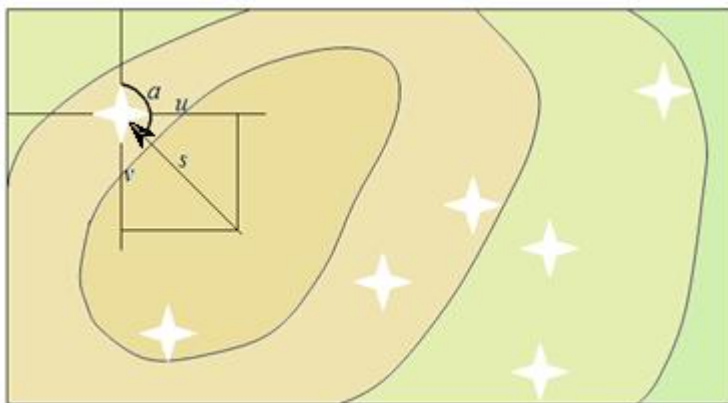


Рис. 1. Исходные данные: 7 ветряных электростанций и метеорологические прогнозы (u – зональная компонента скорости ветра, v – меридиональная компонента скорости ветра, s – скорость ветра, α – направление ветра)

Взаимное расположение ветряных электростанций и характеристики воздушного потока (температура воздуха, влажность и т.д.), которые согласно источнику [3] влияют на вырабатываемую мощность, неизвестны. Однако косвенно характеристики воздушного потока могут быть связаны с порядковым номером дня в году (временем года) и временем суток.

Выбор значимых факторов и преобработка данных

С целью выбора наиболее значимых факторов было использовано дерево регрессии [4]. Дерево регрессии позволяет последовательно разбивать имеющийся набор данных на подмножества с различными выборочными средними. Таким образом, разбиение по тому или иному фактору свидетельствует об изменении выборочной средней, а следовательно, и о наличии некоторой зависимости. Последовательность, в которой происходят разбиения, не учитывается, поэтому для предотвращения выбора факторов, значимых для небольших подмножеств данных, введем условие остановки роста дерева: минимальная мощность подмножеств (минимальный лист) – 500. Пример построенного дерева для ветряной электростанции №1 приведен на рис. 2.

В табл. 1 приведены факторы, значимые для тех или иных ветряных станций.

Табл. 1. Значимые факторы для различных ветряных электростанций

Фактор	Электростанция						
	1	2	3	4	5	6	7
Скорость ветра	+	+	+	+	+	+	+
Зональная компонента скорости		+	+	+		+	+
Меридиональная компонента скорости	+	+	+	+	+	+	+
Направление ветра		+			+		+
Год					+		
Месяц							
День месяца							
День года	+	+			+	+	+
Час	+	+	+	+	+	+	+

Факторы, значимые для пяти и более ветряных электростанций, были выбраны для включения в модель. Введем обозначения: x^1 – зональная компонента скорости ветра, x^2 – меридиональная компонента скорости ветра, x^3 – скорость ветра, x^4 – час, x^5 – порядковый номер дня в году. К этому набору факторов последовательно добавлялись скорости ветра в районах соседних ветряных электростанций: вначале фактор x^6 такой, что его включение в модель максимально улучшает точность прогноза, затем x^7 , выбранный с тем же условием. Прогнозируемую величину обозначим y , объем выборки – n .

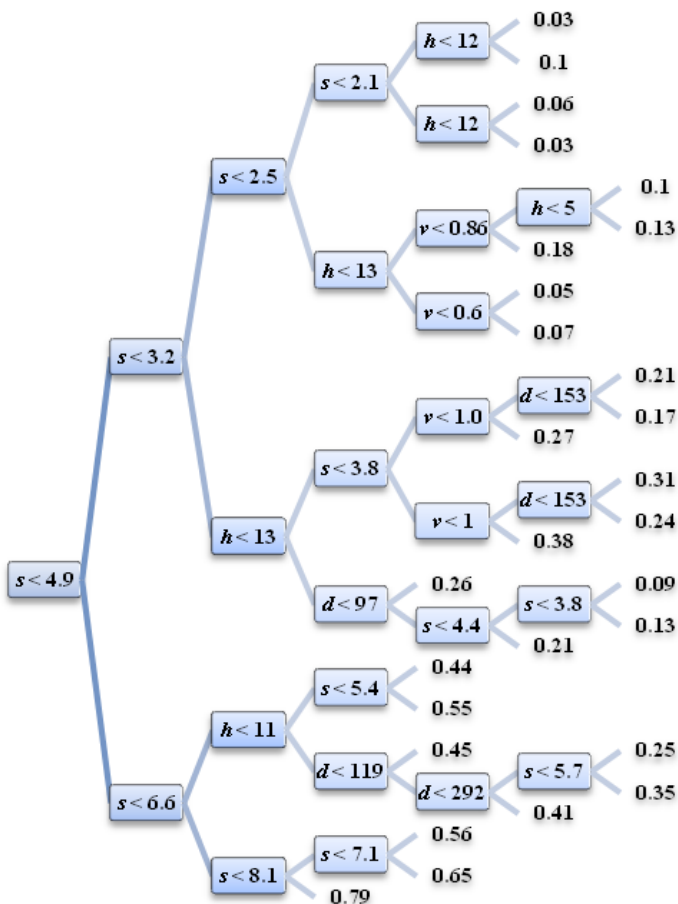


Рис. 2. Дерево регрессии для ветряной станции 1. Верхняя альтернатива – нарушение условия, нижняя – соблюдение; s – скорость ветра, h – час, d – день в году, v – зональная компонента скорости ветра

После выбора значимых факторов необходимо провести предварительную обработку данных.

В исходных данных были замечены два типичных случая аномальных измерений:

1. Высокая мощность при слабом ветре (может быть связано с ошибками в прогнозе погоды),

2. Низкая мощность при сильном ветре (может быть связано как с ошибками в прогнозе погоды, так и с аномальным функционированием ветряной станции).

Такие наблюдения были исключены из обучающей выборки.

Построение модели

Для предсказания мощности использован алгоритм k ближайших соседей. Выбор алгоритма был обусловлен следующими причинами:

а) Интерпретируемость модели. Алгоритм k ближайших соседей позволяет осуществлять прогноз, основываясь на наиболее похожих ситуациях (ближайших соседях) в прошлом в соответствии с выбранным расстоянием. Прогнозирование выполняется простым или взвешенным усреднением выходных значений k ближайших соседей.

б) Циклический характер факторов. Среди факторов, включенных в модель, есть циклические (час и порядковый номер дня в году). Алгоритм k ближайших соседей может быть настроен для работы с ними (в отличии, например, от деревьев решений).

в) Алгоритм не требует повторного обучения при поступлении новых данных. Добавление наблюдений в этом случае лишь расширяет область поиска ближайших соседей.

Для отыскания ближайших соседей были введены следующие меры расстояний между наблюдениями \bar{x}_p ($p = 1, 2, \dots, n$) и \bar{x}_q ($q = 1, 2, \dots, n$):

1. В пространстве одного фактора:

$$d^j(\bar{x}_p, \bar{x}_q) = w_j |x_p^j - x_q^j|, j = 1, 2, 3, 6, 7,$$

2. В пространстве одного циклического фактора:

$$d^4(\bar{x}_p, \bar{x}_q) = w_4 \min(|x_p^4 - x_q^4|, |x_p^4 - x_q^4 - 24|),$$

$$d^5(\bar{x}_p, \bar{x}_q) = w_5 \min(|x_p^5 - x_q^5|, |x_p^5 - x_q^5 - 365|),$$

3. В пространстве всех факторов:

$$D(\bar{x}_p, \bar{x}_q) = \sum_{j=1}^7 d^j(\bar{x}_p, \bar{x}_q).$$

Модель k ближайших соседей имеет вид:

$$\hat{y}(\bar{x}) = \frac{\sum_{q=1}^n \varphi(\bar{x}, \bar{x}_q) y_q}{\sum_{q=1}^n \varphi(\bar{x}, \bar{x}_q)}, \quad (1)$$

где

$$\varphi(\bar{x}, \bar{x}_q) = \begin{cases} D(\bar{x}, k) - d(\bar{x}, \bar{x}_q) + 0.1, & D(\bar{x}, k) \geq d(\bar{x}, \bar{x}_q), \\ 0, & D(\bar{x}, k) < d(\bar{x}, \bar{x}_q), \end{cases}$$

где $D(\bar{x}, k)$ – расстояние между \bar{x} и k -м ближайшим соседом.

Анализ исходных данных показал, что в большинстве случаев прогнозы ветра слабо отличаются друг от друга в течение некоторого промежутка времени. Ближайшие по времени наблюдения, таким образом, будут являться заведомо «хорошими» соседями. Данный факт приводит к занижению количества ближайших соседей и переобучению при использовании в качестве процедуры валидации скользящего экзамена или кратной кросс-проверки [5]. В этом случае модель будет демонстрировать высокое качество краткосрочного прогнозирования (1 – 2 часа), однако она будет неадекватна при долгосрочных прогнозах (до 48 часов). Таким образом, при настройке параметров исключаем ближайшие к проверочному множеству наблюдения из обучающей выборки.

Для настройки параметров модели (1) использовался следующий критерий:

$$W(\bar{w}, k) = \sum_{l=1}^{155} \sum_{i \in V_l} (y_i - \hat{y}(\bar{x}_i, T_l))^2,$$

где

$$V_l = ((\bar{x}_{\lambda(l)}, y_{\lambda(l)}), \dots, (\bar{x}_{\lambda(l)+35}, y_{\lambda(l)+35})), l = 1, 2, \dots, 155,$$

проверочные множества, $\lambda(l) = 13177 + 84(l - 1)$, k ближайших соседей отыскиваются из тестовых множеств:

$$T_l = \{(\bar{x}_f, y_f) : \forall (\bar{x}_z, y_z) \in V_l |z - f| > 48\}.$$

На рис. 3 проиллюстрирован процесс выбора обучающих и проверочных множеств.

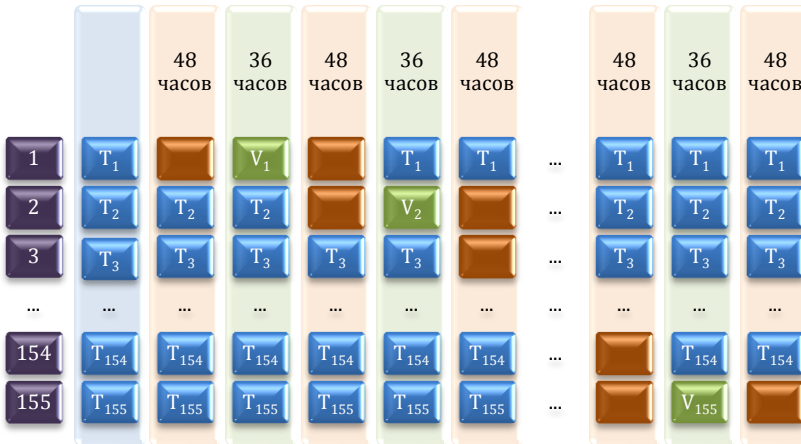


Рис. 3. Формирование проверочных и обучающих множеств. Зеленым выделены проверочные множества, синим – обучающие, красным - подмножества наблюдений, не включенные ни в проверочные, ни в обучающие множества

Для любого \bar{w} количество соседей k выбиралось методом полного перебора в диапазоне от 1 до 250. Оптимизация по параметрам \bar{w} выполнялась с помощью покоординатного спуска.

Результаты прогнозирования

Модель (1) была проверена на тестовой выборке. Тестовая выборка составляла примерно 30% от обучающей, значения мощности электростанций в моменты времени тестового периода были неизвестны участникам до окончания конкурса [2]. На рис. 4 приведено сравнение реальной мощности ветряной станции №1 и прогноза на фрагменте тестовой выборки.

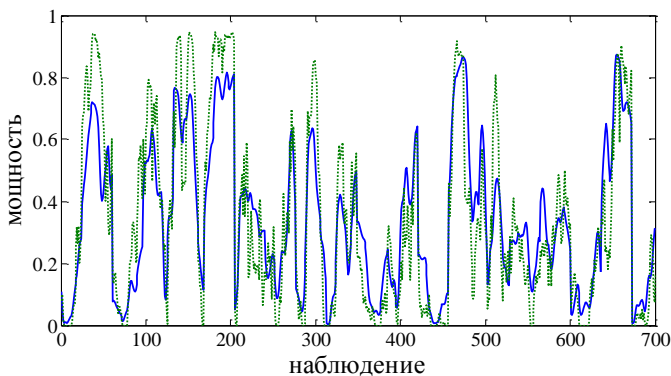


Рисунок 4. Сравнение реальной мощности на фрагменте тестовой выборки (пунктирная линия) и прогноза (сплошная линия)

Точность (среднеквадратическая ошибка) на тестовом множестве достигла уровня 0.1472, что позволило показать второй результат в конкурсе [6]. Близость среднеквадратической ошибки на тестовом множестве к ошибке на проверочном множестве (0.1389) свидетельствует об отсутствии переобучения.

Выводы

1. В работе предложен простой и эффективный алгоритм прогнозирования мощности ветряных электростанций в условиях неполной априорной информации.
2. Модификация кратной кросс-проверки позволяет избежать проблемы переобучения при выборе количества ближайших соседей,

Прогнозирование мощности ветряных электростанций на основе...

осуществлять краткосрочные и долгосрочные прогнозы с одинаково высокой точностью.

Список источников

1. Энергоэффективные технологии «Сименс» в России. URL: <http://w3.siemens.ru/energy-efficiency/energy-efficiency.html>. Дата обращения: 08.02.13.
2. Global Energy Forecasting Competition 2012, wind forecasting. URL: <http://www.kaggle.com/c/GEF2012-wind-forecasting>. Дата обращения: 08.02.13.
3. Crogg K.. Harvesting the Wind: The Physics of Wind Turbines. URL: <https://dspace.lasrworks.org/bitstream/handle/10349/145/fulltext.pdf>. Дата обращения: 08.02.13.
4. Breiman L., Friedman J. H., Olshen R. A., Stone C. J. Classification and Regression Trees. — Wadsworth Inc, 1984.
5. Воронцов К. В. Комбинаторный подход к оценке качества обучаемых алгоритмов. — Математические вопросы кибернетики / Под ред. О. Б. Лупанов. — М.: Физматлит, 2004. — Т. 13. — С. 5–36.
6. Leaderboard – Global Energy Forecasting Competition 2012. URL: <http://www.kaggle.com/c/GEF2012-wind-forecasting/leaderboard>. Дата обращения: 08.02.13.

Интеграция информационных систем с применением семантических технологий

Сергей Горшков

«Бизнес Семантика», Екатеринбург, Россия. serge@business-semantic.ru

Аннотация. Статья посвящена описанию способа построения архитектуры обмена данными между информационными системами, с применением технологий Semantic Web. Рассматриваются принципы преобразования данных при передаче между интегрируемыми системами, приводится пример такого преобразования. Оцениваются отличия предлагаемого способа от других применяемых в настоящее время практик интеграции (MDM, шины обмена сообщениями).

Ключевые слова: Semantic Web; семантические технологии; семантическая интеграция; обмен данными.

Введение

Для интеграции информационных систем применяется широкий набор инструментов, включающий различные варианты обмена через файловые выгрузки, создание веб-сервисов SOAP, использование шин обмена сообщениями, систем управления нормативно-справочной информацией, MDM-систем. Одним из наиболее трудоемких этапов реализации любого из перечисленных способов обмена данными является настройка семантического сопоставления структур данных, существующих в разных информационных системах. Часто для такого сопоставления создается программный код, выполняющий преобразование данных из одного структурного представления в другое. Настроенное

Интеграция информационных систем с применением семантических технологий

сопоставление требует поддержки, то есть нуждается в модификации при изменении структуры данных в одной из информационных систем.

Технологии Semantic Web могут быть применены для решения задач смыслового сопоставления и преобразования данных при их передаче между информационными системами. В этой статье мы рассмотрим один из способов построения архитектуры автоматического обмена информацией, основанный на технологиях «семантической паутины».

Преобразование информации при интеграции информационных систем

При использовании перечисленных выше традиционных способов интеграции информационных систем (далее ИС), чаще всего используется следующая схема: на выходе из ИС-источника данные преобразуются в некое промежуточное представление, а затем ИС-получатель размещает полученные из него данные в своих внутренних структурах. Промежуточным представлением в простейшем случае может быть CSV или XML-файл. Структура этого файла почти всегда повторяет структуру, в которой представлена информация в одной из интегрируемых систем. Подобная зависимость будет иметь место и при использовании других методов передачи: так, MDM-система сама является хранилищем информации, и использует представление, синтезированное из структур данных во всех интегрируемых системах. Сервисы SOAP реализуют методы, каждый из которых, как правило, сообщает информацию или выполняет операцию с информационными объектами одного типа, а набор параметров на входе и выходе каждого метода практически неизбежно зависит от свойств соответствующего объекта.

Суть предлагаемого нами способа состоит в том, что информация, передаваемая между системами, представляется в семантической форме, то есть в виде триплетов «подлежащее – сказуемое – определение». Подлежащим является уникальный идентификатор (URI) информационного объекта, сказуемым – свойство объекта, или вид его связи с другим объектом, а определением – URI другого объекта, или литерал, представляющий собой значение данного свойства.

Консорциумом W3C утвержден в качестве стандартов набор технологий, позволяющих выражать онтологии и данные, представленные в семантической форме, при помощи различных нотаций: OWL, RDFS, RDF. Задача преобразования информации на выходе из ИС-источника состоит в том, чтобы превратить данные, хранящиеся, скорее всего, в реляционной СУБД, в поток сообщений в одном из синтаксисов RDF (например, Turtle). Чтобы иметь возможность сделать это, необходимо определить онтологию, содержащую все требуемые для записи этой

информации понятия. В некоторых случаях есть возможность использовать стандартные онтологии или их расширения, в других – более целесообразным может оказаться создание собственной онтологии. Так или иначе, при выборе или составлении онтологии необходимо в большей степени ориентироваться на понятийный аппарат процесса, для автоматизации или поддержки которого предназначены интегрируемые системы, а не на то, в виде каких структур данные представлены в каждой из систем. Тогда при включении новых информационных систем в процесс обмена, или при изменении внутренней структуры данных в системах, требуемые изменения в интеграционных процедурах будут минимальными.

Пример преобразования данных в семантическую форму при интеграции информационных систем

Приведем пример преобразования данных из табличной формы в семантическую. Пусть в ИС-источнике хранится информация о сотрудниках организации: ФИО, паспорт, адрес и т.д. В другой таблице базы данных ИС-источника хранятся сведения о приказах, связанных с сотрудниками: прием на работу и увольнение, перевод на другую должность, и т.п. Структура таблиц БД ИС-источника будет такой (табл. 1):

Табл. 1. Структура базы данных ИС-источника

Таблица «Сотрудники»	Таблица «Приказы»
ФИО	Дата
Номер паспорта	Номер
Адрес	Сотрудник
...	Вид приказа

В случае реализации выгрузки в XML, скорее всего, файл с промежуточным представлением имел бы вид, показанный на рис. 1:

```
<Employee Name="Иванов И.И." Passport="65 03 111222" Address="ул. Мира, д.1">  
  <Order Date="2012-01-01" Number="1" Type="прием на работу"/>  
  <Order Date="2013-03-01" Number="2" Type="увольнение"/>  
</Employee>
```

Рис. 1. Представление БД источника в виде XML

Интеграция информационных систем с применением семантических технологий

Чтобы преобразовать данные в семантическую форму, необходимо составить онтологию. В данном примере, для простоты, сделаем ее полностью симметричной структуре таблиц базы данных. Онтология будет включать понятия (классы объектов) «Сотрудник» и «Приказ». Данные, выраженные при помощи какой-либо онтологии, удобно представить в виде информационного графа. Объекты, экземпляры классов – собственно сотрудники и приказы – станут вершинами этого графа. Каждый объект может обладать свойствами, типы которых также определены в онтологии. Для сотрудника это ФИО, номер паспорта и адрес, для приказа – сотрудник, дата, номер и вид приказа. Свойства конкретных информационных объектов будут ребрами графа, который мы строим. Значениями свойств могут быть ссылки на другие объекты (например, сотрудника), или литералы – текстовые и числовые величины. Кроме того, у каждого объекта есть идентификатор – URI, который состоит из типа объекта, символа # и уникального идентификатора объекта. Фрагмент графа может выглядеть так, как показано на рис. 2:



Рис. 2. Пример фрагмента информационного графа

После того, как ИС-источник преобразовала хранящуюся в ней информацию в такую форму, она может передать ее другим системам в виде потока триплетов, или фактов. Получится следующий «текст»:

Сотрудник #ivanov имеет имя Иванов И.И. Сотрудник #ivanov проживает по адресу ул. Мира, 1. Сотрудник #ivanov имеет паспорт с номером 65 03 111222. Приказ #0001 относится к сотруднику #ivanov. Приказ #0001 издан 2012-01-01. Приказ #0001 имеет номер 1. Приказ #0001 имеет тип прием на работу.

Такой «текст», записанный в одном из синтаксисов RDF, может быть передан другим системам, которые интерпретируют полученные факты в соответствии со своей логикой, и сохраняют полученную информацию в свои собственные хранилища. Поскольку онтология может не зависеть от внутренних структур каждой ИС, процедуры преобразова-

Интеграция информационных систем с применением семантических технологий

ния получаются универсальными, и исключают зависимость работы (или работоспособности) одной интегрируемой ИС от другой.

Для решения задачи преобразования информации на входе и выходе каждой ИС можно разработать стандартные программные инструменты, которые будут осуществлять работу на основе формализованных правил, и допускать, при необходимости, создание определяемых пользователем процедур.

Управление передачей и обработкой данных

Наиболее интересным вариантом передачи между ИС данных, выраженных в семантической форме, является организация шины передачи сообщений. Такой способ позволяет организовать объединение любого количества ИС, получить широкие возможности настройки и управления процессом передачи.

Для этого необходим центральный сервер, который будет осуществлять маршрутизацию сообщений, необходимые проверки целостности данных, гарантировать доставку, осуществлять обработку ошибок, предоставлять средства настройки, контроля и управления процессом передачи. Сервер может взаимодействовать с интегрируемыми ОС при помощи любых существующих технологий, например, SOAP. В отличие от интеграции ИС при помощи SOAP по методу «точка-точка», в данном случае набор SOAP-методов будет небольшим, и сами определения методов никак не будут связаны с семантикой передаваемой информации. Принципиальная схема взаимодействия двух ИС при помощи такого сервера-посредника показана на рис. 3:

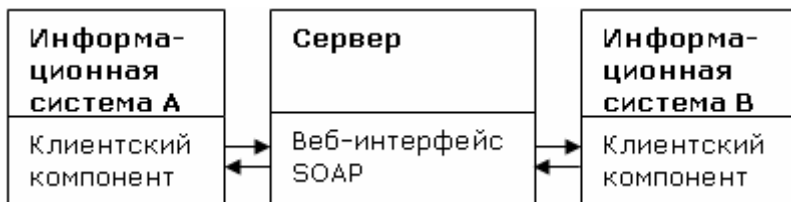


Рис. 3. Схема взаимодействия компонентов обмена

Важным аспектом функциональности сервера является настройка и контроль прав доступа. Для каждого вида объектов, типа свойств необходимо иметь возможность определить, какие ИС могут быть источником информации данного типа, и при каких условиях.

Еще одной важной особенностью предлагаемого нами способа является активная роль клиентских ИС. Каждая ИС-источник должна генерировать информационные сообщения немедленно после изменения

Интеграция информационных систем с применением семантических технологий

каких-либо данных в ней. Этим, в частности, рассматриваемый метод отличается от того, что предлагает стандарт ISO 15926 (части этого стандарта, описывающие собственно процедуры взаимодействия ИС, в настоящее время находятся в стадии утверждения). В этом стандарте описана чем-то похожая схема взаимодействия ИС, в которой, однако, каждая ИС только выставляет имеющуюся у нее информацию на некий «фасад», откуда ее могут запросить другие системы. В ряде применений наш способ, при котором оперативность обновления данных в ИС-корреспондентах гарантируется, будет иметь существенные преимущества. Итак, клиентская ИС должна содержать модуль, или компонент, отвечающий за отслеживание событий, происходящих с данными, и передачу соответствующей информации центральному серверу. Такой модуль может быть стандартным продуктом, взаимодействующим с базой данных клиентской ИС. Принципиальная схема работы клиентского компонента показана на рис. 4:



Рис. 4. Схема работы клиентского компонента

Принципиальная схема работы центрального сервера показана на рис. 5:

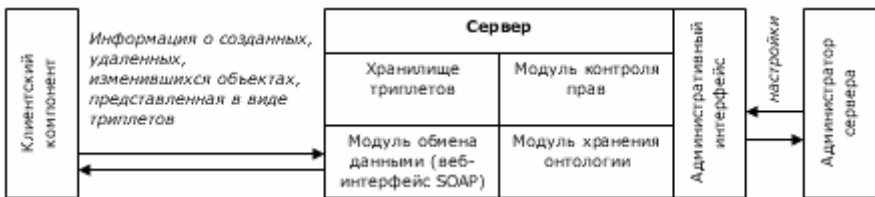


Рис. 5. Схема работы центрального сервера

В целях интеграции нет необходимости постоянно хранить на сервере все передаваемые данные; однако, можно подключить к серверу SPARQL-хранилище, которое будет получать копии всех информационных сообщений, проходящих по шине, и тем самым формировать единый информационный граф, содержащий всю информацию, которой обмениваются ИС. Пользователям это хранилище предоставит аналити-

Интеграция информационных систем
с применением семантических технологий

ческие возможности, которые не в состоянии обеспечить ни одна из интегрируемых ИС в отдельности.

Сравнение с другими способами интеграции

В больших ИТ-инфраструктурах, как правило, для интеграции трех и более информационных систем используются решения класса MDM (Master Data Management), и/или шины обмена сообщениями (Message Queue). Принцип работы MDM-систем состоит в создании хранилища «эталонных данных». Это означает, применительно к нашему примеру, что в собственной базе данных MDM-системы будет храниться информация о каждом сотруднике и приказе, собранная, возможно, из нескольких разных систем в соответствии с определенными правилами и процедурами. Все заинтересованные системы смогут запрашивать у сервера информацию о сотрудниках и приказах.

Такой подход хорошо действует для редко изменяющейся, справочной информации – например, сведениях о сотрудниках или клиентах. Формировать эталонные записи о каждом событии, происходящем в процессе, который поддерживает информационная система (приказ, контакт с клиентом, сделка) – намного сложнее и с точки зрения нагрузки (объема данных и частоты их изменения), и с точки зрения настройки процесса. Предлагаемый нами подход, по принципу распространения информации больше схожий с шиной обмена сообщениями, обещает обеспечить большую эффективность, за счет минимальной нагрузки на сервер, практически мгновенному распространению сообщений о происходящих изменениях.

MDM не подразумевает преобразования информации в семантическую форму. Такое преобразование, между тем, имеет ряд преимуществ, которые облегчают выполнение различных операций с данными. К их числу относится, например, возможность идентифицировать информационные объекты при помощи URI, пространство которых едино для всех взаимодействующих систем. Это облегчает выполнение операций слияния дублей после их выявления, снимает необходимость хранить сопоставление локальных идентификаторов объекта в каждой ИС идентификаторам «золотых записей» MDM.

Классические шины обмена сообщениями, в свою очередь, реализуют только транспортный уровень обмена, предоставляя приложениям самим «договариваться» между собой о форме сообщений и их смысловой нагрузке. Наш способ, в отличие от такого подхода, предоставляет инструментарий для описания семантики передаваемых данных, способов их сопоставления с элементами внутренней информационной структуры каждой из систем (которое происходит на стороне клиент-

Интеграция информационных систем с применением семантических технологий

ских ИС), предоставляет возможность создать средства проверки целостности данных (и, при необходимости, ее восстановления). Таким образом, над транспортным уровнем протокола обмена надстраивается уровень, который можно назвать логическим или семантическим.

Сравнение предлагаемого способа с методами взаимодействия ИС, которые предлагает набор стандартов ISO 15926, было начато выше; кроме уже указанного отличия, следует отметить, что ISO 15926 ориентирован на использование стандартных онтологий, разрабатываемых для различных отраслевых применений, что, на наш взгляд, существенно сужает возможности практического применения средств, в точности соответствующих этому стандарту. Тем не менее, мы не видим принципиальной несовместимости между ISO 15926 и предлагаемым нами способом. Программные продукты, реализующие этот способ, могут быть доработаны для реализации «фасадов» ISO 15926 после того, как применение этого стандарта войдет в практику.

За рамками нашей статьи остались вопросы объединения онтологий, используемых различными информационными системами (семантической интеграции), работы со слабо связанными данными. В этих областях семантические технологии также обеспечивают фундамент для разработки инструментальных средств, решающих различные задачи, возникающие при интеграции информационных систем.

Выводы

- 1. Описана архитектура обмена данными между информационными системами, основанная на передаче сообщений об изменениях в данных посредством центрального сервера.*
- 2. Предложен способ преобразования передаваемой информации в семантическую форму и обратно.*
- 3. Показаны преимущества предлагаемого способа по сравнению с другими технологиями интеграции в крупных ИТ-инфраструктурах.*

Генерация сниппетов в поисковых системах как задача автоматического квазиреферирования

Лиана Ермакова

ПГНИУ, г. Пермь, Россия. liana87@mail.ru

Аннотация. Современные поисковые машины сопровождают ссылки на документы небольшими фрагментами текста – сниппетами, позволяющими принять решение о релевантности документа без его просмотра. В данной статье предложен метод генерации сниппетов, основанный на мультивекторном представлении предложений, сглаживании по локальному контексту, а также системе весовых коэффициентов. Для выбора результирующего фрагмента применяются два алгоритма: (1) отбор предложений моделируется как задача о рюкзаке, которая решается с помощью динамического программирования; (2) скользящее окно поиска фрагмента с максимальным весом.

Ключевые слова: сниппет, информационный поиск, автоматическое реферирование, квазиреферирование, аннотация, задача о рюкзаке, скользящее окно

Введение

Большинство поисковых систем возвращают пользователю в качестве результата ранжированный список документов, изучение которого не представляется возможным в силу его масштабности. В связи с этим современные поисковые машины сопровождают ссылки на документы сниппетами, позволяющими принять решение о релевантности документа без его просмотра. Сниппет представляет собой небольшой отрыв

вок текста, расположенный под результатом поисковой выдачи. Сниппеты могут формироваться как на основе контента веб-страницы, так и метаданных [1]. В идеале сниппет содержит ответ на информационную потребность пользователя. Хороший сниппет должен генерироваться на основе базовых информационных элементов, таких как предложения или сущности XML, быть ограниченным в размерах и отличать данный документ от других документов поисковой выдачи [2]. В рамках данного исследования мы рассматриваем формирование поисковых сниппетов как задачу автоматического квазиреферирования. Предлагаемый подход основан на лингвистическом анализе исходных документов. Мы считаем, что поиск наиболее важной информации не может осуществляться без анализа контекста. Поскольку сниппеты должны быть максимально информативны при ограниченной длине в 1-2 предложения, для формирования результирующего сниппета мы использовали 2 алгоритма: динамическое программирование для решения задачи о рюкзаке [3] и алгоритм скользящего окна. Выбор наиболее подходящих предложений можно смоделировать при помощи задачи о рюкзаке, где весу объектов соответствует количество слов/символов в предложении, а ценности – вычисленная релевантность. Однако наиболее важная информация может оказаться в предложении, чья длина превышает заданный порог. В связи с чем, мы предлагаем использовать алгоритм скользящего окна, максимизирующего вес отрывка вне зависимости от того, совпадает он с границами предложения или нет. При этом может снизиться удобочитаемость сниппета. Поэтому целесообразно установить баланс между информативностью и удобочитаемостью.

Обзор существующих работ

Сниппеты могут формироваться на основе неструктурированных (напр., текстовые документы), полуструктурированных (XML документы) и структурированных данных (онтологии) [4]. Стратегии генерации сниппетов могут различаться в зависимости от типа данных; возможно применение различных эвристик, например, анализ типов узлов XML. Однако в данной работе нас интересуют только методы генерации сниппетов из слабо аннотированных документов. Сниппеты могут быть статическими, т.е. не зависеть от поискового запроса, или же формироваться в зависимости от запроса пользователя. Первые поисковые машины генерировали сниппеты по первым байтам документов. Google первым ориентировал сниппеты на поисковые запросы. В настоящее время считается, что сниппет должен резюмировать содержание документа и включать в себя термины из запроса. В 2009г. компания Yahoo запатентовала метод формирования сниппетов, который сочетает в себе

статический и запросо-ориентированный подходы [5]. Согласно этому методу, статическая релевантность фрагмента представляет собой степень отражения содержания документа и вычисляется по признакам, не зависящим от запроса: положению фрагмента внутри документа, количеству имен, пересечению с заголовком документа и т.д. Зависимость от запроса обуславливается расстоянием между запросом и фрагментом. Для генерации сниппетов применяются традиционные методы расширения запроса, такие как псевдо-обратная связь по релевантности [6–8] и анализ локального контекста [9]. Кроме того, используются частота ключевых слов и расстояние между ними. Поисковик может попытаться определить, хочет ли пользователь попасть в определенное место или же получить информацию по теме. Сниппеты могут содержать лишь часть предложения. В этом случае они обычно ограничены многоточием. Тем не менее, обычно полагают именно предложение минимальной единицей фрагментирования. В веб-документах далеко не всегда соблюдаются нормы пунктуации, поэтому в случае отсутствия явных знаков препинания, в качестве разделителей предложений могут выступать HTML или XML теги. Слишком длинные или слишком короткие предложения часто не считаются "правильными" и отбрасываются [10]. Навигационная информация и реклама также снижают качество сниппетов. Некоторые поисковые машины могут выдавать детализированную информацию для конкретных запросов, например Google предоставляет расширенное описание веб-страниц в формате Microdata, Microformats и RDFa (обзоры, персоналии, товары, организации, события, музыка и т.д.) [11]. Метаданные широко используются для генерации сниппетов, однако поисковые машины могут штрафовать метаданные низкого качества (например, за плохое форматирование, слишком большое количество ключевых слов, избыточную информацию и т.д.) [12]. Поисковые системы могут опираться не на контент страницы, а на собственное представление (например, Directory или DMOZ) [13]. Удобочитаемость сниппета является одним из его ключевых аспектов. Согласно недавно проведенным исследованиям, она влияет на количество переходов по ссылкам [14]. Для оценки удобочитаемости сниппетов Kanungo и Ott предложили использовать градиентное добавление деревьев решений на базе таких признаков как средняя длина слова, среднее количество слогов в словах, доля сложных слов, доля стоп-слов, доля знаков препинания, доля заглавных букв и т.д. Исследования Clarke et al. [15] показали, что наличие терминов из запроса, удобочитаемость и длина URL в существенной мере влияют на переходы по ссылкам. Улучшение удобочитаемости достижимо двумя способами: фильтрация и введение штрафов. В отличие от фильтрации, при введении штрафа фрагменты-кандидаты не исключаются за низкую удобочитаемость, но их вес

уменьшается. Поскольку размер сниппетов мал (обычно пара предложений), перед поисковыми машинами не стоит задача упорядочивания предложений, т.к. это практически не влияет на удобочитаемость.

Описание метода

В предлагаемом методе документ представляется в виде множества предложений. В свою очередь предложение моделируется как кортеж векторов: вектор униграмм, вектор соответствующих частей речи, вектор биграмм, вектор именованных сущностей. Поскольку большая часть компонентов данных векторов равна нулю, используется разреженное представление, т.е. хранятся только элементы, встречающиеся в предложении. Это возможно, т.к. единственная операция – покомпонентное сравнение.

Исследователи отмечают, что наиболее значимая информация содержится в существительных, в то время как вклад стоп-слов практически отсутствует [16]. Поэтому введены весовые коэффициенты, позволяющие ранжировать все части речи: имена собственные и иноязычные слова имеют больший вес, чем имена нарицательные, которые в свою очередь важнее, чем глаголы и прилагательные и т.п. Ранжирование частей речи в частности помогает штрафовать неразрешенную местоименную анафору и др. проблемы удобочитаемости. Структура документа, в т. ч. положение предложения, играет важную роль при генерации сниппетов, что также было отражено в весовых коэффициентах. Расстояние между предложением и запросом определяется как косинус между векторами униграмм и биграмм ($similarity_{unigram}$ и $similarity_{bigram}$ соответственно). Совпадение именованных сущностей рассчитывается по формуле: $NE_{COEF} = \frac{NE_{common}+1}{NE_{query}+1}$, где NE_{common} – количество общих именованных сущностей для запроса и предложения, а NE_{query} – количество именованных сущностей запроса. Предложение может не содержать именованных сущностей, но все равно быть релевантным. Однако без сглаживания, в этом случае коэффициент был бы равен 0, поэтому числитель и знаменатель увеличивается на 1. Во внимание принимались также контекстуальные синонимы, полученные при помощи разрешения анафоры Stanford CoreNLP [17]. Итоговый вес предложения определяется в зависимости от пользовательских настроек как взвешенная сумма или произведение значений полученных метрик.

Мы опираемся на гипотезу, что важность контекста уменьшается по мере удаленности от рассматриваемого предложения. Мы предположили, что для предложений, находящихся на расстоянии более k , значимость контекста равна нулю. Суммарная значимость предложения и его

контекста нормализована до единицы. Таким образом, итоговый вес предложения R_t является взвешенной суммой его веса r_0 и веса соседних предложений r_i :

$$R_t = \sum_{i=-k}^k w_i \times r_i$$

$$w_i = \begin{cases} \frac{1-w_t}{k+1} \times \frac{k-|i|}{k}, & 0 < |i| \leq k \\ w_t, & i = 0 \\ 0, & |i| > k \end{cases}, \quad \sum_{i=-k}^k w_i = 1$$

где w_t – вес целевого предложения, устанавливаемый пользователем, а w_i – веса предложений из контекста. Веса убывают по мере удаления от целевого предложения. Если длина левого или правого контекста меньше k , соответствующие коэффициенты добавляются к целевому предложению w_t . Это позволяет сохранять сумму коэффициентов, равной 1.

Выбор фрагментов

Несмотря на то, что размер сниппета обычно не превышает 150-300 символов, т.е. 1-2 предложения, он должен предоставлять максимум информации о соответствующем документе. Поэтому генерация сниппетов может рассматриваться как задача выбора наиболее важных фрагментов, суммарная длина которых не превышает заданный порог. Это классическая задача комбинаторной оптимизации – **задача о рюкзаке (KS)**: из заданного множества предметов, имеющих стоимость и вес, требуется отобрать некое число предметов таким образом, чтобы получить максимальную суммарную стоимость при одновременном соблюдении ограничения на суммарный вес. В нашем случае вес соответствует количеству слов/символов в предложении, а стоимость – вычисленная релевантность. При этом количество может быть равно 0 или 1. Для решения данной задачи мы использовали алгоритм динамического программирования DP-1 с временем выполнения $O(nc)$, где n – количество объектов, а c – мощность рюкзака.

Применение KS для выбора фрагментов сопряжено с двумя проблемами: (1) если длина всех предложений превышает заданный порог, сниппет будет пустой строкой; (2) алгоритм имеет псевдополиномиальное время выполнения. В связи с этим мы использовали **алгоритм скользящего окна (MW)** для определения фрагментов с максимальной релевантностью. На каждом шаге (1) из фрагмента-кандидата отбрасывается первый токен; (2) в окно добавляются токены справа, пока суммарная длина фрагмента не превышает порог; (3) вычисляется значение релевантности для полученного сегмента. В качестве сниппета выбирается фрагмент с максимальным значением. Наиболее релевантная информация может встретиться в слишком длинном предложении, однако

сниппеты, начинающиеся не сначала предложения обычно менее удобочитаемы. Поэтому для фрагментов, начало которых не совпадает с началом предложений, был введен постоянный штраф.

Оценка результатов

Для оценки результатов использовалась методика, применявшаяся в соревновании по генерации сниппетов, проходящей на форуме INEX. Цель – определить, насколько полно сниппет отражает содержание документа. Ассессоры должны были сначала оценить релевантность документов исходя только из соответствующих сниппетов, а затем – на основе содержимого самих документов. При этом 1 соответствовала релевантному документу, а 0 – нерелевантному. Оценка релевантности, произведенная только с учетом сниппетов, была сопоставлена с релевантностью соответствующих документов посредством следующих метрик:

- Средняя точность прогноза (MPA) – доля правильно оцененных объектов: $MPA = \frac{TP+TN}{(TP+FN+TN+FP)}$, где TP – количество истинно-положительных срабатываний, TN – истинно-отрицательных, FN – ложно-отрицательных, а FP – ложно-положительных.
- Полнота (R): $R = \frac{TP}{(TP+FN)}$.
- Отрицательная полнота (NR): $NR = \frac{TN}{(TN+FP)}$.
- Положительное согласие (PA): $PA = 2 * \frac{TP}{(2*TP+FP+FN)}$.
- Отрицательное согласие (NA): $NA = 2 * \frac{TN}{(2*TN+FP+FN)}$.
- Средняя нормализованная точность прогноза (MNPA): $MNPA = 0.5 * \frac{TP}{(TP+FN)} + 0.5 * \frac{TN}{(TN+FP)}$.
- Геометрическое среднее (GM) R и NR: $GM = \sqrt{R * NR}$

В 2011г. мы не принимали участие в соревновании по генерации сниппетов, но мы сравнили наши результаты с результатами участников. Тестовые данные были представлены материалами англоязычного дампа Википедии (ноябрь 2011г.), а также 50 запросами, каждый из которых помимо собственно поискового запроса содержал неформальное описание информационной потребности пользователя. Каждому запросу соответствовал ранжированный список из 10 документов и соответствующим им сниппетов (официальные результаты базировались на списке из 50 документов). Результаты, полученные на этих данных, представлены в таблице 1. Оба метода KS и MW показали результаты выше, чем системы-участники, однако это может быть связано с уменьшением количества оцениваемых документов с 50 до 10. При этом

KS был оценен выше, чем MW, а корреляция между ними, вычисленная при помощи коэффициента $\varphi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\cdot}n_{\cdot 0}n_{\cdot 1}n_{\cdot 0}}} = 0.68$. В 2012г. количество оцениваемых запросов было уменьшено с 50 до 35, а количество оцениваемых сниппетов – до 20 на каждый запрос [18]. Кроме того, длина сниппетов была сокращена до 180 символов, что негативно сказалось на результате. Официальные результаты приведены в таблице 2. В отличие от 2011г., KS показал результаты хуже, чем MW. Это, прежде всего, объясняется уменьшением размера сниппетов, т.к. KS лучше работает для более длинных предложений. Оценка KS незначительно отличается от оценки MW, что также связано с тем, что при невозможности выбрать релевантный фрагмент, длина которого не превышает заданного порога, применялся MW.

Таблица 1. Результаты неофициальной оценки INEX 2011

Run	MPA	MNPA	R	NR	PA	NA	GM
KS	0.81	0.81	0.76	0.86	0.80	0.83	0.81
MW	0.76	0.75	0.63	0.87	0.72	0.79	0.74
Best_2011	0.7582	0.643	0.4641	0.8219	0.3748	0.8292	0.5705

Таблица 2. Официальные результаты конкурса INEX 2012

Run	MPA	MNPA	R	NR	PA	NA	GM
Best_2012	0.7443	0.6844	0.5071	0.8476	0.5292	0.7685	0.6121
MW	0.7100	0.6487	0.4881	0.8356	0.4895	0.6872	0.5171
KS	0.7314	0.6474	0.3906	0.8869	0.4076	0.7661	0.5039
Worst_2012	0.7500	0.6164	0.3092	0.9405	0.3726	0.7899	0.4063

Выводы

В статье предложен метод генерации сниппетов на основе мультивекторного представления предложений, сглаживания по локальному контексту, а также системы весовых коэффициентов. Сглаживание по локальному контексту применимо вне зависимости от национального языка. Сравнение именованных сущностей также возможно для всех языков, однако в настоящее время наиболее распространены инструменты для извлечения именованных сущностей из англоязычных текстов. При замене английского языка другим требуется введение новых весовых коэффициентов для частей речи, т.к. системы частей речи разных языков могут существенно отличаться. Для выбора результирующего фрагмен-

та текста применялись два алгоритма: динамического программирования для решения задачи о рюкзаке и скользящее окно поиска фрагмента с максимальным весом. Было проведено 2 серии испытаний. В первом случае мы сравнили наши результаты с результатами участников соревнования по генерации сниппетов на форуме INEX 2011, где предложенная система показала результат, значительно превышающий лучшую систему. В 2012 г. мы приняли официальное участие в INEX. Результаты оказались хуже, чем в 2011г., что объясняется уменьшением размера сниппетов почти в 2 раза. MW, показавший существенно более низкие результаты на первой коллекции, превзошел KS, согласно официальным данным 2012г. Это связано с тем, KS лучше работает для более длинных предложений. Планируется использование методов расширения запроса и изучение влияния параметров на результаты. В настоящее время ведется разработка новых методов взвешивания предложений.

Список источников

1. Turpin A. et al. Fast generation of result snippets in web search // Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. Amsterdam: ACM, 2007. P. 127–134.
2. Huang Y., Liu Z., Chen Y. eXtract: a snippet generation system for XML search // Proc. VLDB Endow. 2008. Vol. 1, № 2. P. 1392–1395.
3. Kellerer, Hans, Pferschy, Ulrich, Pisinger, David. Knapsack problems. Springer-Verlag, Berlin, 2004. 546 p.
4. Penin T. et al. Snippet Generation for Semantic Web Search Engines // Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web. Bangkok, Thailand: Springer-Verlag, 2008. P. 493–507.
5. United States Patent Application: 0090292683. URL: [http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetachtml%2FPTO%2Fsearch-adv.html&r=1&p=1&f=G&l=50&d=PG01&S1=20090292683.PGNR.&OS=dn/20090292683&RS=DN/20090292683](http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetachtml%2FPTO%2Fsearch-adv.html&r=1&p=1&f=G&l=50&d=PG01&S1=20090292683.PGNR.&OS=dn/20090292683&RS=DN/20090292683) (accessed: 08.10.2012).
6. Leal L., Scholer F., Thom J. RMIT at INEX 2011 Snippet Retrieval Track // Focused Retrieval of Content and Structure, 10th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2011), Geva, S., Kamps, J., Schenkel, R. (Eds.). Lecture Notes in Computer Science, Springer. 2012.

7. Wang S., Hong Y., Yang J. PKU at INEX 2011 XML Snippet Track // Focused Retrieval of Content and Structure, 10th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2011), Geva, S., Kamps, J., Schenkel, R. (Eds.). Lecture Notes in Computer Science, Springer. 2012.
8. Ko Y., An H., Seo J. Pseudo-relevance feedback and statistical query expansion for web snippet generation // Inf. Process. Lett. 2008. Vol. 109, № 1. P. 18–22.
9. Sanderson M. Accurate user directed summarization from existing tools // Proceedings of the seventh international conference on Information and knowledge management. Bethesda, Maryland, United States: ACM, 1998. P. 45–51.
10. Kupiec J., Pedersen J., Chen F. A trainable document summarizer // Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. Seattle, Washington, United States: ACM, 1995. P. 68–73.
11. Rich snippets (microdata, microformats, and RDFa) – Webmaster Tools Help. URL: <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=99170> (accessed: 08.10.2012).
12. Anatomy of a Google Snippet. URL: <http://searchengineland.com/anatomy-of-a-google-snippet-38357> (accessed: 08.10.2012).
13. How a Search Engine May Choose Search Snippets – SEO by the Sea. URL: <http://www.seobythesea.com/2009/12/how-a-search-engine-may-choose-search-snippets/> (accessed: 08.10.2012).
14. Kanungo T., Orr D. Predicting the readability of short web summaries // Proceedings of the Second ACM International Conference on Web Search and Data Mining. Barcelona, Spain: ACM, 2009. P. 202–211.
15. Clarke C.L.A. et al. The influence of caption features on click-through patterns in web search // Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. Amsterdam, The Netherlands: ACM, 2007. P. 135–142.
16. Silber H.G., Mccoy K.F. Efficiently computed lexical chains as an intermediate representation for automatic text summarization // Computational Linguistics – Summarization. 2002. Vol. 28, № 4. P. 1–11.
17. The Stanford NLP (Natural Language Processing) Group. URL: <http://nlp.stanford.edu/software/corenlp.shtml> (accessed: 20.02.2013).
18. Trappett M. et al. Overview of the INEX 2012 Snippet Retrieval Track // Focused Retrieval of Content and Structure / ed. Geva S., Kamps J., Schenkel R. Springer Berlin Heidelberg, 2013 (to appear).

Об одном подходе к локализации антропометрических точек

Александр Шушарин¹, Константин Черенков², Александр Гаврилюк³,
Андрей Валик⁴

¹ООО «3DiVi», Екатеринбург, Россия. shusharin.alex@gmail.com

²ООО «3DiVi», Екатеринбург, Россия. k.cherenkov@gmail.com

³ООО «3DiVi», Екатеринбург, Россия. alexander.gavriliouk@gmail.com

⁴ООО «3DiVi», Миасс, Россия. vav@3divi.com

Аннотация. В статье описан подход к локализации антропометрических точек, используемых в методах идентификации/верификации личности по лицу. Подход основан на комбинации метода модели активного контура (ASM) с каскадами хааровских классификаторов и случайным лесом деревьев решений, обученных на двоичных дескрипторах FREAK. Проведено сравнение с результатами аналогичных исследований.

Ключевые слова: биометрия; идентификация по лицу; верификация по лицу; метод активного контура; active shape model; случайный лес деревьев решений; random forest trees; FREAK.

Введение

Задача распознавания личности человека по изображению лица является нетривиальной проблемой компьютерного зрения, привлекающей внимание многих специалистов в течение последних, по крайней мере, 30 лет [1]. Решению задачи препятствует зависимость обычного изображения (цветного или в оттенках серого) от геометрии лица и от

трудно контролируемых факторов съемки: особенностей отражения света от кожи, ориентации лица относительно камеры и освещения.

Появившиеся относительно недавно компактные устройства (сенсоры глубины или 3D-сканеры) получения так называемых 2.5D-изображений (карт глубины, *depth map*), каждый пиксель которых кодирует расстояние от камеры до точки сцены, открывают новые возможности для идентификации по изображению лица, поскольку 2.5D-изображения практически инвариантны к неравномерности освещения и содержат больше информации о геометрии лица, чем обычные изображения. Технические характеристики этих устройств уже можно считать достаточными для успешного решения задачи: сенсоры имеют разрешение от 640x480 до 1280x960 точек, в том числе разрабатываемый в компании 3DiVi сенсор глубины имеет разрешение 1280x960 точек при 60 кадрах в секунду и дальности съемки до 5 м.

Обзор существующих методов распознавания лиц (как по 2D, так и по 2.5D) можно найти в недавней статье [2]. Мы лишь отметим, что эти методы могут как анализировать локальные особенности лица, так и рассматривать все изображение как многомерный вектор-наблюдение. Но, пожалуй, во всех случаях обязательным этапом, предваряющим классификацию, является выравнивание лица, под которым понимается выравнивание лица во фронтальное положение относительно камеры или приведение совокупности лиц (например, в обучающей выборке для обучения классификатора) к единой системе координат. Для реализации этого этапа необходима локализация на изображении характерных для всех лиц *антропометрических* точек – чаще всего это центры зрачков или уголки глаз. Разные исследователи выделяют разные группы таких точек. Из-за ограничения объема статьи мы вновь отсылаем читателя к работе [2], в которой кроме исторического обзора проведен анализ дискриминационных свойств групп точек, т.е. их важность с точки зрения распознавания лиц. На рис. 1, заимствованном из [2], отмечены 10 наиболее важных (по результатам того исследования) точек.

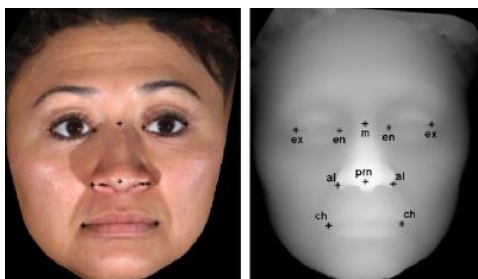


Рис. 1. Антропометрические точки, отобранные в [2].

Таким образом, локализация антропометрических точек необходима как предварительный этап некоторых алгоритмов распознавания лиц. Кроме того, она может представлять интерес для распознавания эмоций по лицу человека или управления «аватаром» (копирование мимики лица человека на моделируемое лицо).

В статье описываются подход для локализации указанных 10 точек (в принципе, расширяемый до большего количества), условия и результаты его тестирования и сравнения с альтернативными методами.

Предлагаемый подход

В основе нашего подхода лежит классический метод модели активного контура (active shape model, ASM) [3]. Подробное описание этого метода и его вариаций, в т.ч. для анализа изображений лиц, можно найти, например, в [4], здесь мы опишем его упрощенный вариант.

Идея метода ASM применительно к нашей задаче заключается в учете статистических связей между совместным расположением антропометрических точек. Пусть имеется обучающая выборка из L изображений лиц, снятых в анфас, с размеченными (экспертом) антропометрическими точками, N точек на каждом лице, все точки пронумерованы в одинаковом порядке. Пусть $(x_{ij}, y_{ij}), j = \overline{1, N}$ – координаты точек (в системе координат изображения), отмеченных на i -м лице из обучающей выборки, $i = \overline{1, L}$. Для того чтобы привести координаты на всех изображениях к единой системе обычно выполняется т.н. обобщенный прокрустов анализ, см. [4]. В простейшем случае, если все лица сняты с одинаковым масштабом, можно ограничиться центрированием. Далее будем считать, что координаты центрированы. Составим L векторов высоты $2N$, описывающих «форму» или «контур» расположения точек:

$$s_i = (x_{i1} \ \dots \ x_{iN} \ y_{i1} \ \dots \ y_{iN})^T.$$

Пусть $\bar{s} = \frac{1}{L} \sum_{i=1}^L s_i$ – средняя форма, и положим $\bar{s}_i = s_i - \bar{s}, i = \overline{1, L}$. Далее, по совокупности векторов $\{\bar{s}_i\}$ вычисляется матрица ковариации их координат $K = \sum_{i=1}^L \bar{s}_i \bar{s}_i^T$.

Пусть λ_i – все собственные значения матрицы K , упорядоченные в порядке убывания, с соответствующими собственными векторами $u_i, i = \overline{1, 2N}$. Совокупность векторов u_i является базисом $2N$ -мерного векторного пространства так, что всякий вектор \bar{s}_i может быть представлен в виде их линейной комбинации. В силу статистических связей между расположением точек это представление может быть заменено следующим приближенным [3]:

$$s_i \approx \bar{s} + b_{i,1}u_1 + b_{i,2}u_2 + \dots + b_{i,p}u_p = \bar{s} + \Phi b_i,$$

где матрица Φ составлена из p главных компонент – собственных векторов $u_j, j = \overline{1, p}$, которые отвечают p наибольшим собственным значениям λ_j , а b_i – вектор из p коэффициентов, называемых параметрами модели. Число p выбирается из условия $\sum_{j=1}^p \lambda_j / \sum_{j=1}^{2N} \lambda_j > 0.98$, [3].

Модель ASM определяется матрицей Φ и вектором средней формы \bar{s} . Всякая форма может быть приближенно описана с помощью модели и параметров, определяемых из соотношения $b_i = \Phi^T \bar{s}_i = \Phi^T (s_i - \bar{s})$. Можно считать, что средняя форма \bar{s} «отвечает» за общую закономерность расположения точек, а индивидуальные особенности конкретной формы выражаются небольшим количеством параметров модели.

Локализация точек на новом, не входящем в обучающую выборку, изображении лица осуществляется следующим образом. Прежде всего, мы уточняем положение лица на изображении с помощью каскадного классификатора Виолы – Джонса, см. [5], который возвращает окно с лицом. С центром этого окна совмещается средняя форма \bar{s} , координаты которой умножаются на масштабный коэффициент μ , пропорциональный ширине окна. Размещенная таким образом средняя форма определяет начальное приближение к положению антропометрических точек. В системе координат изображения соответствующую форму обозначим через $t^{(0)}$. Далее, мы будем итеративно уточнять положение точек так, что можно считать, что $t^{(0)}$ – это форма на 0-й итерации.

Предварительно для каждой антропометрической точки с номером i обучается каскадный классификатор C_i типа Виолы – Джонса [5]. Для изображения C_i возвращает множество точек, классифицированных как антропометрические точки с номером i . При обучении классификатора положительными примерами являются регионы изображения с центром в антропометрической точке, а отрицательными примерами – регионы, пересекающиеся с положительными примерами, см. рис. 2.

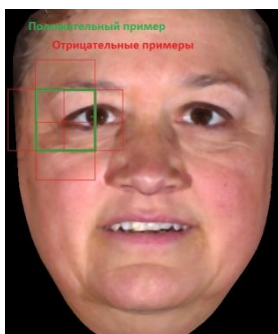


Рис. 2. Обучающие примеры для каскадного классификатора.

На j -й итерации алгоритма, $j = 1, \dots$, для i -й антропометрической точки, $i = \overline{1, N}$, соответствующий каскадный классификатор C_i применяется к небольшой области изображения с центром в точке $(t_i^{(j-1)}, t_{i+N}^{(j-1)})$. Поскольку классификатор обычно возвращает несколько точек, классифицируемых как антропометрические, мы полагаем верно найденной точку, ближайшую к точке $(t_i^{(j-1)}, t_{i+N}^{(j-1)})$.

Пусть \tilde{t} – форма, составленная по точкам, найденным каскадными классификаторами на j -й итерации, и координаты которой центрированы и поделены на масштабный коэффициент μ . Эта форма проверяется на соответствие статистической модели ASM: определяется вектор параметров $\tilde{b} = \Phi^T(\tilde{t} - \bar{s})$, координаты которого затем ограничиваются по правилу

$$b_k = \begin{cases} \tilde{b}_k, & \text{если } |\tilde{b}_k| < \alpha \lambda_k \\ \alpha \lambda_k \text{sign}(\tilde{b}_k), & \text{в противном случае} \end{cases}$$

где $k = \overline{1, p}$ и α обычно полагают равным 2 или 3, см. [3,4].

Параметры b модели определяют форму $t^{(j)}$ для следующей итерации. Процедура локализации точек завершается через фиксированное количество итераций (в наших экспериментах мы ограничивались 3-мя).

В нашем подходе итеративная процедура повторяется в два этапа. На первом этапе мы применяем по описанной процедуре модель ASM, состоящую из 25 точек. На втором – по той же процедуре модель ASM из 10 точек. Такой подход дает лучшие результаты (см. следующий раздел), чем применение одной модели ASM с 10 или 25 точками. Это может быть объяснено следующим образом: модель с 25 точками сходится хуже, чем модель с 10 точками, поскольку сложнее, в тоже время модель с 10 точками менее устойчива в том смысле, что в процессе итераций форма может «съезжать», что приводит к большим ошибкам в локализации точек. Применение сначала сложной модели с 25 точками позволяет достаточно хорошо и устойчиво приблизиться к верному расположению точек, а последующее применение более простой модели с 10 точками обеспечивает сходимость алгоритма. Данное наблюдение напоминает также используемый на практике подход с применением моделей ASM для пирамиды изображений [4]. Заметим, что каскадные классификаторы C_i обучались отдельно для каждой модели (для модели с 25 точками размер примеров больше, чем для модели с 10 точками).

Заключительным этапом в локализации точек является применение еще одного классификатора – случайного леса регрессионных деревьев решений, см. [6], обученных на двоичных дескрипторах областей изображений FREAK, см. [7]. Подаваемой на вход области изображения алгоритм FREAK ставит в соответствие двоичный вектор – дескриптор,

который в определенной степени инвариантен к масштабу, повороту и наличию шума в изображении, что позволяет искать подобную область на других изображениях. В статье [7] авторов FREAK поиск предлагается осуществлять сравнением дескрипторов в метрике Хэмминга.

В нашем исследовании областями интереса являются области, содержащие антропометрическую точку, а сравнение дескрипторов мы проводили с помощью случайного леса регрессионных деревьев решений. Для их обучения использовалась обучающая выборка из дескрипторов положительных и отрицательных примеров, сформированных по тому же принципу, что и для обучения каскадных классификаторов C_i . Случайные леса деревьев решений применялись к небольшим областям вокруг точек, найденных на последней итерации второй модели ASM.

Обучение и тестирование модели

Для обучения и тестирования предложенного подхода мы использовали базу лиц T3FRD, подробное описание которой можно найти в [8]. Она включает по 1149 2D- и 2.5D-изображений 751x501 пикселей 116 субъектов разного пола, возраста и этнической принадлежности. На всех изображениях экспертами отмечены 25 точек (включая точки с рис. 1), 2D- и 2.5D-изображения совмещены друг с другом. Карты глубины в базе T3FRD получены с помощью лазерного сканера высокого разрешения. На наш взгляд, потребительские сенсоры, используемые, например, в игровых приставках, обладают высоким уровнем шума, и получаемые с них карты глубины непригодны для локализации антропометрических точек без предобработки (собственно говоря, нам не известны наборы тестовых данных, полученные на потребительских моделях сенсоров).

Отметим, что описанный в предыдущем разделе статьи подход может быть применен к любому типу изображения – к картам глубины или к изображениям в оттенках серого. В таблицах 1 и 2 приведены результаты тестирования нашего подхода для двух вариантов обучения. В первом варианте (табл. 1) каскадные классификаторы обучались на изображениях, во втором (табл. 2) – на картах глубины. В обоих вариантах случайные леса деревьев решений обучались на дескрипторах FREAK, вычисленных как на изображениях в оттенках серого, так и на картах глубины (отклики случайных лесов для изображений двух типов суммировались с последующим выбором наиболее правдоподобной точки).

В обоих вариантах метод обучался на 574 изображениях из базы T3FRD, тестировался на 575. Выборки не пересекаются по субъектам. Ошибка локализации каждой точки измеряется в мм (на карте глубины

Об одном подходе к локализации антропометрических точек

0.32 мм в 1 пикселе, см. [8]) как расстояние между экспертной разметкой и результатом метода.

В таблицах 1 и 2 также приведены результаты применения модели ASM из 25 точек и комбинации этой модели с моделью ASM из 10 точек. Для сравнения в таблице 3 приведены результаты тестирования (на той же базе) метода из работы [9] для трех вариантов его обучения: только по изображениям, только по картам глубины и смешанное обучение по изображениям и картам глубины.

Для каждого метода и варианта его обучения в левой колонке указано среднее значение ошибки (мм), в правой – СКО (мм). В последней строке таблицы указано время работы алгоритмов (мс) на одном изображении и компьютере комплектации Intel Core i7 3.4GHz, 8Gb RAM. Разработанные нами алгоритмы были реализованы с использованием библиотеки компьютерного зрения OpenCV.

На рис. 3 проиллюстрированы этапы локализации нашим методом (в порядке слева направо и сверху вниз): средняя форма, совмещенная с лицом (в обозначениях выше $t^{(0)}$); результат применения модели ASM из 25 точек (желтые точки – экспертная разметка, зеленые – модель ASM, красные – точки, найденные каскадными классификаторами на последней итерации); результат применения модели ASM из 10 точек (желтые точки – экспертная разметка, красные – модель ASM); окончательно определенное положение точек лесами деревьев решений (желтые точки – экспертная разметка, синие – результат работы метода).

Табл. 1. Ошибки локализации антропометрических точек, 1-й вариант обучения (на основе изображений).

Точка, рис. 1	ASM, 25 точек		ASM, 25+10 точек		Весь подход	
ex, левая	1.66	0.98	1.39	0.87	1.20	0.76
ep, левая	1.90	1.08	1.25	0.91	1.14	0.73
ex, правая	1.65	1.04	1.40	0.98	1.29	1.01
ep, правая	1.93	1.14	1.32	0.90	1.18	0.74
al, левая	1.36	0.74	1.05	0.62	0.99	0.56
al, правая	1.43	0.82	1.10	0.71	1.01	0.68
ch, левая	1.69	1.29	1.42	1.22	1.39	1.29
ch, правая	1.61	1.44	1.39	1.35	1.39	1.52
m	1.98	1.26	1.88	1.33	1.80	1.23
rpm	1.43	0.85	1.39	0.82	1.34	0.83
Время, мс	18		30		65	

Табл. 2. Ошибки локализации антропометрических точек, 2-й вариант обучения (на основе карт глубины).

Точка, рис. 1	ASM, 25 точек		ASM, 25+10 точек		Весь подход	
ex, левая	2.21	1.10	1.73	0.99	1.19	0.75
ep, левая	2.40	1.49	1.94	1.21	1.22	0.78
ex, правая	2.19	1.35	1.86	1.24	1.30	1.01

Об одном подходе к локализации антропометрических точек

ep, правая	2.05	1.32	1.74	1.06	1.23	0.75
al, левая	1.21	0.66	1.03	0.57	0.99	0.55
al, правая	1.64	0.84	1.18	0.75	1.01	0.69
ch, левая	2.14	1.44	1.82	1.34	1.36	1.22
ch, правая	2.12	1.73	1.83	1.73	1.44	1.85
m	1.87	1.13	1.70	1.04	1.79	1.22
rgn	2.15	1.27	1.78	1.01	1.35	0.84
Время, мс	18		30		65	

Табл. 3. Ошибки локализации антропометрических точек из [9].

Точка, рис. 1	На изображениях		На картах глубины		Смешанное обучение	
ex, левая	1.83	2.8	3.93	4.5	1.47	1.8
ep, левая	1.45	1.9	1.65	1.7	1.17	1.0
ex, правая	1.49	1.9	3.77	4.5	1.37	1.3
ep, правая	1.35	1.5	1.63	1.5	1.09	1.0
al, левая	1.26	1.3	1.04	0.7	1.01	0.7
al, правая	1.18	1.2	0.96	0.9	0.92	0.7
ch, левая	1.45	1.7	1.89	1.5	1.31	1.1
ch, правая	1.64	2.0	1.81	1.9	1.35	1.2
m	3.56	3.6	2.67	1.9	2.4	1.07
rgn	1.26	0.9	1.40	0.9	1.18	0.8
Время, мс	300		300		600	

Выводы

Методы локализации антропометрических точек, предлагаемые в большом количестве в литературе, непросто сравнивать из-за различных условий тестирования и измерения ошибок. Однако можно считать, что точность, достигнутая в [9] (предлагаемый в [9] метод основан на корреляционном анализе откликов банка фильтров Габора) и оцененная в мм на базе T3FRD, достаточна для решения задачи распознавания по лицу [2]. В настоящей работе мы добились сопоставимых по точности результатов, см. табл. 1, при существенном сокращении времени локализации, которое также является критически важной рабочей характеристикой системы распознавания.

Основными особенностями нашего подхода являются: использование двух моделей ASM с различным количеством точек, что позволяет повысить точность и устойчивость метода по сравнению с одной моделью ASM, а также заключительная коррекция положения точек с помощью случайных лесов регрессионных деревьев решений, обученных на двоичных дескрипторах FREAK. Высокая скорость работы метода обеспечивается применением на этапе подбора параметров модели ASM каскадных классификаторов по типу Виолы – Джонса.



Рис. 3. Этапы процесса локализации.

Список источников

1. Zhao W., Chellappa R., Phillips P.J., Rosenfeld A. Face Recognition: a literature survey // ACM Computing Surveys, Vol. 35, No. 4, 2003, pp. 399-459.
2. Gupta S., Markey M.K., Bovik A.C. Anthropometric 3D Face Recognition // International Journal of Computer Vision (2010), Online First: <http://dx.doi.org/10.1007/s11263-010-0360-8>
3. Cootes T.F., Taylor C.J., Cooper D.H., Graham J. Active Shape Models – Their Training and Application // Computer Vision and Image Understanding, Vol. 61, No.1, 1995, pp. 38-59.

4. Cootes T.F. Model-Based Methods in Analysis of Biomedical Images // Image Processing and Analysis (Chapter 7), Oxford University Press, 2000, pp. 223-248.
5. Viola P., Jones M.J. Robust real-time face detection // International Journal of Computer Vision, Vol. 57, No. 2 (2004), pp. 137-154.
6. Breiman L. Random Forests // Machine Learning, Vol. 45, No 1, 2001, pp. 5-32.
7. Alahi A., Ortiz R., Vandergheynst P. Fast Retina Keypoint // CVPR2012.
8. Gupta S., Castleman K.R., Markey M.K., Bovik A.C. Texas 3D Face Recognition Database // Image Analysis & Interpretation, 2010 IEEE Southwest Symposium, pp. 97-100.
9. Jahanbin S., Hyohoon Choi, Bovik A.C. Passive Multimodal 2D+3D Face Recognition // IEEE Transactions on Information Forensics and Security, Vol. 6, No 4, 2011, pp. 1287-1304.

Использование ресурсов Интернета для построения таксономии

Екатерина Черняк, Борис Миркин

Отделение Прикладной Математики и Информатики
Национальный Исследовательский Университет – Высшая Школа Экономики

Аннотация. В работе предложен двухшаговый подход к построению предметных таксономий на русском языке. На первом шаге строятся высокие уровни таксономии на основе паспортов специальностей ВАК. На втором шаге таксономические темы последовательно достраиваются новыми темами, извлеченными и отфильтрованными из дерева категорий и статей русского сегмента Википедии. Во всех расчетах используется мера сходства между строкой и текстом, основанная на аппарате аннотированных суффиксных деревьев.

Ключевые слова: Достраивание таксономий, мера сходства строки тексту, Википедия, суффиксное дерево

Введение

Таксономии, или иерархические онтологии, – это популярный инструмент для представления, хранения и использования знаний о какой-либо предметной области [1,2]. Таксономия представляет собой корневое дерево, организованное как иерархия понятий, или тем, узкой предметной области. В таком дереве темы находятся в отношении «А – часть В» или «А – более общее понятие, чем В». Автоматическое построение таксономий – важная задача, которая относится и к области автоматиче-

ской обработки текстов, и к области информационного поиска [3,4]. Наиболее популярные подходы к решению этой задачи предполагают использование больших коллекций неструктурированных текстов, относящихся к рассматриваемой предметной области. Из этих текстов извлекают ключевые слова и словосочетания, находящиеся в четко выраженном отношении «наследования», то есть, образующие искомую иерархию понятий. Недостатки такого подхода к построению таксономий хорошо известны: 1) не для каждой предметной области можно найти достаточно большую коллекцию неструктурированных текстов, 2) методы обнаружения семантических отношений между словами далеки от совершенства, поэтому охваченные темы и структура таксономии, как правило, неудовлетворительны [5]. Поэтому, в качестве замены коллекциям текстов предлагают использовать Интернет-ресурсы, например, Википедию [6]. Более того, Википедия устроена таким образом, что уже имеет некоторые задатки таксономии, например, иерархию тем (дерево категорий) и огромное количество понятий (названия статей и категорий). Однако, оказывается, что дерево категорий Википедии нельзя использовать само по себе как таксономию, поскольку качество некоторых статей или фрагментов дерева категорий вызывает сомнение в силу не профессиональности части авторов Википедии.

В данной статье представлен полуавтоматический метод построения таксономии предметной области. Он состоит из двух этапов. На первом этапе строят основу таксономии, ее два или три верхних уровня, в соответствии с официальными документами, формализующими рассматриваемую предметную области. На втором этапе происходит пошаговое достраивание тем таксономии фрагментами дерева категорий и статей из русского сегмента Википедии, очищенных от лишних тем, шума. Во всех расчетах используется мера сходства темы тексту, основанная на аннотированных суффиксных деревьях. Метод достраивания таксономии состоит из нескольких шагов. Для каждой из тем верхнего уровня: 1) находим соответствующее теме поддерево дерева категорий; 2) определяем релевантность теме статей Википедии, принадлежащих к выбранному поддереву; 3) проводим очистку поддерева от иррелевантных статей и категорий; 4) извлекаем ключевые слова и словосочетания из оставшихся после очистки статей; 5) достраиваем тему таксономии оставшимися категориями и статьями; 6) помещаем извлеченные слова и словосочетания на последний уровень таксономии в качестве уточняющих описаний листовых тем. Таксономия, построенная таким образом, отличается от большинства таксономий тем, что она сбалансирована как и по глубине, так и по числу детей и листьев в разных разделах таксономии, поскольку эти параметры контролируются в ходе построения. Ниже приведено подробное описание метода и его применение к

математической предметной области «Теория вероятностей и математическая статистика», иллюстрирующее и достоинства, и недостатки метода.

Потребность в построении таксономий математических областей вызвана нашей предыдущей работой по использованию таксономий для визуализации и интерпретации аннотаций математических статей и учебных программ по математике и информатике. Единственная доступная таксономия математики на русском языке – это рубрикатор РЖ «Математика», модифицированный последний раз в 1999 г.. Стоит заметить, что таксономия не только устарела, но и нелогична и несбалансирована. Так, например, в ней отсутствует одна из основных тем – «Дискретная математика», под темой «Дифференциальные уравнения» находятся более 80 других тем, а под более современной темой «Теория игр» – всего 10.

К счастью, существуют и другие русскоязычные классификации научных тем, например, классификации ВАКа представляющие собой двухуровневые деревья, покрывающие все научные темы. Еще несколько уровней достаточно общей классификации можно извлечь из паспортов специальностей ВАК. Однако, для того, чтобы дойти до таких частных понятий, как «производная» или «функция», требуется еще от двух до четырех уровней менее общих понятий.

Отсюда вытекает постановка рассматриваемой задачи. Нам требуется метод достраивания таксономии на основе ресурсов Википедии. На выходе этого метода должно получаться более или менее сбалансированное дерево, глубина и число детей на разных уровнях в разных разделах не сильно бы различалось. Дополнительное требование к таксономии заключается в наличии уточнений, то есть, множества слов или словосочетаний, объясняющих листовые темы.

Требованиям сбалансированности и наличия уточнений удовлетворяет классификационная система Ассоциации Вычислительной техники АСМ-CCS, которую по праву можно считать золотым стандартом таксономии.

Методам достраивания таксономии посвящено несколько работ. В исследовании [8] используются результаты поиска шаблонных запросов вида “А состоит из ...” . В результате поиска такого словосочетания предполагается получить множество понятий, которые можно рассматривать как потенциальные подтемы понятия А. Можно использовать и уже существующие онтологии и таксономии для достраивания исходной. Если и достраиваемая таксономия, и используемая в качестве источника онтология или таксономия описаны формальными моделями языка OWL, эта задача решается с помощью агрегации и введения новых логических отношений, как это сделано в работе [9]. В целом, для

доставления таксономии могут быть выбраны как структурированные, так и неструктурированные источники. Во многих исследованиях [10-12] используется компромиссное решение: данные для доставления таксономии извлекают из полуструктурированной интернет-энциклопедии Википедии. Многообразии данных в Википедии (инфобоксы, тексты статей, дерево категорий) и универсальность тематики энциклопедии позволяют использовать ее для построения таксономий и онтологий разных предметных областей. В работе [5] приведены и другие аргументы в пользу Википедии как источника данных для построения таксономии:

- Википедия постоянно обновляется, поэтому таксономию легко обновлять и пополнять;
- Википедия многоязычна, и, скорее всего, методика построения таксономий, разработанная для одного языка подойдет и для другого.

В работах [10-12] представлены различные способы построения или пополнения таксономий на основе ресурсов Википедии. В [10] в качестве источника таксономических тем использованы тексты статей, в [11] – дерево категорий Википедии, а в [12] – инфобоксы. Основное отличие нашего подхода заключается в использовании и структурированных данных – фрагментов дерева категорий Википедии, и неструктурированных текстов статей, что позволяет нам следовать золотому стандарту ACM-CCS. Мы специально решили ограничиться такой узкой предметной областью, как теория вероятностей и математическая статистика, чтобы, во-первых, избавиться от проблем обработки больших объемов данных из Википедии, во-вторых, получить на выходе таксономию разумного размера и иметь возможность скорректировать ее вручную. Кроме того, использование структурированных данных, таких как дерево категорий Википедии, в качестве источника тем для доставления, позволяет избежать трудностей с поиском шаблонов и низкой точностью результатов такого поиска.

Метод доставления таксономии на основе ресурсов Википедии

Мы задали основу таксономии, основываясь на паспортах специальностей ВАК [14]. В этих паспортах содержатся двух- или трехуровневые деревья тем специфических областей математики, в том числе, теории вероятностей и математической статистики. Извлеченное из соответствующего паспорта дерево (Таблица 1) и представляет собой небольшое трехуровневое дерево, на первом уровне которого расположе-

Использование ресурсов Интернета для построения таксономии

ны 2 раздела: теория вероятностей, математическая статистика. В первом разделе 5 листов, во втором 6.

ТВиМС.01	Теория вероятностей	
	ТВиМС.01.01	Модели и характеристики случайных явлений
	ТВиМС.01.02	Распределения вероятностей и предельные теоремы
	ТВиМС.01.03	Комбинаторные и геометрические вероятностные задачи
	ТВиМС.01.04	Случайные процессы и поля
	ТВиМС.01.05	Оптимизационные и алгоритмические вероятностные задачи
ТВиМС.02	Математическая статистика	
	ТВиМС.02.01	Методы статистического анализа и вывода
	ТВиМС.02.02	Статистические параметры и их оценивание по выборке
	ТВиМС.02.03	Статистические критерии и проверка статистических гипотез
	ТВиМС.02.04	Временные ряды и случайные процессы
	ТВиМС.02.05	Машинное обучение
	ТВиМС.02.06	Многомерная статистика и анализ данных

Табл 1. Основа таксономии теории вероятностей и математической статистики

Мы использовали соответствующую категорию Википедии, то есть, “Теория вероятностей и математическая статистика”, в качестве единственного источника тем для достраивания. Заметим, что мы не обращались к другим разделам Википедии, поскольку название и тематика

Использование ресурсов Интернета для построения таксономии

данной категории полностью совпадает с названием и тематикой достраиваемой таксономии. По данным на конец 2011 года категория “Теория вероятностей и математическая статистика” насчитывала 48 подкатегорий и 640 статей.

Мы пользовались двумя типами данных из доступных данных Википедии:

1. Иерархической структурой дерева категорий Википедии
2. Коллекцией неструктурированных текстов статей Википедии.

Дерево категорий мы использовали для наращивания дерева исходной таксономии: к каждой таксономической теме на первом и втором уровне присоединим несколько подходящих по содержанию категорий Википедии и лежащие ниже подкатегории. Листьями таксономии считали названия статей, а извлеченные из текстов статей ключевые слова – уточнения листьев.

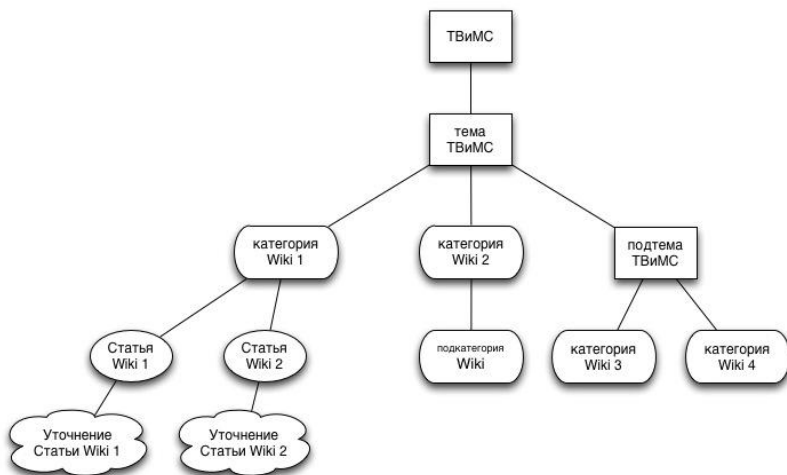


Рис. 1. Схема достраивания таксономии. Прямоугольники – темы исходной таксономии, скругленные прямоугольники – категории и подкатегории Википедии, овалы – статьи Википедии, облака – уточнения листьев, состоящие из ключевых слов и словосочетаний.

Таким образом, что каждую тему таксономии следует достроить двухуровневым деревом, состоящим из категории Википедии и принадлежащих к ней статей.

Структура категорий Википедии оказалось зашумленной: между некоторыми подкатегориями и категориями может не быть смысловой связи (например, категория Оптимизация находится в категории Машинное

обучения, которая, в свою очередь находится в категории Математическая статистика). Строго говоря, дерево категорий является решеткой, а не деревом, поскольку в некоторых случаях может содержать циклы. Одно из объяснений этому феномену согласно [15] заключается в том, что стандарт разметки Википедии допускает размещение подкатегории или статьи в неограниченном числе категорий, а авторы Википедии склонны, как правило, помещать статьи и подкатегории в как можно большее число категорий. Таким образом, для достраивания таксономии необходимо провести предварительную обработку данных из Википедии и очистить дерево категорий от иррелевантных статей и категорий.

Основные этапы достраивания таксономии:

1. Определить таксономическую тему для достраивания
2. Извлечь из Википедии фрагмент дерева категорий и статьи, соответствующие достраиваемой теме
3. Очистить дерево категорий от иррелевантных статей
4. Очистить дерево категорий от иррелевантных подкатегорий
5. Достроить следующий уровень таксономии под выбранной таксономической темой релевантными категориями
6. Достроить каждую категорию релевантными статьями – новыми листьями в таксономии
7. Извлечь ключевые слова и словосочетания из статей и использовать их в качестве уточнений листьев.

Опишем каждый из этапов на примере таксономии теории вероятностей и математической статистики (ТВиМС) и соответствующей категории Википедии.

1. Определение таксономической темы для достраивания: теория вероятностей и математическая статистика.
2. Извлечение из Википедии фрагмент дерева категорий и статьи, соответствующие достраиваемой теме. В Википедии существует категория с таким же названием, т.е. этот этап совершается путем загрузки поддерева категорий с корнем в категории “Теория вероятностей и математическая статистика”. В результате в нашем распоряжении оказались 640 статей Википедии, организованных в 48 категорий. Максимальная глубина загруженного дерева – 5, средняя глубина – 3. Некоторые категории содержат только подкатегории, но, в большинстве случаев, категории содержат и статьи, и подкатегории.
3. Очистка дерева категорий от иррелевантных статей. Среди вершин этого дерева оказались некоторые, очевидным образом не связанные с теорией вероятностей и математической статистикой, например, “Оптимизация программного обеспечения” или “Natural Language Toolkit”. Чтобы определить, релевантна ли статья категории, в которой она нахо-

дится, мы рассчитывали степень сходства названия категории с текстом статьи и, если найденная степень сходства ниже определенного порога, считали статью иррелевантной. Мы оценивали степень сходства с помощью метода аннотированного суффиксного дерева, описанного ниже. Согласно методу АСД, мера сходства, изменяющаяся от 0 до 1, выражает среднюю условную вероятность появления символа в строке после префикса строки. Чем ниже мера сходства, тем меньше вероятность, что название категории связано с содержанием статьи. Так, например, 7 из 12 статей в категории “Факторный анализ” оказались иррелевантными согласно сформулированному принципу, среди них “Линейная регрессия на корреляции” и “RANSAC”.

4. Очистка дерева категорий от иррелевантных категорий. Связь между данной категорией и ее подкатегорией определяется по аналогии со связью между категорией и статьей в ней. Вместо текста статьи мы использовали совокупность всех текстов в подкатегории и рассматривали их как один текст. Степень сходства названия категории с таким текстом должна превышать заданный порог, чтобы подкатегория была релевантной для своей родительской категории. К сожалению, такой принцип определения связи между подкатегориями и категориями не всегда эффективен. Так, подкатегория “Деревья принятия решений” оказалась иррелевантной категории “Машинное обучение”, поскольку ни одна из четырех статей в ней не содержит ни строки “Машинное обучение”, ни ее подстрок.

5. Достаивание следующего уровня таксономии под выбранной таксономической темой релевантными категориями. После очистки дерева категорий от иррелевантных статей и категорий мы достаивали оставшиеся категории под соответствующие таксономические темы существующей таксономии. Для определения соответствия категорий таксономическим темам снова был использован метод АСД. Мы вычисляли степень сходства между таксономическими темами и текстами статей в каждой категории, слитых в один текст. Поскольку топология дерева категорий не учитывалась, в некоторых случаях к одной и той же таксономической теме были достроены и категория, и ее подкатегории. Например, категории “Непрерывные распределения” и “Дискретные распределения”, принадлежащие к категории “Распределения вероятностей”, были достроены к таксономической теме “Распределения вероятностей и предельные теоремы” вместе со своей родительской категорией. В этом случае мы создали новый уровень в таксономическом дереве под темой “Распределения вероятностей” и поместили на него обе подкатегории, сохранив при этом структуру дерева категорий.

6. Достаивание категорий релевантными статьями. В достаиваемой таксономии листьями служат названия статей Википедии. Пусть к так-

сономической теме достроена некоторая категория Википедии. Все релевантные статьи, оставшиеся в этой категории после очистки, помещаются на последний уровень таксономического дерева и их названия становятся листьями таксономии.

7. Извлечение ключевых слов и словосочетаний из статей и использование их в качестве уточнений листьев. Каждый лист таксономии следует снабдить множеством слов и словосочетаний, описывающих его содержание так, как это сделано в таксономии ACM-CCS. Мы использовали в качестве уточнений ключевые слова и словосочетания, извлеченные из текстов статей. Для их извлечения мы не пользовались сложными алгоритмами и считали, что ключевое слово – это существительное, частота которого в тексте статьи достаточно велика, а ключевое словосочетание – это частотная пара слов, удовлетворяющая синтаксическим шаблонам “прилагательное + существительное” или “существительное + существительное”. Таким образом, лист “Корреляция” получил такое уточнение: “коэффициент корреляции”, “случайная величина”, “ранг” и т.д., а лист “Метод максимального правдоподобия” – “функция правдоподобия”, “параметр”, “выборка”.

Метод АСД

Суффиксное дерево – это структура данных, используемая для хранения и поиска символьных строк и их фрагментов [16]. В некотором смысле, суффиксное дерево можно считать альтернативой векторной модели представления текстов (VSM) [17]. Если текст представлен суффиксными деревом, то его элементами оказываются не отдельные слова (как это происходит в наипростейшей модели “мешок слов”), а строки неограниченной длины, которые могут быть как и фрагментом слова, так и целым словом, словосочетанием и даже предложением.

Аннотированное суффиксное дерево (АСД) – это суффиксное дерево, узлы (а не ребра!) которого аннотированы частотами фрагментов строк. Алгоритм построения АСД и использования его в задаче фильтрации спама описан в [18], а в [6, 19] представлены другие приложения метода АСД.

В наших расчетах любая статья Википедии считалась множеством строк, состоящих из трех слов, а название статьи включалось в это множество без изменений. Для того, чтобы оценить сходство отдельной строки со множеством строк, мы, во-первых, строили АСД для множества строк, во-вторых, находили все совпадающие фрагменты данной строки в построенном АСД. После этого мы вычисляли оценку каждого совпавшего фрагмента: среднюю частоту символа в фрагменте, нормированную длиной всего фрагмента. Общая оценка строки вычислялась

как усредненная оценка всех совпавших фрагментов. Таким образом, окончательная оценка лежит между 0 и 1 и может считаться условной вероятностью. По сравнению с мерами сходства, описанными в [18] данная оценка имеет естественную вероятностную интерпретацию и независима от длины оцениваемой строки.

Результаты

В результате уточнения таксономии, представленной в таблице 1, была получена таксономия, глубина которой изменяется от 4 до 7. Фрагмент достроенной таксономии представлен на Рисунке 2. На этапе очистки из дерева категорий Википедии было удалено 100 иррелевантных статей и 2 иррелевантные категории. Некоторые таксономические темы, например, “Методы статистического анализа и вывода” не были достроены.

Основной недостаток построенного таксономического дерева – это положение темы “Деревья принятия решений”. Согласно представленному методу, это тема должна быть потомком темы “Многомерная статистика и анализ данных”, то есть, иметь общего родителя с темой “Машинное обучение”. Объяснение этому приведено выше: мера сходства строки “Машинное обучение” со статьями в категории “Деревья принятия решений” удивительно мала.

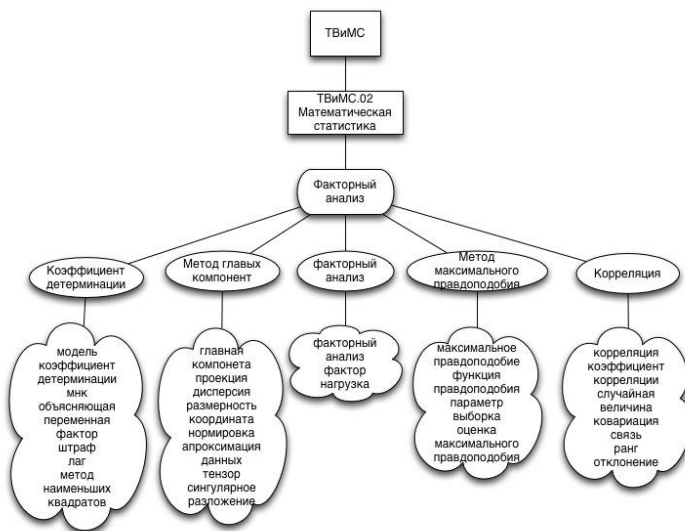


Рис. 2. Фрагмент достроенной таксономии: “Факторный анализ”

В задаче достраивания таксономии метод АСД использовался трижды:

1. Для очистки дерева категорий Википедии от иррелевантных статей;
2. Для очистки дерева категорий Википедии от иррелевантных категорий;
3. Для определения связей между таксономическими темами и категориями Википедии.

В двух первых случаях требовалось задать значение порога отсеечения иррелевантных статей и категорий. Эксперименты показали, что разумно установить порог на уровне 0.2 как $1/3$ от максимального получаемого значения.

Заключение

Метод автоматического достраивания таксономии – часть двухшагового подхода к построению таксономий. На первом шаге эксперт задает основу таксономии. На втором шаге таксономия автоматически достраивается тема за темой до необходимого уровня детализации. Этот подход позволяет защитить таксономию от зашумления и избежать появления иррелевантных или слишком узких тем. Википедия оказалась хорошим источником тем для достраивания, поскольку содержит и структурированные (дерево категорий), и неструктурированные (тексты статей) данные.

Использование метода АСД в задаче достраивания таксономий обладает своими достоинствами и недостатками. К его достоинствам относится независимость от языка и его грамматики. Главный недостаток метода заключается в том, что метод основан на посимвольных и пословных совпадениях и не позволяет использовать синонимы. Развитие метода будет направлено на использование синонимических отношений. Кроме того, мы будем рассматривать и другие источники таксономических тем: ГОСТы, научные статьи или учебные программы, выдачу поисковых систем по запросам – таксономическим темам.

Список источников

1. ACM Computing Classification System (ACM CCS), (1998), available at: <http://www.acm.org/about/class/ccs98-html>.
2. Chernyak E. L., Chugunova O. N., Mirkin B.G. (2012), Annotated suffix tree method for measuring degree of string to text belongingness [Metod anotirovannogo suffiksnogo dereva dlja otsenki stepeni vhozhdenija strok v tekstovie dokument], *Biznes-Informatika* [Business Informatics], no.3, pp. 31-41.
3. Chernyak E.L., Chugunova O.N., Askarova J.A., Nascimento S., Mirkin B. G. Abstracting concepts from text documents by using an ontology. Proceedings of the 1st International Workshop on Concept Discovery in Unstructured Data. Moscow, 2011, pp. 21-31.
4. Grau B.C., Parsia B., Sirin E. Working with Multiple Ontologies on the Semantic Web. In Proceedings of the 3d International Semantic Web Conference, Hiroshima, Japan 2004, pp. 620-634.
5. Grineva M., Grinev M., Lizorkin D. (2009) Text documents analysis for thematically grouped key terms extraction, in *Trudy Instituta sistemnogo programirovaniya RAN, Institute for System Programming*, pp. 155-156 (in Russian).
6. Gusfield D. (1997), *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press.
7. Higher Attestation Commission of RF Reference, (2009), available at: http://vak.ed.gov.ru/ru/help_desk/.
8. Kittur A., Chi E.H., Suh B. What's in Wikipedia? Mapping topics and conflict using socially annotated category structure. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, USA, 2009, pp. 1509-1512.
9. Liu X., Song, Y., Liu S., Wang H. Automatic Taxonomy Construction from Keywords. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, USA, New York, 2012, pp. 1433-1441.
10. Loukachevitch N.V. (2011), *Tezaurusy v zadachah informatsionnogo poiska* [Thesauri in information retrieval tasks], MSU, Moscow.

11. Pamparathi R., Mirkin B., Levene M., (2006), A suffix tree approach to anti-spam email filtering, Machine Learning 2006, Vol. 65(1), pp. 309-338.
12. Ponzetto S.P., Strube M. Deriving a Large Scale Taxonomy from Wikipedia. In Proceedings of AAAI Conference on Artificial Intelligence, Vancouver, Canada, 2007, pp. 78-85.
13. Robinson P.N., Bauer, S. (2011), Introduction to Bio-Ontologies, CRC, USA.
14. Sadikov E., Madhavan J., Wang L., Halevy A.Y., Clustering query refinements by user intent. In Proceedings of the 19th International Conference on World Wide Web, New York, USA, 2008 pp. 841-850.
15. Taxonomy of Abstracting Journal “Mathematics” (1999), VINITI. Available at: <http://www.viniti.ru/russian/math/files/271.htm>.
16. Van Hage W.R., Katrenko S., Schreiber G., A Method to Combine Linguistic Ontology-Mapping Techniques. In Proceedings of 4th International Semantic Web Conference, 2005, Galway, Ireland, pp. 34-39.
17. White R.W., Bennett P.N., Dumais S.T. Predicting short-term interests using activity-based search contexts. In Proceedings of 19th ACM conference on Information and Knowledge Management, Toronto, Canada, 2010, pp. 1009-1018.
18. Wu F., Weld. D. Automatically refining Wikipedia Infobox Ontology. In Proceedings of the 17th International World Wide Web Conference, Beijing, China, 2008, pp. 635-645.
19. Zamir O, Etzioni. O. Web document clustering: A feasibility demonstration. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, USA, 1998, pp. 46-54.
20. Zirn C., Nastase V., Strube M., Distinguishing between Instances and Classes in the Wikipedia Taxonomy. In Proceedings of 5th European Semantic Web Conference, Tenerife, Spain, 2008, pp. 376-387.

Аннотированные суффиксные деревья: особенности реализации

Михаил Дубов¹, Екатерина Черняк²

¹Отделение программной инженерии НИУ ВШЭ, Москва, Россия.
msdubov@gmail.com

²Отделение прикладной математики и информатики НИУ ВШЭ, Москва,
Россия. ek.chernyak@gmail.com

Аннотация. В статье описываются особенности эффективной программной реализации разработанной с участием одного из авторов модификации суффиксных деревьев, предполагающей аннотацию узлов дерева частотами встречаемости соответствующих им подстрок в исходной коллекции текстов. Данная структура данных имеет ряд практически важных приложений, таких как оценка степени вхождения последовательности символов в текст или анализ связей между ключевыми словосочетаниями. Предложенные в данной работе модификации известных алгоритмов быстрого построения обычных суффиксных деревьев, а также описываемые в ней приемы хранения аннотированных суффиксных деревьев в памяти делают возможным их практическое применение для анализа больших коллекций текстов, что подтверждается приводимыми в тексте статьи результатами сравнительного исследования производительности различных реализаций метода АСД на реальных данных.

Ключевые слова: анализ текстов, аннотированные суффиксные деревья, обобщенные суффиксные деревья, алгоритм Укконена, комбинированные алгоритмы, суффиксные массивы.

Введение

Среди многообразия методик анализа текстов метод аннотированного суффиксного дерева (далее – АСД) выделяется тем, что принадлежит к немногочисленному классу моделей, представляющих текст как последовательность символов, но не как последовательность слов. Это позволяет достичь высокой степени независимости от языка анализируемых текстов, а также исключает необходимость выполнять их предобработку, которая зачастую оказывается довольно трудоемкой.

В основе метода АСД лежит построение суффиксного дерева специального вида, позволяющего оценивать степень вхождения ключевых словосочетаний в исходный корпус текстов. Получаемая оценка может использоваться в таких приложениях, как фильтрация спама [2], анализ связей между ключевыми словосочетаниями или анализ структуры корпуса текстов путем их иерархической группировки [3].

До настоящего времени в публикациях практически не уделялось внимания эффективной программной реализации рассматриваемой структуры данных. Между тем, производительная реализация метода АСД чрезвычайно важна для анализа крупных коллекций текстов. Основным способом достижения высокой производительности метода АСД и посвящена настоящая работа.

В данной научной работе использованы результаты, полученные в ходе выполнения проекта «Методы визуализации текстовой информации с помощью построения суффиксных деревьев, мультифасетных классификаций и иерархических онтологий: алгоритмическое и программное обеспечение», выполненного в рамках Программы «Научный фонд НИУ ВШЭ» в 2013 году, грант № 13-05-0047.

Наивная реализация метода АСД

Приведем краткое изложение метода АСД. Более подробное его описание читатель может найти в [3].

Основой для анализа набора текстов по методу АСД являются собственно построение обобщенного АСД для поступающей на вход коллекции строк, а также наложение на полученное АСД ключевых словосочетаний.

В существующих публикациях [2][3] предполагается организация АСД как корневого дерева, в котором каждый узел (кроме корня) помечен одним из символов алфавита, на котором определена входная коллекция строк. АСД при этом кодирует все суффиксы всех строк в коллекции: им соответствуют пути от корня до листьев дерева. Каждый узел v в дереве помечен целым числом $f(v)$, обозначающим количество

вхождения соответствующего фрагмента строки (от корня до данной вершины) в исходный набор текстов. Пример такого дерева для коллекции, состоящей из одной строки “ХАВХАС”, приведен на рис. 1.

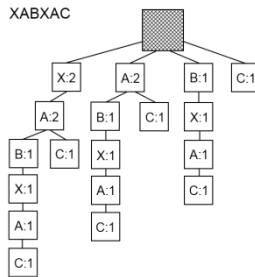


Рис. 1. Наивная реализация АСД

Для построения АСД предлагается следующий алгоритм.

Алгоритм **NaiveConstruction**(C)

Вход. Коллекция строк $C = \{S_1 \dots S_m\}$.

Выход. Обобщенное АСД для C .

1. **for** $i \leftarrow 1$ **to** m
2. **for** $j \leftarrow 1$ **to** $n_i = |S_i|$
3. **do** $k \leftarrow$ длина совпадения суффикса $S_i[j:]$ с АСД
4. **for** узел u из $S_i[j: (j + k - 1)]$
5. **do** присвоить $f(u) \leftarrow f(u) + 1$
6. **for** $l \leftarrow j + k$ **to** n_i
7. **do** вставить узел v
8. присвоить $f(v) \leftarrow 1$

Анализ трудоемкости данного алгоритма достаточно прост. Для каждой строки мы посимвольно просматриваем все ее суффиксы, затрачивая, таким образом, на i -ю строку длины n_i количество операций, пропорциональное $(1 + 2 + \dots + n_i) = \theta(n_i^2)$. Общее время работы алгоритма для коллекции из m строк, таким образом, может быть оценено как $\theta(n_1^2 + \dots + n_m^2)$, или, если использовать несколько более грубую оценку, $\mathbf{O}(mn_{max}^2)$. Отметим, что обустроенное описанным выше способом АСД невозможно построить с использованием меньшего числа операций, так как само оно занимает в памяти место, квадратично зависящее от длины закодированных в нем строк.

На втором этапе метода АСД оценивается степень вхождения ключевых словосочетаний в исходную коллекцию слов с помощью их

«наложения» на построенное дерево. Результатом этой оценки является число в интервале $[0; 1]$, характеризующее степень вхождения строки в коллекцию текстов, на основе которой было построено АСД.

Для оценки вхождения строки S в коллекцию текстов (или, что то же самое, в дерево T) необходимо выполнить следующие шаги [3]. Назовем условной **вероятностью узла v** число $\hat{p}(v) = f(v)/f(\text{parent}(v))$, где $f(v)$ – частота узла v , $f(\text{parent}(v))$ – частота узла-родителя v (частотой корня будем считать сумму частот узлов на первом уровне дерева). Пусть для некоторой строки s максимальное ее совпадение с символами в АСД по любому пути от корня к листьям имеет длину k . В таком случае оценка совпадения $v_1 \dots v_k$ вычисляется как сумма условных вероятностей входящих в него узлов, нормированная по длине совпадения:

$$\text{score}(s) = \frac{\sum_{i=1}^k \hat{p}(v_i)}{k}$$

Наконец, степень вхождения строки S в АСД вычисляется как сумма оценок совпадений всех ее суффиксов, нормированных по длине строки:

$$\text{SCORE}(S) = \frac{\sum_{i=1}^{|S|} \text{score}(S[i:|S|])}{|S|}$$

Завершая данный раздел, упомянем важное **свойство АСД**: частота любого его внутреннего узла равна сумме частот соответствующих дочерних узлов. Это свойство будет использовано в алгоритме быстрого построения обобщенного АСД, который мы опишем далее.

Построение АСД за линейное время

Отметим, что описанная выше структура данных, называемая в [2] и [3] аннотированным суффиксным деревом, строго говоря, суффиксным деревом не является. Действительно, в ней нарушено одно из основных свойств суффиксных деревьев [1]: в большом количестве присутствуют узлы с единственным потомком, в то время как в суффиксном дереве у каждой внутренней вершины, отличной от корня, должно быть не менее двух детей. Выполнить это условие можно, только схлопнув каждую цепь из узлов с единственным потомком в одну вершину и пометив входящее в нее ребро конкатенацией символов, которыми были помечены узлы в этой цепи. Частота самой вершины остается неизменной, так как у всех вершин в цепи она была одинаковой. Преобразовав таким образом исходное дерево, получим структуру данных, изображенную на рис. 2а (дерево вновь построено для строки “ХАВХАС”):

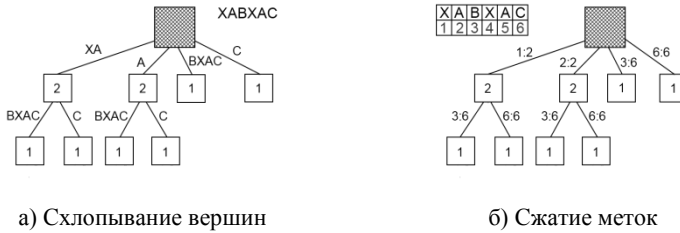


Рис. 2. Оптимизация представления АСД

Реализованное таким образом дерево все еще требует $O(mn_{max}^2)$ памяти из-за необходимости хранить все метки ребер в явном виде. Существует, однако, простой прием сжатия дуговых меток, позволяющий снизить количество используемой деревом памяти до линейного относительно длин строк в коллекции. Он заключается в том, чтобы хранить в каждом ребре только индексы начала и конца соответствующей подстроки, а не всю подстроку в явном виде [1]. Окончательный вид АСД после всех описанных оптимизаций представлен на рис. 2б.

Отметим, что, несмотря на уменьшение количества вершин в дереве, вычисление степени вхождения в него строк все еще остается возможным. Все, что требуется – это немного видоизменить формулу оценки совпадения строки с АСД:

$$score(s) = \frac{(\sum_{i=1}^k \hat{p}(v_i)) + l - k}{l},$$

где k , как и ранее – число узлов в совпадении, а l – фактическая длина совпадения в символах.

Возможность столь значительного снижения количества используемой для хранения АСД памяти позволяет также рассчитывать на существование асимптотически менее трудоемких алгоритмов построения АСД, чем тот, который был описан выше. Действительно, существует целый ряд линейных по времени алгоритмов построения обычных (неаннотированных) суффиксных деревьев. В литературе в качестве основных принято выделять алгоритмы П. Вайнера (1973), Э. МакКрейга (1976) и Э. Укконена (1995) [1]. Определенную проблему, однако, представляет построение с помощью этих алгоритмов аннотированных суффиксных деревьев, где узлы помечены частотами соответствующих им подстрок. Дело в том, что временная эффективность всех этих алгоритмов достигается путем игнорирования на каждом шаге определенных путей в дереве, что делает невозможным своевременное обновление меток узлов, как это происходит в наивном алгоритме. Оказывается, однако, что аннотировать дерево можно и после его построения, если предварительно выполнить несложную предобработку коллекции строк:

Алгоритм **LinearConstruction**(C)

Вход. Коллекция строк $C = \{S_1 \dots S_m\}$.

Выход. Обобщенное АСД для C .

1. Построить $C' = \{S_1\$1 \dots S_m\$m\}$, где $\$i$ - уникальные символы.
2. Построить обобщенное суффиксное дерево T для коллекции C' , используя алгоритм с линейной сложностью.
3. **for** l **in** $leaves(T)$
4. **do** присвоить $f(l) \leftarrow 1$
5. Выполнить **обход** дерева T **снизу вверх**; в каждом внутреннем узле v присвоить $f(v) \leftarrow \sum_{u \in T: parent(u)=v} f(u)$.

Основанием для аннотирования в шаге 3 всех листьев числом 1 является тот факт, что на первом шаге мы приписываем к каждой строке в коллекции уникальный символ – а значит, каждая подстрока, соответствующая одному из путей от корня до листа, встречается в исходном наборе строк только один раз. Метод присвоения меток внутренним вершинам в шаге 5 напрямую следует из описанного выше свойства АСД. Обход дерева при этом требует времени, пропорционального числу вершин. Таким образом, все шаги алгоритма выполняются за линейное время, и общая оценка его трудоемкости составляет $\theta(n_1 + \dots + n_m)$ или $O(mn_{max})$.

Сравнительный анализ алгоритмов построения АСД

Обратимся к экспериментальному исследованию производительности описанных выше алгоритмов. В качестве основы для экспериментов выступала реализация метода АСД на языке Python 2.7 (без использования дополнительных библиотек вроде NumPy). Основой для линейного алгоритма стал алгоритм Укконена, как наиболее простой в реализации.

Важно отметить, что с использованием описанного выше оптимизированного способа хранения АСД в памяти оценка трудоемкости наивного алгоритма становится квадратичной только в худшем случае. Действительно, теперь для каждой части суффикса, которой нет в дереве, алгоритму требуется добавить в дерево не целый ряд вершин, а только один лист, дуга к которому помечена отсутствующими в дереве символами. В ряде случаев это делает наивный алгоритм сопоставимым с линейным, а иногда и превосходящим его по трудоемкости (как в случае построения АСД для коллекции случайно генерируемых строк, в структуре которых отсутствуют какие-либо закономерности, рис. 3а). Это объясняется тем, что алгоритм Укконена организован сложнее наивного и требует для своей работы ряда накладных расходов (например, на организацию в дереве так называемых суффиксных связей).

Худшим же случаем для наивного алгоритма является такая коллекция строк, при обработке которой он вынужден на каждой итерации опускаться на всю глубину уже построенной части дерева. Примером такой коллекции может быть набор строк, различающихся только одним-двумя последними символами. На таких входных данных трудоемкость наивного алгоритма действительно деградирует до квадратичной, а выигрыш от использования линейного алгоритма построения АСД становится значительно более очевидным (рис. 3б).

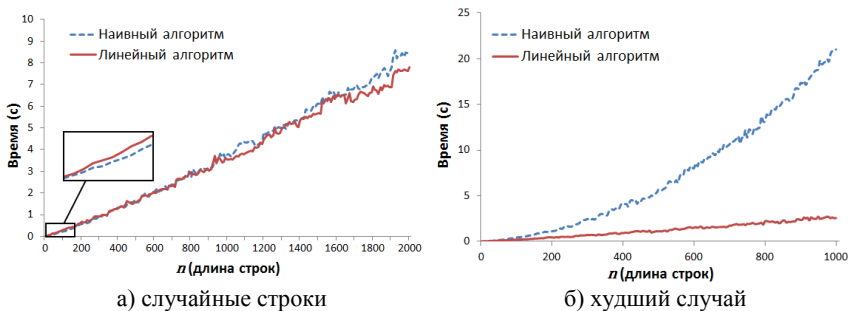


Рис. 3. Экспериментальная трудоемкость построения обобщенного АСД для коллекции из 100 строк длины n

Заметим, что и в случае случайных строк при росте числа закодированных в дереве символов наивный алгоритм все больше начинает уступать линейному в скорости: это вызвано тем, что плотность дерева на верхних уровнях растет и наивному алгоритму в среднем приходится глубже опускаться по дереву при добавлении в него новых строк.

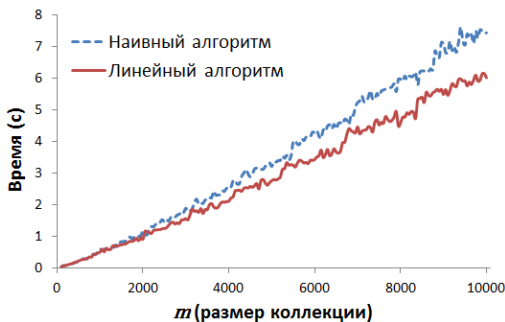


Рис. 4. Экспериментальная трудоемкость построения обобщенного АСД для набора из m строк, полученного из коллекции текстов Reuters [4]

В текстах на естественных языках слова, начинающиеся с одинаковых приставок или имеющие одинаковые основы, встречаются достаточно часто, поэтому мы можем ожидать в среднем большей глубины дерева, чем для случайных строк, и, следовательно, лучшей производительности линейного алгоритма, чем наивного. Действительно, при построении АСД для англоязычной коллекции текстов Reuters-21578 [4] линейный алгоритм в среднем оказался на 25-30% производительнее, чем наивный (рис. 4). Отметим, что при построении АСД тексты из коллекции разбивались на строки по 3 слова (такая предобработка была предложена в [3] для получения более адекватных результатов наложения на дерево ключевых словосочетаний, длина которых, как правило, также не превышает трех-четырех слов).

Комбинированные алгоритмы

Тот факт, что наивный алгоритм оказывается немного производительнее линейного при малой глубине дерева (которая является следствием малого числа закодированных в нем строк), дает основания для попыток построения комбинированных алгоритмов конструирования АСД, использующих преимущества как наивного, так и линейного алгоритмов. Наиболее очевидными представляются следующие стратегии:

- Для каждой следующей строки найти длину ее совпадения при наложении на дерево. Если это число ниже некоторого порога t , то использовать наивный алгоритм, иначе - алгоритм Укконена (точное значение t варьируется в пределах от 1 до 10 и определяется экспериментально);
- Первые k строк из поступающей на вход коллекции добавляются в дерево наивным алгоритмом, остальные - алгоритмом Укконена (точное значение k определяется экспериментально).

Первый алгоритм в ходе экспериментов не дал никакого выигрыша: накладные расходы на проверку вхождения строки в АСД превысили выгоду от комбинирования двух алгоритмов. Второй же алгоритм позволяет достичь небольшого преимущества в скорости: оно составило до 5% на использованной для тестирования коллекции Reuters и до 10-15% на случайных строках. Столь небольшой выигрыш отчасти объясняется тем, что использование наивного алгоритма в комбинации с линейным требует усложнения первого: теперь в наивном алгоритме также необходимо добавлять в дерево суффиксные связи, чтобы обеспечить нормальное выполнение алгоритма Укконена на втором этапе.

Использование суффиксных массивов

Чрезвычайно привлекательным при работе с суффиксными деревьями является переход к использованию суффиксных массивов [1]. Эта структура данных, также как и суффиксное дерево, кодирует все суффиксы строки, но устроена при этом таким образом, что позволяет сократить использование памяти в 2-3 раза.

Одним из последствий перехода к суффиксным массивам, однако, является отказ от понятия «узел», и, как следствие, от родительско-дочернего отношения между узлами. Так как лежащее в основе метода АСД вычисление степени вхождения строк в дерево активно использует понятие «условной вероятности узла», которая является отношением частоты узла к частоте узла-родителя, использование суффиксных массивов при реализации метода АСД представляется нам невозможным.

Выводы

В ходе нашего исследования была разработана программная реализация метода АСД, значительно превосходящая по эффективности реализации, предложенные в более ранних работах. Результаты экспериментального исследования ее производительности при этом наглядно продемонстрировали, что эффективность используемых нами алгоритмов сильно зависит от вида входных данных и, вероятно, до некоторой степени варьируется между коллекциями текстов на разных языках. Изучение этих зависимостей и разработка алгоритмов построения АСД, чувствительных к особенностям входных данных, составляет предмет дальнейших исследований.

Список источников

1. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, 1997.
2. Pampapathi, R., Mirkin, B., Levene M. A suffix tree approach to anti-spam email filtering. Machine Learning, v.65 n.1, October 2006. p. 309-338.
3. Миркин Б. Г., Черняк Е. Л., Чугунова О. Н. Метод аннотированного суффиксного дерева для оценки степени вхождения строк в текстовые документы. Бизнес-информатика, №3(21), 2012. с. 31-41.
4. Reuters-21578 Text Categorization Test Collection[Электронный ресурс] URL: <http://www.daviddlewis.com/resources/testcollections/reuters21578/> (дата обращения: 10.02.2013)

Серелекс: поиск и визуализация семантически связанных слов

Панченко А.И.^{1,2}, Романов П.В.², Романов А.В.¹,
Филиппович А.Ю.², Морозова О.И.¹, Филиппович Ю.Н.²

¹ Université catholique de Louvain, Лувен, Бельгия

² МГТУ им. Н. Э. Баумана, Москва, Россия

Аннотация. В статье представлена система Серелекс, которая выдает в ответ на поисковый запрос список семантически связанных с ним слов. В настоящее время система работает на английском языке, ведутся также разработки для французского и русского языков. Слова ранжируются в соответствии с оригинальной метрикой семантической близости, обученной на корпусе естественно-языковых текстов. Точность работы системы сравнима с аналогами, основанными на WordNet и словарях. При этом система использует только информацию, извлеченную непосредственно из текстов. Исследование показывает, что пользователи полностью удовлетворены результатами поиска семантически связанных слов в 70% случаев.

Ключевые слова: метрика семантической близости; визуализация семантических отношений.

1 Введение

В данной статье представлена система Серелекс, которая на английский запрос выдает список связанных с ним слов в порядке их

семантической близости ¹. Программа помогает изучить значение иностранных слов и интерактивно исследовать связанные лексические единицы и их семантические поля. В отличие от систем, основанных на словарях и тезаурусах, таких как *Thesaurus.com* или *VisualSynonyms.com*, Серелекс использует информацию, извлечённую из корпуса естественно-языковых текстов. В отличие от аналогичных систем, извлекающих информацию из текстов, таких как *BabelNet* ², *ConceptNet* ³ и *UBY* ⁴, Серелекс не использует дополнительно информацию из таких семантических ресурсов, как *WordNet*.

В основе разработанной системы лежит оригинальная метрика семантической близости, использующая лексико-синтаксические шаблоны [2]. Согласно экспериментам, точность использованного подхода сопоставима с существующими аналогами для английского языка. Кроме того, представленная система характеризуется большим лексическим покрытием, чем аналоги, основанные на словарях, предлагает три альтернативных способа визуализации результатов запроса (в виде списка, графа и набора изображений) и имеет открытый исходный код.

2 Система

Серелекс находится в открытом доступе в интернете ⁵. Система состоит из экстрактора, сервера и пользовательского интерфейса (см. Рис. 1). Задача экстрактора заключается в извлечении семантических отношений между словами из корпуса естественно-языковых текстов. Извлечённые отношения сохраняются в базе данных. Сервер обеспечивает быстрый доступ к извлечённым отношениям через HTTP. Пользователь взаимодействует с системой через веб-интерфейс или API. Исходный код системы, данные и скрипты оценки качества работы доступны на условиях лицензии LGPLv3 ⁶.

2.1 Экстрактор

Подсистема извлечения семантических отношений основана на метрике семантической близости *PatternSim* и формуле ранжирования *Efreq-Rnum-Cfreq-Pnum* [2]. Метрика семантической близости использует лексико-синтаксические шаблоны, подобно [3]. Данные

¹ Данная статья является расширенной версией [1].

² <http://lcl.uniroma1.it/bnexplorer/>

³ <http://conceptnet5.media.mit.edu/>

⁴ <https://uby.ukp.informatik.tu-darmstadt.de/webui/tryuby/>

⁵ <http://serelex.cental.be> или <http://serelex.it-claim.ru>

⁶ <http://serelex.cental.be/page/about>

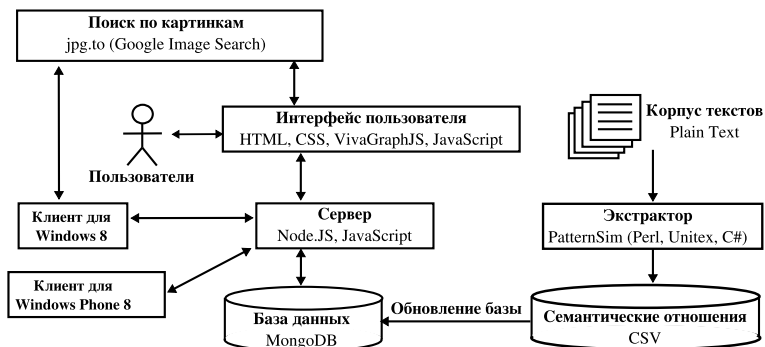


Рис. 1. Архитектура системы.

шаблоны извлекают из корпуса текстов множество конкордансов, таких как:

- such diverse {[occupations]} as {[doctors]}, {[engineers]} and {[scientists]}
- such {non-alcoholic [sodas]} as {[root beer]} and {[cream soda]}
- {traditional[food]}, such as {[sandwich]}, {[burger]}, and {[fries]}
- {[mango]}, {[pineapple]}, {[jackfruit]} and other{[fruits]}
- {primitive [snake]}, such as {[boa]} and {[python]}
- {[France]}, {[Belgium]} and other {European [countries]}

Существительные в конкордансах (обозначены квадратными скобками) были лемматизированы с помощью словаря DELA ⁷. Семантическое сходство двух таких лемм пропорционально количеству конкордансов, в которых они совместно встретились. Однако окончательное значение семантической близости вычисляется с учетом и других факторов, таких как частота слов в корпусе и количество извлеченных отношений для каждого из слов [2]. Извлечение отношений было произведено из коллекции текстовых документов, состоящей из заголовков статей Википедии и корпуса ukWaC [4] (см. Таблицу 1). Обработка данного корпуса заняла около 72 часов на стандартном компьютере (Intel i5, 4Гб ОЗУ, HDD 5400 об/мин). В результате извлечения было выявлено 11,251,240 нетипизированных семантических отношений, таких как $\langle Canon, Nikon, 0.62 \rangle$, между 419,751 леммами.

⁷<http://infolingua.univ-mlv.fr/>, доступен на условиях лицензии LGPLLLR.

Название	# Документов	# Словоформ	# Лемм	Размер
Википедия	2,694,815	$2,026 \cdot 10^9$	3,368,147	5.88 Гб
ukWaC	2,694,643	$0.889 \cdot 10^9$	5,469,313	11.76 Гб
Википедия + ukWaC	5,387,431	$2.915 \cdot 10^9$	7,585,989	17.64 Гб

Таблица 1. Корпуса текстов, использованные системой.

2.2 Сервер

Сервер возвращает множество связанных слов для каждого запроса, отсортированных согласно их семантической близости, сохранённой в базе данных. Запросы перед обработкой лемматизируются при помощи словаря DELA. Для слов, для которых не нашлось ни одного результата, выполняется приблизительный поиск с помощью расстояния Левенштейна. Система позволяет импортировать семантические отношения, которые были извлечены альтернативными экстракторами, в формате CSV.

2.3 Пользовательский интерфейс

Для работы с системой можно использовать веб-интерфейс, приложение для Windows 8, приложение для Windows Phone 8 или RESTful веб-сервис. Веб-интерфейс состоит из трёх основных элементов: строки поиска, списка результатов и графа результатов (см. Рис. 2). Пользователь взаимодействует с системой, формулируя поисковый запрос, который может быть выражен словом, таким как “mathematics”, или словосочетанием, таким как “computational linguistics”.

Кроме графового интерфейса пользователя, реализован интерфейс, основанный на изображениях. При этом всю рабочую область занимает графическое представление слов, связанных с результатами поиска. Выбор изображений осуществляется на основе веб-сервиса jpg.to⁸. Кликнув на изображение, пользователь может перейти к словам, семантически связанным с словом, представленном на изображении.

В дополнение к веб-интерфейсу были разработаны приложения для Windows 8⁹ и Windows Phone¹⁰. Данные клиенты используют веб-сервис Серелекса для получения результатов запросов и

⁸<http://jpg.to/about.php>. Данный сервис использует Google Image Search: <http://images.google.ru/>.

⁹<http://apps.microsoft.com/windows/app/1s8e/48dc239a-e116-4234-87fd-ac90f030d72c>

¹⁰<http://www.windowsphone.com/s?appid=dbc7d458-a3da-42bf-8da1-de49915e0318>

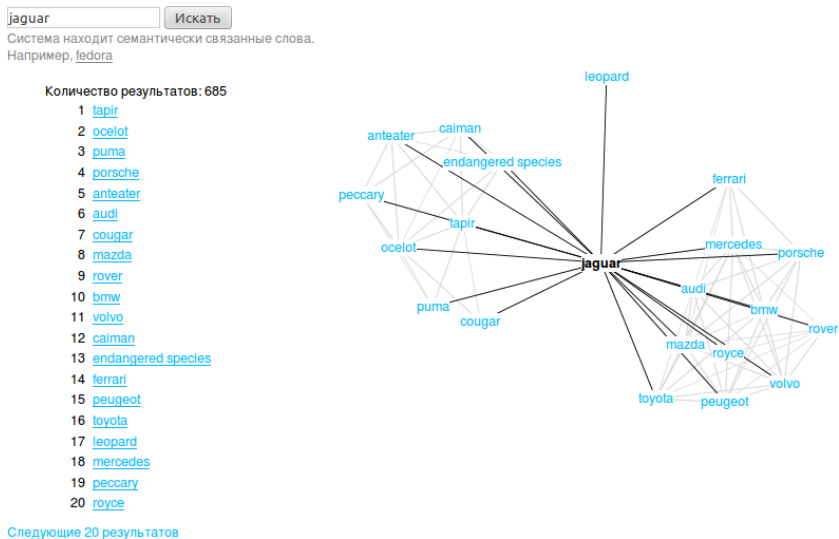


Рис. 2. Визуализация результатов поиска.

сервис [jrg.to](#) для получения изображений (см. Рис. 1). Приложения выполнены с учетом рекомендаций по построению пользовательского интерфейса приложений для Windows и Windows Phone, а их исходный код является открытым ¹¹. В рамках создания приложений для Windows была создана переносимая библиотека классов (Portable Class Library), которая может быть полезна для доступа к веб-сервису Серелекса из сторонних приложений. Отличительной особенностью клиента системы для Windows Phone является то, что он позволяет сразу же выполнить поиск в Google по результатам запроса (см. Рис. 5).

3 Результаты

Мы оценили качество работы системы, проведя четыре эксперимента, подробное описание которых приведено в [2].

3.1 Корреляция с суждениями о семантической близости

Для оценки корреляции с суждениями о семантической близости использовались три проверочных набора данных, широко распространенных в англоязычной литературе по лексической семантике:

¹¹<https://github.com/jgc128/Serelex4Win>

pizza Искать
 Система находит семантически связанные слова.
 Например, mango

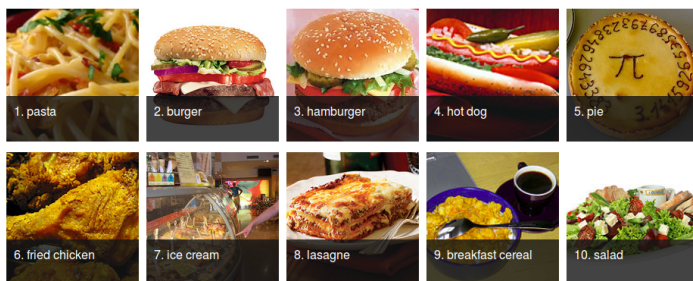


Рис. 3. Интерфейс, основанный на изображениях.

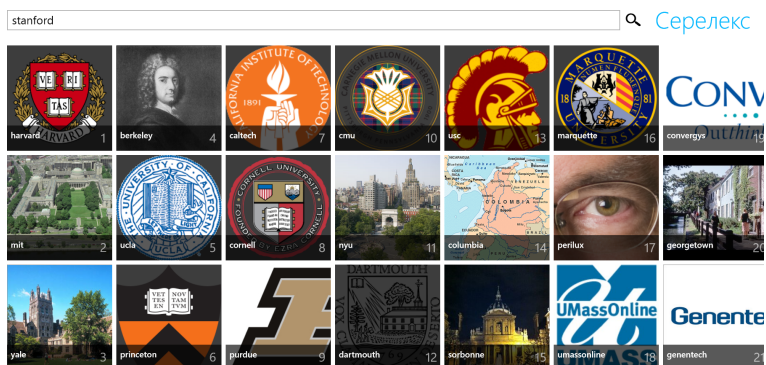


Рис. 4. Клиент системы для платформы Windows 8.

MC [5], *RG* [6] и *WordSim* [7]. Данные коллекции содержат множество пар слов, для каждой из которых вручную задана мера их семантической близости, например:

- automobile; car; 3.92
- brother; monk; 2.84
- glass; magician; 0.11

Согласно результатам проведенных экспериментов, корреляция Спирмена между значениями семантической близости, предоставляемыми системой, и суждениями субъектов достигает 0.665, 0.739 и 0.520 для *MC*, *RG* и *WordSim* соответственно. Данные характеристики Серелекса сравнимы с показателями существующих метрик семантической близости (см. [2]), основанных на WordNet (*WuPalmer* [8], *LeacockChodorow* [9], *Resnik* [10]), словарях

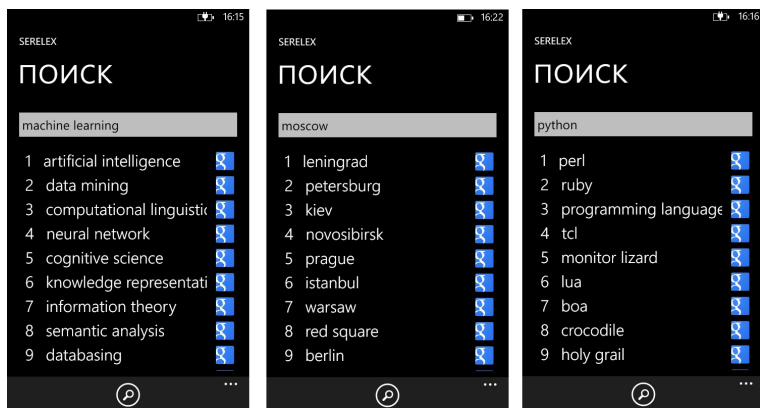


Рис. 5. Клиент системы для Windows Phone 8.

(*ExtendedLesk* [11], *GlossVectors* [12], *WiktionaryOverlap* [13]) и корпусах текстов (*ContextWindow* [14], *SyntacticContext* [14], *LSA* [15]).

3.2 Ранжирование семантических отношений

В данном тесте нужно отсортировать некоторое множество слов по семантической близости с заданным словом. Например, дано 50 слов, 25 из которых семантически связаны со словом “alligator”, в то время как 25 других с ним не связано. Задача заключается в ранжировании слов таким образом, чтобы семантически связанные пары имели более высокий ранг, например:

- 1; alligator; animal (related)
- ...
- 25; alligator; lizard (related)
- 26; alligator; twin (random)
- ...
- 50; alligator; electronic (random)

Задача ранжирования основана на наборе семантических отношений BLESS [16] и SN [17]. В отличие от трех других тестов, данная задача позволяет оценить не только относительную точность, но и относительную полноту системы.

Точность Серелекса на данной задаче сопоставима с 9 указанными выше альтернативными метриками, однако полнота серьезно ниже в связи с разреженностью подхода, основанного на шаблонах (см. Рис 6). К примеру, *SyntacticContext* достигает полноты 0.744, в то время как Серелекс достигает полноты около 0.389 [2]. При оценке полноты следует также учитывать, что количество семантических

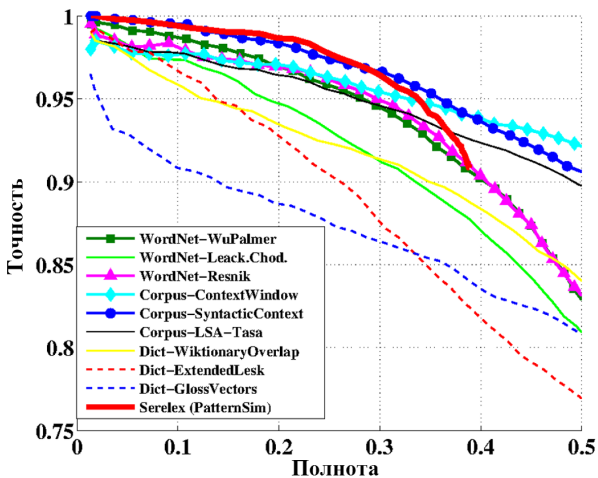


Рис. 6. Результаты: задача ранжирования семантических отношений.

отношений распределено экспоненциально [18]. Поэтому большинство слов имеют только около 10-100 семантически связанных слов.

3.3 Извлечение семантических отношений

Кроме двух описанных выше тестов, была оценена точность извлечения семантических отношений для 49 слов из лексикона *RG*. В данном эксперименте трем ассессорам было предложено аннотировать результаты поиска и указать для каждого из 50 первых результатов, является ли он релевантным или нет. Например, для запроса “fruit”:

- 1; vegetable (relevant)
- 2; mango (relevant)
- ...
- 50; house (non-relevant)

На основании полученной статистики вычислена точность для k первых результатов, где $k \in \{1, 5, 10, 20, 50\}$. Согласно результатам данного эксперимента, приведенным на Рис 7 (а), средняя точность извлечения варьируется между 74% (для первого результата, $k = 1$) и 56% (50 первых результатов, $k = 50$). Мы зафиксировали значительную степень согласия ассессоров для данного эксперимента в терминах кашы Флейса: 0.61–0.80.

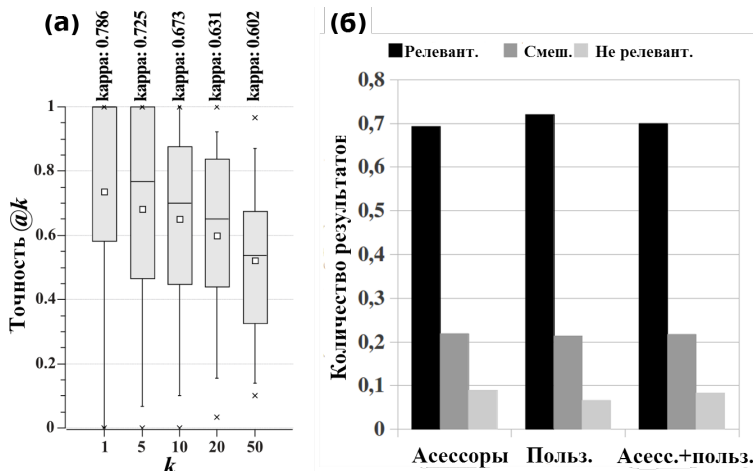


Рис. 7. Результаты: (а) задача извлечения семантических отношений; (б) удовлетворенность пользователей первыми 20 результатами поиска.

3.4 Удовлетворенность пользователей качеством поиска

Каждому из 23-х ассесоров, участвующих в исследовании, было предложено выбрать 20 запросов по своему усмотрению и оценить первые 20 результатов поиска как релевантные, нерелевантные или как частично релевантные. В результате данной оценки было собрано 460 суждений ассесоров и 233 суждения анонимных пользователей системы. Пользователи и ассесоры вместе осуществили 594 уникальных запроса. В соответствии с этим экспериментом, результаты поиска являются релевантными для 70% запросов и нерелевантными для 10% запросов (см. Рис 7 (б)). В 20% случаев первые 20 результатов оказались частично релевантными.

4 Выводы

Разработана система Серелекс, которая позволяет осуществлять поиск семантически связанных слов. Оценка качества работы системы на четырех экспериментах показала, что точность системы сопоставима с аналогичными существующими разработками. При этом, в отличие от большинства аналогов, Серелекс не использует составленные вручную словари. За счет этого достигается лучшее лексическое покрытие, так как семантические отношения извлекаются непосредственно из текста. Опрос пользователей показал, что

первый результат поиска релевантен в 74% случаях и в 70% запросов пользователи полностью удовлетворены первыми 20 результатами.

Мы работаем над построением аналогичной системы для французского и русского языков. При адаптации системы к новому языку мы планируем перевести набор шаблонов разработанных для английского языка. При этом будут использоваться стандартные словари и средства морфологического анализа, включенные в Unitex. Кроме того, мы работаем над интеграцией в систему модуля распознавания имен собственных (Named Entities Recognition), что позволит извлечь отношения не только между словами и словосочетаниями из словаря, но и между названиями компаний, именами публичных людей и т.п.

Список источников

1. Panchenko, A., Romanov, P., Morozova, O., Naets, H., Romanov, A., Philippovich, A., Fairon, C.: Serelex: Search and visualization of semantically related words. In Proceedings of the 35th European Conference in Information Retrieval (2013)
2. Panchenko, A., Morozova, O., Naets, H.: A semantic similarity measure based on lexico-syntactic patterns. In: Proceedings of KONVENS 2012. (2012) 174–178
3. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: ACL. (1992) 539–545
4. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: A collection of very large linguistically processed web-crawled corpora. LREC **43**(3) (2009) 209–226
5. Miller, G.A., Leacock, C., Teng, R., Bunker, R.T.: A semantic concordance. In: Proceedings of the workshop on Human Language Technology, ACL (1993) 303–308
6. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. ACM **8**(10) (1965) 627–633
7. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. In: WWW 2001. (2001) 406–414
8. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL'1994. (1994) 133–138
9. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification. WordNet (1998) 265–283
10. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: IJCAI. Volume 1. (1995) 448–453

11. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: IJCAI. Volume 18. (2003) 805–810
12. Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. EACL 2006 (2006) 1–12
13. Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from wikipedia and wiktioary. In: LREC'08. (2008) 1646–1652
14. Van de Cruys, T.: Mining for Meaning: The Extraction of Lexico-Semantic Knowledge from Text. PhD thesis, University of Groningen (2010)
15. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. *Discourse processes* **25**(2-3) (1998) 259–284
16. Baroni, M., Lenci, A.: How we blessed distributional semantic evaluation. In: GEMS (EMNLP), 2011. (2011) 1–11
17. Panchenko, A., Morozova, O.: A study of hybrid similarity measures for semantic relation extraction. Innovative hybrid approaches to the processing of textual data workshop of EACL 2012 (2012) 10–18
18. Panchenko, A.: Similarity Measures for Semantic Relation Extraction. PhD thesis, Université catholique de Louvain (2013)

Методология создания программного комплекса «Интерактивная информационная доска»

Дроздова Юлия Александровна

ПГНИУ, Пермь, Россия.

Аннотация. Статья посвящена вопросу создания программного комплекса «Интерактивная информационная доска», который является альтернативой информационных стендов учебных заведений. Комплекс использует технологию дополненной реальности, в основе которой лежит компонента распознавания маркеров. В статье рассмотрен принцип работы комплекса. Особое внимание уделяется методологии создания компонента обнаружения и распознавания маркеров дополненной реальности.

Ключевые слова: дополненная реальность, контурный анализ, ORB (Oriented-BRIEF), метод потенциальных функций.

Введение

Учебный процесс любого ВУЗа сопровождается большими потоками информации. В этом можно убедиться, взглянув на информационные доски различных кафедр университетов. Однако с увеличением объема данных размещаемых на доске понижается эффективность обмена информацией между студентами и преподавателями. От того, насколько быстро и своевременно будет получена эта информация, зависит результативность организации учебного процесса. Для повышения продуктивности использования данных, расположенных на информационных стендах учебных заведений предлагается внедрить технологию

Методология создания программного комплекса «Интерактивная информационная доска» дополненной реальности. На основе этой технологии был разработан программный комплекс «Интерактивная информационная доска».

Схема работы программного комплекса

Для пользователя интерактивная доска представляет собой набор из нескольких распечатанных изображений, которые означают основные информационные группы, расположенные на доске. Пользователю необходимо навести камеру мобильного устройства на изображения для получения списка пунктов меню доски конкретной кафедры (или университета) и выбрать интересующий его пункт. Таким образом пользователь получает возможность взаимодействовать с информационным контентом доски через мобильное устройство: автоматизировано искать интересующую его информацию, делать заметки, сохранять расписание – использовать все преимущества цифрового мира. Содержание такой интерактивной информационной доски может быть настроено для каждого конкретного пользователя, чего нельзя добиться, используя физические аналоги досок.

Программный комплекс разделен на 2 основные части (рис. 1): клиентская, расположенная на мобильном устройстве и серверная.

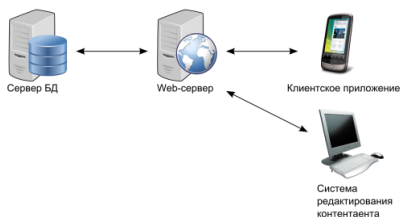


Рис. 1. Архитектура программного комплекса.

Серверная часть включает компоненту редактирования и формирования контента информационных досок, на мобильном устройстве расположена компонента обнаружения и распознавания маркеров, которой уделено особое внимание.

Был произведен анализ данных расположенных на информационных стендах кафедр университета. В результате построена иерархия объектов используемых для управления контентом. Были выделены основные информационные группы, в соответствие каждой информационной группе определен маркер дополненной реальности. Компонента обнаружения и распознавания обрабатывает изображение, полученное с камеры мобильного устройства, и формирует список пунктов меню, доступных пользователю, основываясь на обнаруженных маркерах. Таким образом, задача разработки компоненты обнаружения и распозна-

Методология создания программного комплекса «Интерактивная информационная доска»
вания маркеров становится одной из основных при создании программного комплекса.

Обзор существующих решений

В основе комплекса лежит технология дополненной реальности, которая подразумевает дополнение реального мира виртуальными объектами. Для того, чтобы принять решения о том, какую информацию требуется вывести на экран используемого устройства, система должна обнаружить и идентифицировать интересующий пользователя объект. Существуют два основных подхода к идентификации объектов окружающего мира: идентификация по положению объекта в пространстве и идентификация объекта по его изображению.

Нами была рассмотрена платформа дополненной реальности Layar. Основной способ определения точек расширения в Layar – анализ GPS-координат. Однако технология, на которой базируется проект, не применима в тех случаях, когда информационные доски находятся внутри одного помещения. Следовательно, для определения позиций размещения дополнительного контента необходимо использовать принципы видеозрения. Мы рассмотрели библиотеки ARToolKit и ARTag, которые ориентированы на поиск маркеров дополненной реальности. Однако библиотеки способны распознавать только монохромные маркеры, и содержание распознаваемых маркеров выбирается не с учетом проблемной области, а с учетом особенностей алгоритма распознавания.

Таким образом, нам необходимо разработать собственный компонент, распознавания и обнаружения маркеров, который устраняет вышеперечисленные недостатки рассмотренных библиотек дополненной реальности. При разработке компонента следует учитывать особенности его применения: компонент используется на мобильном устройстве. Следовательно, алгоритмы обнаружения и распознавания маркеров должны быть не только точными, но и эффективными. Изображения, используемые в качестве маркеров должны выбираться с учетом предметной области. Для реализации поставленной задачи нам была выбрана библиотека компьютерного зрения OpenCV, которая предоставляет возможности обработки и анализа цветных изображений. Однако, в отличие от ARTag и ARToolKit, OpenCV не предоставляет возможности поиска на изображении маркеров. Перед нами встает необходимость разработать алгоритм обнаружения и распознавания маркеров дополненной реальности, использующий существующие подходы к анализу изображений.

Методология создания программного комплекса «Интерактивная информационная доска»

Компонент обнаружения и распознавания маркера

Разработка была разделена на следующие этапы: формирование маркера дополненной реальности, обнаружение маркера в видеопотоке, его распознавание. Все этапы разработки взаимосвязаны: от эффективности решения проблем предыдущего этапа зависит эффективность работы текущего этапа разработки.

Формирование маркера

В теории, маркером дополненной реальности является любой объект. Однако следует учесть, что на практике мы ограничены особенностями цветопередачи, разрешением камеры и вычислительной мощностью используемого мобильного устройства

Одним из фундаментальных этапов анализа видеoinформации является сегментация изображения, означающая разбиение поступающего изображения на множество областей, ассоциируемых с объектами наблюдаемой сцены [1]. Нас интересуют такие сегменты изображения, как тела маркеров. Для повышения эффективности алгоритма обнаружения маркера мы ввели ограничения на его внешний вид: форма – квадратная, тело маркера помещено в рамку черного цвета, маркеры располагаются на белом фоне (рис. 2).



Рис. 2. Распознаваемый маркер

Дополнительные точки, определяющие местоположение камеры относительно маркера не использовались.

Обнаружение маркера

Сравнение целых изображений требуют больших вычислительных затрат нежели сравнение их частей. Для сужения пространства распознавания мы использовали сегментацию изображения на основе контурного анализа. В результате алгоритм обнаружения маркера включает следующие этапы: бинаризация изображения, выделение контуров, поиск контура маркера.

Были рассмотрели модели адаптивной (равнозначный вес пикселей и вес пикселей задан Гауссианом) и глобальной бинаризации (вес пикселей равнозначен и пороговое значение определено экспериментально) с целью выбрать модель с минимальной вычислительной сложностью и

Методология создания программного комплекса «Интерактивная информационная доска» максимальной результативностью. Вычислительная сложность модели адаптивной бинаризации зависит от размера ядра бинаризации, и она заведомо выше сложности алгоритмов глобальной бинаризации. Таким образом, при практически равной результативности мы отдали предпочтение алгоритму глобальной бинаризации с пороговым значением, которое было установлено экспериментальным путем. Выбранный вид бинаризации позволяет обнаруживать маркеры при дневном освещении почти в 100% случаев.

На бинарном изображении, с помощью алгоритма, представленного в статье[3] выделяются контуры, и компонента обнаружения переходит к этапу обнаружения контура тела маркера. На основе принципа вложенности компонента обнаружения строит B-дерево из выделенных контуров. В таком дереве объемлющий контур — родительская вершина дерева, внутренние контуры — его потомки. Под контуром тела маркера следует понимать квадратный контур, родительский контур которого тоже является квадратным. Определим квадратный контур как замкнутый контур, который имеет 4 точки излома и углы между соседними сегментами контура близкие к 90° . Для обнаружения контуров тел маркеров необходимо совершить обход дерева и выбрать контуры, удовлетворяющие вышеописанным условиям. Таким образом, на рисунке 3 контур тела маркера выделен желтым цветом, а контур тела маркера — зеленым. Далее можно приступить к распознаванию тела маркера.



Рис. 3. Результат работы алгоритма обнаружения маркера

Распознавание маркера

Под телом маркера следует понимать сегмент изображения ограниченный соответствующим контуром. На основе анализа изображения тела маркера весь маркер будет классифицирован. Процесс классификации заключается в определении наиболее похожего изображения из n эталонных по определенному пространству признаков. Мы выделили следующие этапы распознавания маркеров: выделение особенностей изображений, разработка классификационных правил на основе особенностей, коррекция результатов классификации. Было решено использовать особые точки изображения как его особенности. Выделим на рас-

Методология создания программного комплекса «Интерактивная информационная доска»

познаваемом объекте ключевые точки и участки вокруг них, построим дескрипторы. Аналогично поступим с эталонными изображениями. Ключевой точкой будем считать такую точку, которая имеет признаки, существенно отличающие ее от основной массы точек. Для решения данной задачи мы выбрали алгоритм FAST and Rotated BRIEF(ORB)[2]. Пример выделения особых точек на изображении представлен на рис. 4. Исследования показывают, что ORB находит меньше особых точек, чем SURF или SIFT, однако обнаруженные точки имеют устойчивость того же порядка, что и точки обнаруженные SURF[4]. Это позволяет применить ORB алгоритм для мобильных приложений.



Рис. 4. Выделение особых точек на распознаваемом объекте

В качестве классифицирующего правила был выбран метод потенциальных функций. Для каждого эталонного изображения произведем отображение его особых точек на особые точки распознаваемого изображения. Отображение основывается на минимальном расстоянии Хэмминга между точками. В результате получаем множество, каждый элемент которого характеризуется расстоянием между точками. Вычислим

потенциал P создаваемый множеством: $P = \frac{\sum_{j=1}^k \frac{1}{1 + \text{Hamming}_j^2}}{K}$, где k —

мощность множества связей. Находим потенциалы отображений всех эталонных объектов и находим среди них максимум. Относим распознаваемое изображение к тому эталону, на чьем отображении был достигнут максимальный потенциал. В среднем метод правильно классифицирует изображения с точностью 98%. Для повышения точности распознавания, нами было принято решение более тщательно выбирать пары особых точек, используемых для классификации. Мы использовали вероятностную функцию плотности отображения: для каждой особой точки эталонного изображения находится не одна, а две наиболее близкие особые точки распознаваемого объекта. Пусть расстояние до ближайшей точки - d_1 , а до второй ближайшей точки - d_2 . Тогда функция оценки плотности совпадений $PDF = d_1/d_2$. Очевидно, что область значения функции $(0; 1]$. Следовательно, пороговое значение будем искать в этих пределах. Если значение функции ниже некоторого порогового

Методология создания программного комплекса «Интерактивная информационная доска»

значения, то для классификации можно использовать расстояние $d1$. На рис. 5 приведена зависимость результатов распознавания от выбора порогового значения.

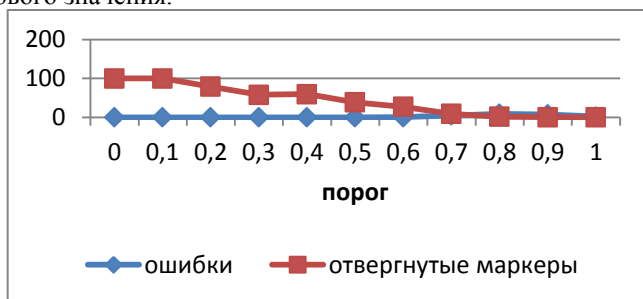


Рис. 5. Зависимость результатов распознавания от порога.

Из рисунка 5 видно, что при низком пороговом значении мы получаем точность распознавания близкую к 100% (синим цветом обозначен процент ошибок классификации первого рода), однако большинство распознаваемых объектов невозможно классифицировать (красный цвет), т.к. нет пар особых точек удовлетворяющих пороговому значению. Потенциалы таких изображений равны 0. Т.е. распознавание становится более точным, но менее стабильным. Для стабилизации распознавания мы ввели учет предыдущего кадра: если значение потенциала предыдущего кадра больше значения потенциала текущего кадра, то производим замену текущего потенциала и соответственно результата классификации. Результат такой коррекции представлен на рис. 6.

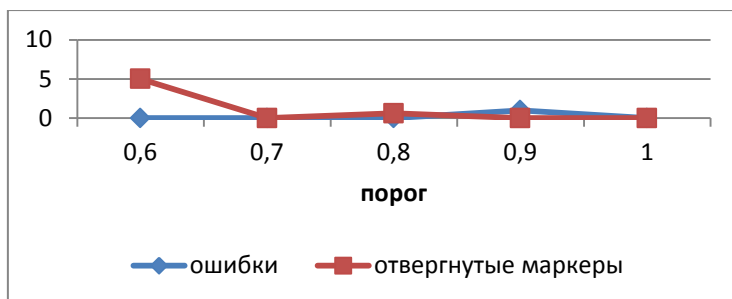


Рис. 6. Результаты распознавания со стабилизацией

Анализируя рисунок 6, мы пришли к выводу, что оптимальное пороговое значение равно 0,7. При этом точность составила около 100%.

Для тестирования компонента было использовано около 300 изображений на каждый класс маркеров (напомним, что в работе 5 ис-

Методология создания программного комплекса «Интерактивная информационная доска» (следующих классов). Изображения были получены с камеры мобильного устройства с разрешением 5mpx при дневном освещении. При выборе тестовых изображений учитывались угол обзора (отклонение до 80° от перпендикуляра к плоскости) и удаленность камеры (маркер занимает от 10% изображения).

Выводы

Нами был разработан комплекс «Интерактивная информационная доска», использующий технологию дополненной реальности.

В рамках комплекса был создан метод распознавания маркеров, который объединяет существующие подходы анализа изображений и адаптирует их для мобильных устройств. В методе использовались элементы контурного анализа, метод поиска особых точек изображения и метод потенциальных функций. Разработанный метод позволяет классифицировать маркеры с точностью до 100%, что позволяет говорить о его применимости в реальной системе.

Разработанный программный комплекс уже реализован и готов к практическому использованию.

Список источников

1. Chochia P. A Pyramidal Image Segmentation Algorithm // Journal of Communications Technology and Electronics, 2010, V.55, No.12, pp.1550-1560.
2. Ethan Ruble, Vincent Rabaud, Kurt Konoligi Gary ORB: an efficient alternative to SIFT or SURF
3. Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)
4. Comparison of the OpenCV's feature detection algorithms-II. URL: <http://computer-vision-talks.com/2011/07/comparison-of-the-opencvs-feature-detection-algorithms-ii/>

Применение методов машинного перевода для анализа древнерусских музыкальных рукописей

Марина Даньшина¹, Андрей Филиппович²

¹МГТУ имени Н.Э. Баумана, Москва, Россия. marina_danshina@mail.ru

²МГТУ имени Н.Э. Баумана, Москва, Россия. philippovich@list.ru

Аннотация. Статья посвящена применению методов машинного перевода для дешифровки древнерусских музыкальных рукописей из знаменной нотации в современную линейную нотацию. Рассматривается обработка исходных рукописей для построения итогового словаря для перевода. Приведен пример языковой модели и модели перевода.

Ключевые слова: знаменные песнопения, статистический машинный перевод, семиография, дешифровка, древние рукописи, модель языка, модель перевода, переводной словарь, продукционная модель

Введение

Предшественником линейной нотации, используемой в настоящее время для фиксирования мелодии, была знаменная нотация. Ее особенностью было то, что мелодия записывалась не с помощью нот на линейках, а специальными знаками – крюками, которые имели сложную структуру. Изначально рукописи не содержали подсказок исполнителю о высоте или длительности ноты, однако спустя некоторое время в музыкальные книги стали добавлять пометы, облегчающие чтение песнопения. С течением времени знания о том, как необходимо воспроизводить мелодию стали фиксировать в специальных книгах (азбуках), поз-

же появились рукописи, содержащие мелодию в двух нотациях – знаменной и нотной. Такие книги являются аналогами параллельных корпусов текстов и именно они являются главным источником информации для расшифровки знаменных песнопений, несмотря на то, что данные в них неполны и иногда противоречивы. Помимо этого следует учитывать, что рукописи содержат специальные структуры (фиты, лица), которые, аналогично фразеологизмам в тексте, необходимо переводить особым образом. [7]

Общее количество знамен, с помощью которых производилась запись, оценивается по-разному. В нашем исследовании были экспериментально выявлено 202 знамени. При этом каждое знамя может переводиться одной или несколькими нотами. Пример знаменной рукописи приведен на рисунке 1.



Рис. 1. Пример знаменной рукописи

Обработка рукописей

В рамках проекта «Компьютерная семиография» реализуются задачи по созданию конкретных инструментов, позволяющих автоматизировать рутинные операции перевода знаменных песнопений в линейную нотацию. Данная работа поддержана грантом РГНФ №110412025в.

В качестве основных исходных данных были выбраны четыре типа музыкальных рукописей:

- музыкальные азбуки;
- кокизники (сборники фит и лиц);
- сборники попевок;
- двознаменники.

Для обработки каждого типа рукописей предложены отдельные инструменты и технологии. Например, для перевода на основе азбук можно составить список продукционных правил с приоритетами и осуществить экспериментальную дешифровку. Для этого создан музыкальный проигрыватель, который показывает результаты перевода не только визуально, но и позволяет проанализировать мелодию на слух. Приоритеты используются в тех случаях, когда при дешифровке нужно перевести сочетания знамен.

Пример результата перевода музыкальным проигрывателем приведен на рисунке 2.

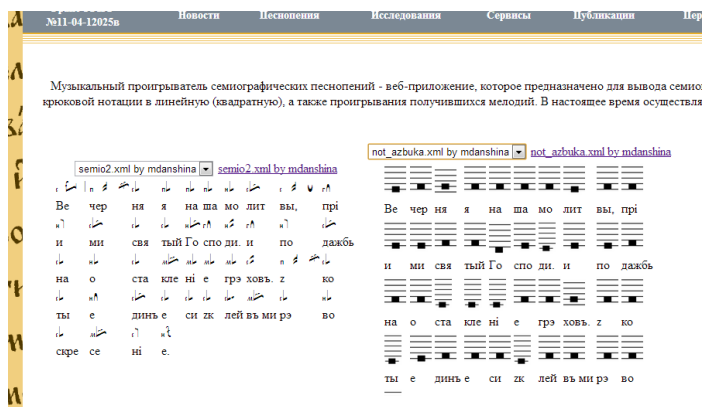


Рис. 2. Пример перевода музыкальным редактором

Исходными данными для музыкального редактора являются:
1. Знаменное песнопение в формате XML (Рис. 3)

```

▼<ROWDATA>
<ROW Znam="a" Slog="ко" Stil=" обычный" VPom="м" DPom=""/>
<ROW Znam="Ap" Slog="кэч" Stil=" обычный Italic" VPom="в" DPom=""/>
<ROW Znam="a" Slog="но" Stil=" Bold" VPom="н" DPom=""/>
<ROW Znam="a" Slog="му" Stil=" Bold" VPom="н" DPom=""/>
<ROW Znam="a" Slog="т" Stil=" Bold" VPom="н" DPom=""/>
<ROW Znam="a" Slog="от" Stil=" Bold" VPom="н" DPom=""/>
<ROW Znam="a" Slog="шу" Stil=" Bold" VPom="н" DPom=""/>
<ROW Znam="a" Slog="ся" Stil=" Italic" VPom="н" DPom=""/>
    
```

Рис. 3. Пример песнопений в XML-формате

2. Словарь для перевода в формате XML (Рис. 4), в котором закодированы ноты, которыми переводится знамя или последовательность знамен, а также соответствующие длительность и приоритет.

```
▼<ROWDATA>
  <ROW cod="200" note="e_1" length="04" p="0.7"/>
  <ROW cod="144" note="f_1" length="04" p="0.9"/>
  <ROW cod="304" note="g_1" length="04" p="0.9"/>
  <ROW cod="269" note="f_1" length="04" p="0.4"/>
  <ROW cod="177" note="e_1" length="04" p="0.4"/>
  <ROW cod="256" note="f_1" length="04" p="0.5"/>
```

Рис. 4. Пример словаря для дешифровки рукописи

Данные словари строятся на основе предварительно введенных в базу данных рукописей (песнопений и азбук), а также на основе результатов построения модели перевода.

Для анализа двузнаменников разработаны и апробированы различные технологии статистического перевода:

- методы построения "модели языка": на основе N-граммной модели – вероятность следования знамени определяется с учетом вероятностей предшествующих знамен.

- построение "модели перевода" в зависимости от характера "знаменных конструкций" (их размерности) может быть реализовано на основе:

- 1) текстовых фраз, которые сопровождают нотную запись – выбираются последовательности знамен, соответствующие предложению или его части (до знака препинания);

- 2) попевок – устойчивых сочетаний знамен из соответствующих сборников, составленных вручную древними авторами или исследователями;

- 3) фиксированного контекстного окна – выбранного количества знамен (используется в N-граммной модели).

Построение модели языка и модели перевода

В качестве модели языка строится триграммная языковая модель. Исходными данными для построения является двоезнаменный Ирмологий, который переведен в электронный вид и хранится в базе данных (Рис. 5).



Рис. 5. Пример двузначника в электронном виде

Согласно статистическому машинному переводу модель языка назначает наибольшую вероятность наиболее частотным строкам (словам или фразам). В качестве «граммы» для знаменных песнопений выбраны последовательности нот, которые соответствуют знамени.

Входными данными является упорядоченный массив T , элементами которого является ноты в том порядке, в котором они следуют в исследуемой рукописи.

$$T(r) = \varphi_1(r) = \{t_1, t_2, \dots, t_m\}, \text{ где } r - \text{исследуемая рукопись, } t - \text{нота, } m - \text{количество нот в рукописи } r.$$

Данный массив разбивается на триграммы TrT :

$$TrT(r) = \varphi_2(T(r)) = \{\{t_i, t_{i+1}, t_{i+2}\}\}, \text{ где } t \in T(r)$$

Для учета недостатка неполноты исходных данных используется метод сглаживания Лапласа, в соответствии с которым вероятность каждой n -граммы вычисляется следующим образом:

$$P_{lap}(w_i | w_{i-1}) = \frac{c(w_{i-1}w_i) + 1}{c(w_{i-1}) + |V|} \quad (1)$$

Где c – исходное количество триграммы в тексте, $|V|$ – число уникальных грамм в тексте [5,6,9].

Результаты построения языковой модели представлены в электронном виде на сайте проекта.

Выходными данными является массив триграмм, каждой из которых ставится в соответствие вероятность встречаемости этой триграммы в песнопении.

$$LM(r) = \varphi_3(TrT(r)) = \{\langle t_i, t_{i+1}, t_{i+2} \rangle, p\}$$

N-грамма	Вероятность со сглаживанием	Вероятность без сглаживания	N
	0,01139	0,666667	3
	0,015909	0,857143	2
	0,00431	0,032258	3
	0,056338	0,22963	2
	0,004348	0,037037	3
	0,025271	0,04	2
	0,004228	0,025	3
	0,049878	0,102828	2

Рис. 6. Пример языковой модели песнопений

Модель перевода вычисляется по двуязычному корпусу и назначает наибольшую вероятность парам строк (слов или фраз) с одним значением.

Для построения модели перевода рассчитывается вероятность для каждой пары $P(n|z)$, где z – последовательность знамен, а n – перевод этой последовательности. Данная вероятность рассчитывается по формуле (2.2).

$P(n|z) = \frac{C(n,z)}{C(z)}$ (2), где $C(n,z)$ – количество раз, когда последовательность знамен z переводится нотами n .

На рис.7 представлен пример модели перевода.

Для каждой триграммы ставится в соответствие ноты, которыми переводятся знамена, входящие в триграмму, а также вероятность встречаемости триграммы

$$TM(r) = \varphi_4(TrT(r)) = \{\langle z_i, z_{i+1}, z_{i+2} \rangle, \langle t_i, t_{i+1}, t_{i+2} \rangle, p\}, \text{ где } z_j \in RZ(r), RZ(r) = \varphi_5(r) = \{z_1, z_2, \dots, z_m\}$$

Триграмма			Перевод			Вероятность
						0,017327
						0,318182
						0,014851
						0,272727
						0,073529
						0,3125
						0,153846
						0,111111
						0,333333
						0,2

Рис. 7. Пример языковой модели песнопений

Построение общего словаря для расшифровки знаменных песнопений

Итоговый словарь строится на основе предварительно полученных на основе азбук, сборников попевок и двознаменников словарей. А также у экспертов имеется возможность добавить в словарь новые правила, полученные на основе анализа материалов. При этом каждому правилу в словаре соответствует приоритет, который определяет очередность замены знамен. Таким образом, учитывается перевод попевок и других специальных структур.

Выводы

1. Полученный словарь позволяет перевести древнерусские музыкальные рукописи из крюковой нотации в линейную с учетом особенностей знаменных песнопений.

2. Использование нескольких методов построения словарей позволяет исследователям различным образом анализировать полученные переводы.

3. Построенные модель языка и модель перевода являются исходными данными для следующего этапа расшифровки знаменных песнопений – декодирования.

Список источников

1. Даньшина И.В., Даньшина М.В. Структура и обработка древнерусских певческих рукописей.// Сборник тезисов докладов «Печатные средства информации в современном обществе (к 80-летию МГУП)». Секция «Электронные средства информации в современном обществе», М. 2010
2. Даньшина М.В. Программа для ввода и обработки семиографических песнопений IPSM. Информационные технологии и письменное наследие: материалы междунар. науч. конф. / отв. Ред. В.А.Баранов. - Уфа;Ижевск: Вагант, 2010
3. Даньшина М.В. Метод выделения, сохранения и обработки попевок в музыкальной рукописи. Информационные технологии и письменное наследие: материалы IV междунар. науч. конф. (Петрозаводск, 2012 г.)
4. Даньшина М.В. Исследование семантической структуры попевок в знаменных песнопениях. Тезисы докладов на международном междисциплинарном форуме по прикладной когнитивистике CrossLingua'2012 “Когниция. Коммуникация. Культура”
5. Даньшина М.В. Использование n-граммной языковой модели для изучения знаменных песнопений. Сборник тезисов и статей Российско-Германской молодежной дистанционной научной школы «Актуальные и перспективные направления создания систем, обеспечивающих семантический анализ данных в режиме реального времени», 2012.
6. Knight K. A Statistical MT Tutorial Workbook. 1999
7. Бражников М.В. «Древнерусская теория музыки». – «Музыка», 1972г.
8. Шабалин Д.С. Певческие азбуки Древней Руси. – Кемерово: Кузбассвуиздат, 1991
9. Bird S., Klein E., Loper E. Natural Language Processing with Python, O'Reilly, 2009

Автоматическое извлечение правил для снятия морфологической неоднозначности

Екатерина Протопопова, Виктор Бочаров

СПбГУ, Санкт-Петербург, Россия, protoev@gmail.com,
victor.bocharov@gmail.com

Аннотация. Морфологическая неоднозначность представляет собой основную сложность при решении задачи морфологического анализа текстов. Известные методы снятия неоднозначности используют большое количество лингвистических данных, полученных вручную (правила или корпуса текстов с разметкой). В статье описана попытка реализации алгоритма, не требующего таких ресурсов, приведены различные оценки результатов работы данного алгоритма.

Ключевые слова: морфологическая разметка; русский язык; омонимия; корпуса текстов

Введение

Морфологическая неоднозначность (омонимия) считается одной из основных проблем при морфологическом анализе текстов, поэтому основные усилия при создании морфологических анализаторов направлены именно в сторону снятия омонимии. Среди известных подходов к разрешению неоднозначности принято выделять детерминированные (основанные на правилах) и вероятностные. Эти подходы во многом основаны на данных, полученных вручную: первые используют правила, написанные лингвистами, вторые – большие вручную размеченные корпуса текстов. В данной работе предпринимается попытка создания

инструмента для снятия морфологической омонимии с использованием небольшого количества лингвистических ресурсов. Также оценивается размер необходимого и достаточного для обучения системы корпуса.

Используемый нами подход (в некоторых работах называемый комбинированным) впервые описан в работе [1] и был применен для английского языка. Следует сразу отметить, что этот алгоритм в нашей работе применялся только для снятия частеречной омонимии. Работа алгоритма сводится к следующему. Из автоматически размеченного корпуса собирается статистическая информация о встречающихся частеречных тегах и их окружении (контекстах). На основе этой статистической информации выводятся правила преобразования омонимичных тегов в неомонимичные, затем каждому правилу приписывается вес, полученный с помощью специальной функции оценки.

Лучшее правило применяется к разметке корпуса, затем процедура повторяется до тех пор, пока лучшее правило имеет положительный вес. Лучшие правила, которые получаются на каждом шаге, сохраняются и могут затем применяться при разметке другого корпуса.

Для оценки размера необходимого корпуса было проведено несколько экспериментов. Система обучалась на корпусах различных размеров – от тысячи предложений до 170 тысяч – и таким образом были получены различные списки правил, которые потом сравнивались между собой.

Предыдущие работы

Все известные нам алгоритмы снятия морфологической неоднозначности для русского языка были обучены на вручную размеченных данных Национального корпуса русского языка (ruscorpora.ru) и представляют собой вероятностные модели. Алгоритм, описанный в [3], основан на словарях контекстов; вероятность выбора леммы вычисляется как сумма вероятностей порождения леммы элементами контекста. Кроме того, каждому элементу контекста приписан экспериментально полученный вес, показывающий степень влияния данного элемента на выбор правильного разбора. Точность работы алгоритма составляет около 97.4%.

В работе [4] описывается алгоритм, основанный на скрытой марковской модели. Используется триграммная модель для тегов (вероятность того, что тег 2 появится после тега 1 и тега 0) и биграммная модель для слов (вероятность того, что определенному слову будет приписан тег 1 после тега 0). В качестве обучающего набора использовался подкорпус со снятой омонимией из НКРЯ (5 миллионов слов). Точность работы алгоритма достигает 98%.

Автоматическое извлечение правил для снятия морфологической неоднозначности

Похожий подход (с некоторыми дополнениями) представлен в работе [2]. Используя теггер TnT и тот же подкорпус со снятой омонимией, авторы показали, что задача снятия морфологической неоднозначности может быть решена с достаточно высокой точностью (97%) без дополнительного обращения к лингвистическим ресурсам.

Алгоритм и используемые данные

Исходный алгоритм

Основная идея алгоритма была описана выше, здесь мы постараемся более подробно изложить принципы его работы. Текст, используемый в качестве обучающего набора, размечается неоднозначно, то есть каждому слову приписываются все возможные варианты его морфологического разбора. Затем собирается статистическая информация о тегах и контекстах, в которых они встречаются. Для каждого тега X подсчитывается $\text{freq}(X)$ – абсолютная частота тега и $\text{incontext}(X, C)$ – частота тега X в контексте C . Далее для каждого омонимичного тега рассматриваются различные варианты снятия омонимии: для каждого возможного варианта вычисляется параметр $\frac{\text{freq}(Y)}{\text{freq}(Z)} \cdot \text{incontext}(Z, C)$, где $Z \neq Y$. Из $Z \neq Y$ выбирается тег R , для которого значение этого параметра максимально. На основе этих данных составляются правила преобразования омонимичных тегов в неомонимичные:

Заменить тег X на тег Y в контексте C ;

каждому такому правилу приписывается вес:

$$\text{score} = \text{incontext}(Y, C) - \frac{\text{freq}(Y)}{\text{freq}(R)} \cdot \text{incontext}(R, C)$$

$$\text{где } R = \underset{z}{\text{argmax}} \frac{\text{freq}(Y)}{\text{freq}(Z)} \cdot \text{incontext}(Z, C), Z, Y \in x, Z \neq Y, \text{freq}(Z)$$

– частота тега Z в корпусе, $\text{incontext}(Z, C)$ – частота тега Z в контексте C .

На каждом шаге алгоритм находит правило с наибольшим весом, обучение продолжается, пока вес лучшего правила положителен. При тестировании на наборе размером 200 тысяч слов из Penn Treebank алгоритм показал точность 95.1%, на наборе размером 350 тысяч слов из Брауновского корпуса – 96.0%.

Отличия в реализации и использованные данные

Стоит отметить, что мы применяли алгоритм только для снятия частеречной омонимии, хотя в русском языке сильно распространены и другие случаи морфологической омонимии – например, падежная омонимия. Кроме того, мы использовали только четыре контекстных признака – по одному слову и тегу справа и слева. Правила записывались следующим образом:

ADJF NOUN → NOUN | 1:tag=PNCT

то есть «Заменить тег ADJF NOUN на тег NOUN, если следующий тег – PNCT».

В качестве обучающих корпусов для получения правил мы использовали наборы предложений разного размера, выбранные случайным образом из корпуса статей с сайта <http://www.chaskor.ru/>. Корпус объемом 15 миллионов токенов был размечен с помощью словаря проекта OpenCorpora (<http://opencorpora.org/>); использовалась упрощенная разметка следующего вида:

2 Школа 393872 школа NOUN inan femn sing nomn

Слова, отсутствующие в словаре, размечались тегами UNKN (неизвестная последовательность кириллических символов), LATN (неизвестная последовательность символов латиницы), NUMR (цифры) and PNCT (знаки препинания).

Результаты

Одной из основных целей работы было определение необходимого и достаточного объема корпуса для получения набора правил, дающего достаточную точность разметки. Для решения этой задачи было проведено несколько экспериментов. Как было сказано выше, на корпусах разного размера – от тысячи до 170 тысяч предложений – были получены различные наборы правил, которые сравнивались между собой.

Наборы правил для снятия неоднозначности

Наиболее очевидный способ сравнить между собой различные списки правил – сравнить их размер и содержание. Результаты (рис.1) подтверждают наше предположение о том, что количество правил увеличивается при увеличении размера обучающего корпуса. Это в основном связано с тем, что правила основываются на контекстных признаках, разнообразие которых увеличивается при увеличении корпуса. С другой стороны, стоит отметить, что количество правил, использующих частеречный тег, стабилизируется на больших корпусах, что объясняется ограниченным количеством частеречных тегов в целом.

Кроме того, для каждого двух наборов правил, полученных на одном размере корпуса, был вычислен коэффициент ранговой корреляции Спирмена. Стоит, однако, отметить, что сравнивались только правила, встреченные в обоих наборах. Наблюдается (рис.2) увеличение значения коэффициента корреляции при увеличении размера корпуса, что, вероятно, свидетельствует о том, что правила, полученные на больших корпусах, располагаются в схожем порядке.

Автоматическое извлечение правил для снятия морфологической неоднозначности

Результаты применения правил без эталона

С другой стороны, оценить, является ли данный набор правил достаточным для решения задачи снятия морфологической омонимии, можно, применяя различные наборы правил к тестовому корпусу и сравнивая результаты. В качестве такого тестового корпуса была взята случайная выборка объемом 1000 предложений, выбранных случайно из корпуса текстов проекта OpenCorpora. Полученные наборы правил применялись к тестовому корпусу, затем сравнивались результаты разметки. При увеличении размера обучающего корпуса наблюдается увеличение согласованности – количество слов, размеченных по-разному сокращается (Рис.3).

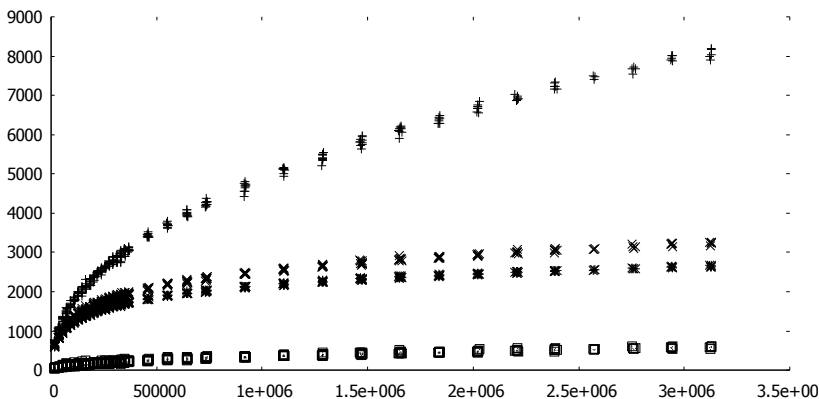


Рис.1. Изменение набора правил

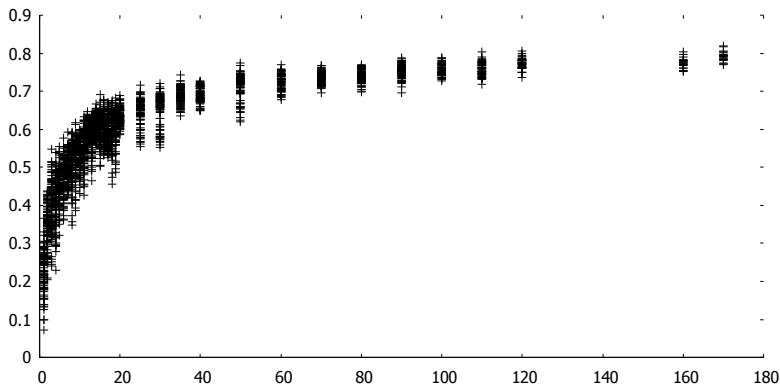


Рис.2. Изменение коэффициента ранговой корреляции (размер корпуса - в тысячах предложений)

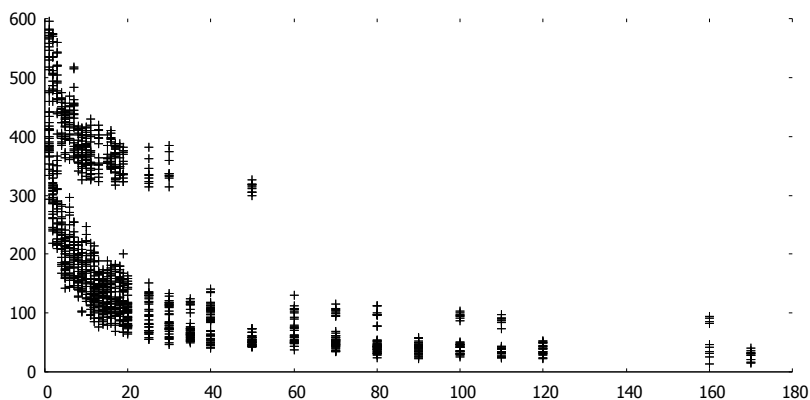


Рис.3. Количество расхождений при разметке корпуса с использованием разных наборов правил.

Сравнение результатов применения правил с исходной разметкой показывает, что увеличение размеров обучающего корпуса приводит также к увеличению полноты снятия неоднозначности (снижается количество оставшихся слов с омонимичной разметкой). Полнота определена как отношение числа однозначно размеченных слов к размеру корпуса.

Такие результаты объясняются увеличением количества правил при увеличении объема корпуса: большие списки покрывают большее число контекстов.

Оценка точности разметки.

Для оценки правильности результатов разметки был создан эталон разметки – корпус размером около ста предложений (выбранных случайно из корпуса текстов проекта OpenCorpora), омонимия в разметке была снята вручную. Затем тот же корпус был размечен с помощью морфологического словаря и различных списков правил. Для оценки точности результаты сравнивались с эталонной разметкой, определялось количество ошибок и их типы (омонимичный тег, преобразованный неверно или не преобразованный).

Общее количество ошибок изменяется в пределах 70-80 при обучении на корпусе размером 60 тысяч предложений и не уменьшается при увеличении размера обучающего корпуса. При этом количество типов ошибок почти не изменяется при увеличении размера корпуса от 20 тысяч предложений.

Из каждого списка ошибок были выбраны три наиболее частотные, затем ошибки были ранжированы по сумме частот. Результаты представлены в таблице 1.

CONJ_PRCL	3639
ADJF_NPRO	1993
ADJF_NOUN	1406
ADVB_CONJ_NPRO	517
ADJS_ADVB	510
ADVB_CONJ_PRCL	505

Таблица 1. Наиболее частотные ошибки в разметке.

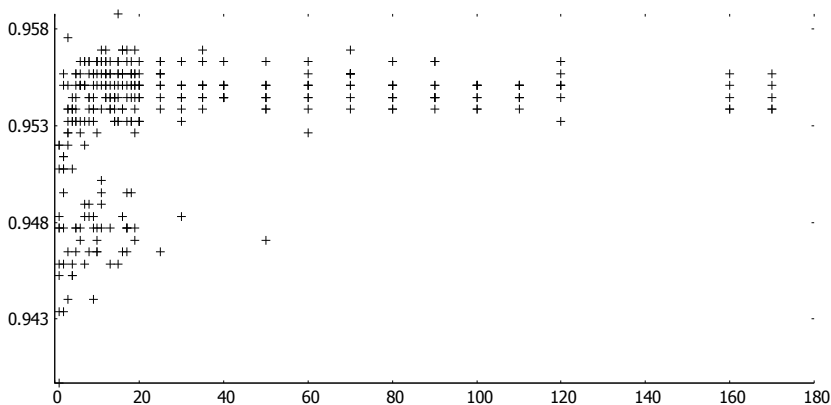


Рис.4. Точность разметки.

Точность разметки (рис.4) вычислялась как отношение числа правильно размеченных токенов к размеру корпуса. На графике сплошной линией показана средняя точность. Следует отметить, что наибольшая точность – 95.7% – получена при обучении на небольшом корпусе (19-20 тысяч предложений), что может объясняться случайным совпадением набора контекстов в обучающем и тестовом корпусах. Кроме того, средняя точность разметки при использовании больших обучающих корпусов может быть признана достаточной, если учитывать небольшое количество использованных контекстных признаков и жанровое своеобразие обучающего корпуса.

Выводы

1. В статье описывается алгоритм извлечения контекстных правил для снятия морфологической неоднозначности, приводятся различные оценки полученных результатов.

2. Показано, что при увеличении размера обучающего корпуса стабилизируются содержание списков правил и результаты их применения.
3. Однозначного ответа на вопрос о достаточном размере обучающего корпуса получено не было, однако результаты экспериментов показывают, что большинство параметров стабилизируются уже на корпусе объемом 60 тысяч предложений.
4. В статье также приведена оценка точности работы описанного алгоритма. Для увеличения точности планируется увеличить число используемых контекстных признаков и оценить работу алгоритма при обучении на корпусах различных жанров.

Список источников

1. Brill E. Unsupervised Learning Of Disambiguation Rules For Part-Of-Speech Tagging. In Proceedings of the Third Workshop on Very Large Corpora, MIT, Cambridge, Massachusetts, USA, 1995.
2. Sharoff S., Joakim Nivre. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 25 - 29 мая 2011 г.).
3. Зеленков Ю.Г. и др. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов./ Зеленков Ю.Г., Сегалович И.В., Титов В.А. // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2005. – М., 2005.
4. Сокирко А.В., Толдова С.Ю. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп). URL: <http://www.aot.ru/docs/RusCorporaHMM.htm>

Применение метода комитета большинства для принятия решения по выдаче кредита

Чернавин Федор

ОАО «Сбербанк России», Екатеринбург, Россия,
ФГАОУ ВПО «УрФУ имени первого Президента России Б.Н.Ельцина», Екате-
ринбург, Россия. Chernavin_fedor@mail.ru

Аннотация. Статья содержит обзор результатов в теории комитетных решений несовместных систем ограничений. Формулируется постановка задачи принятия решения по выдаче кредита большинством членов комитета. Рассматривается сведение задачи построения комитета большинства к задаче частично-целочисленного программирования. Приводятся результаты применения метода комитетов для принятия решений по выдаче кредита.

Ключевые слова: Анализ данных, метод комитетов, анализ заемщика, принятие решений, распознавание образов, дискриминационный анализ, классификация.

Введение

На разработку систем принятия решений по кредитным заявкам банками затрачиваются значительные средства. Большинство зарубежных и российских банков использует в своей практике два метода оценки кредитоспособности: балльные системы оценки (кредитный скоринг) и метод экспертных оценок.

В данной статье рассматривается возможность применения метода комитета большинства при принятии решения о выдаче кредита заем-

щику. Задачей является построение такого комитетного решения, которое бы с высокой точностью соответствовало бы решениям принятым автоматизированной системой банка по конкретной заявке. В случае, если комитетное решение соответствует решениям принимаемым автоматизированной системой банка, то возможно построение системы принятия решений по заявкам на основе метода комитетов.

Банком предоставлены сведения о заявках заемщиков и принятых по данным заявкам решениям. В банке используется автоматизированная система принятия решений по заявкам, механизм которой является коммерческой тайной. Отметим, что ранее в русскоязычной литературе метод комитета большинства не применялся в задачах, связанных с принятием решения по кредитным заявкам.

Понятие комитета впервые появилось в работах по распознаванию образов: в совместной статье Эйблоу и Кейлора [1] было введено понятие комитетного решения для системы строгих однородных линейных неравенств. В более общем виде различные виды обобщения понятия решения на случай несовместных систем были введены в екатеринбургской школе распознавания образов Института математики и механики УрО РАН. Современная теория комитетных конструкций опирается на результаты, полученные Вл.Д. Мазуровым и М.Ю. Хачаем[2-7]. Ими было доказано необходимое и достаточное условие существования комитета несовместной системы линейных неравенств, получены первые оценки числа членов минимального комитета, введены обобщения понятия комитета и получены условия существования для систем линейных неравенств и некоторых более общих систем включений; предложены и обоснованы вычислительные схемы для построения комитетов, в том числе минимальных.

Постановка задачи

Основываясь на статьях Вл. Д. Мазурова и М. Ю. Хачая [3,4,7] сформулируем постановку задачи принятия решения по выдаче кредита комитетом большинства.

Пусть имеются m заемщиков, по которым может быть принято решение выдать кредит или отказать в выдаче кредита. Решение принимается комиссией из q равноправных "экспертов". Договоримся нумеровать членов комиссии индексом t , а заемщиков индексом j . Допустим, что j -му заемщику соответствует множество параметров M_j , а у каждого t -го эксперта имеются соответствующие предпочтения в выдаче кредита в зависимости от параметров заемщика, обозначенные b^t . Через B обозначим множество всех допустимых предпочтений. В случае, если па-

параметры заемщика соответствуют предпочтениям эксперта, то $b^t \in M_j$, данный эксперт голосует за выдачу кредита.

Комитетом системы называется последовательность $Q = (t_1, t_2, \dots, t_q)$, если при каждом $j \in N_p$ справедливо неравенство $|\{i | b^{t_i} \in M_j\}| > \frac{q}{2}$.

При принятии комитетом из q экспертов решения по j заемщику, в случае, если $|\{i | b^{t_i} \in M_j\}| > \frac{q}{2}$, принимается решение выдать кредит, в случае, если $|\{i | b^{t_i} \in M_j\}| < \frac{q}{2}$, принимается решение отказать в выдаче кредита. Далее показано сведение задачи построения комитета большинства к задаче частично-целочисленного программирования.

Сведение задачи построения комитета большинства к задаче частично-целочисленного программирования

Рассмотрим задачу дискриминационного анализа [2], а именно разделения множества одной гиперплоскостью. Пусть заданы множество параметров заемщика $M \subset R^n$ и класс функций $F \subset \{R^n \rightarrow R\}$. Известно, что $M = K_1 \cup K_2$, причем множества K_1 и K_2 заданы своими конечными подмножествами $A \subset K_1, B \subset K_2$. Задачей дискриминантного анализа называется задача нахождения функции $f \in F$ такой, что

$$\begin{cases} f(a) > 0 \text{ при } a \in A, \\ f(b) < 0 \text{ при } b \in B. \end{cases} \quad (1)$$

Найдем такую функцию f , что $K_1 = \{x \in M | f(x) > 0\}$ и $K_2 = \{x \in M | f(x) < 0\}$. Построение данной функции невозможно в случае, если задача несовместна

Пусть $x_j \in K_1$, где $j \in J$ и $x_i \in K_2$, где $i \in I$, J -число элементов первого множества, I - число элементов второго множества.

Поскольку $M \subset R^n$, где n является размерностью пространства (числом параметров заемщика), договоримся обозначать $x \in M$, как $x_{z,j} \in K_1, x_{z,i} \in K_2$, где $z \in Z$. Через Z обозначается множество параметров $\{1, 2, \dots, n\}$.

Тогда задача дискриминации множества на подмножества заключается в построении такой гиперплоскости, для которой выполняется следующая система линейных неравенств:

$$\begin{cases} \sum_z b_z * x_{z,j} - c > 0 \text{ при } j \in J, \\ \sum_z b_z * x_{z,i} - c < 0 \text{ при } i \in I \end{cases} \quad (2)$$

Где b_1, b_2, \dots, b_n являются коэффициентами в линейном неравенстве, а c , соответственно, является свободным членом линейного неравенства.

Для рассматриваемого множества заемщиков данная система является несовместной, поэтому переходим к обобщенному решению системы, а именно к комитету большинства.

Введем множества "невызок"
 $V = (v_1, v_2, \dots, v_j)$ и $W = (w_1, w_2, \dots, w_i)$.

Тогда задача построения гиперплоскости сводится к задаче линейного программирования следующего вида:

$$\begin{cases} \sum_z b_z * x_{z,j} - c + v_j > 0 \text{ при } j \in J, \\ \sum_z b_z * x_{z,i} - c - w_i < 0 \text{ при } i \in I \end{cases} \quad (3)$$

$$\min \sum_j v_j + \sum_i w_i$$

Далее вводим комитетные условия. Так комитетом из q членов будем называть последовательность $Q = (t_1, t_2, \dots, t_q)$. t -ым членом комитета будем называть последовательность $t_q = (b_1, b_2, \dots, b_n)$. Введем верхний индекс t , указывающий принадлежность неравенства определённому члену комитета.

При этом для всех j, i должно быть справедливо неравенство:

$$\sum_t v_j^t + \sum_t w_i^t < \frac{q}{2} \quad (4)$$

$$V, W \in \{0,1\}.$$

То есть, сумма "невызок" для каждого элемента подмножеств K_1, K_2 должно быть меньше половины числа членов комитета.

Задача построения комитета большинства для разделения множества на подмножества сводится к задаче частично-целочисленного линейного программирования следующего вида:

$$\begin{cases} \sum_z b_z^t * x_{z,j} - c^t + L * v_j^t > 0 \text{ при } j \in J, \\ \sum_z b_z^t * x_{z,i} - c^t - L * w_i^t < 0 \text{ при } i \in I \end{cases} \quad (5)$$

$$\min \sum_t \sum_j v_j^t + \sum_t \sum_i w_i^t$$

где L есть некоторое большое число.

Данные и полученные результаты

Так у нас имеются данные об около 10 000 заявках заемщиков, имеющих 27 признаков, по которым в 61% случаев было принято решение о выдаче кредита, а в 39% случаев было отказано в выдаче кредита. Нами были построены комитеты большинства для обучающих выборок из 250, 500 и 1000 заявок. Расчеты производились в пакете IBM ILOG CPLEX Optimization Studio. Далее на экзаменующей выборке (все кредиты, не вошедшие в обучающую выборку) проверена точность комитетного решения. В таблице 1 приведены полученные результаты.

Табл. 1. Результаты, полученные при построении комитета большинства

Размер обучающей выборки	Число членов комитета	Время на построение (мин.)	Точность комитетного решения*, %
250	3	5,72	92,2
500	5	125,33	93,9
1000	9	745,45	94,1

*под точностью комитетного решения понимается доля кредитов, по которым было принято решение соответствующее решению банка, в общем числе кредитов в экзаменующей выборке

Как видно из таблицы 1 с увеличением размера обучающей выборки точность решения возрастает, однако, так же возрастает и сложность решаемой задачи. Заметим, что задача построения комитета большинства является NP-трудной [7].

Выводы

1. Нами была рассмотрена возможность применения метода комитета большинства при принятии решений по кредитным заявкам. Сформулирована задача принятия решения по кредитной заявке большинством членов комитета, которая сведена к задаче частично-целочисленного программирования.

2. Поскольку построенное комитетное решение на 94% соответствует решениям принимаемым автоматизированной системой банка, то возможно построение системы принятия решений по кредитным заявкам на основе метода комитета большинства. Исследования по данной теме представляют практический интерес для банковской сферы. В дальнейшем планируются исследования, направленные на применение метода комитетов для уменьшения кредитного риска банка.

Список источников

1. Ablow, CM. and Kaylor, D.J., Inconsistent Homogenous Linear Inequalities // Bulletin of the American Mathematical Society, 1965, vol. 71, no 5, p. 724.
2. Мазуров Вл. Д. Метод комитетов в задачах оптимизации и классификации. - М.:Наука,1990.
3. Мазуров Вл.Д., Хачай М.Ю. Комитетные конструкции / Мазуров Вл.Д., Хачай М.Ю. // Известия Уральского университета. Сер. Математика-механика. 1999. Вып. 2 (14). С. 77-109.
4. Мазуров Вл.Д., Хачай М.Ю. Комитетные конструкции как обобщение решений противоречивых задач исследования операций / Мазуров Вл.Д., Хачай М.Ю. // Дискретный анализ и исследование операций. 2003. Т.10, № 2. С.56-66.
5. Хачай М.Ю. О существовании комитета большинства // Дискретная математика. 1997. Т.9, №3. С.82-95.
6. Хачай М.Ю. Об оценке числа членов минимального комитета системы линейных неравенств // ЖВМ и МФ. 1997. Т.37, №11. С.1399-1404.
7. В. Д. Мазуров, М. Ю. Хачай, А. И. Рыбин, “Комитетные конструкции для решения задач выбора, диагностики и прогнозирования”/ В. Д. Мазуров, М. Ю. Хачай, А. И. Рыбин //Математическое программирование. Регуляризация и аппроксимация, Сборник статей, Тр. ИММ УрОРАН, 8 № 1, 2002, 66–102.
8. Кривоногов А.И. О некоторых комитетных конструкциях классификации / Кривоногов А.И. // Методы оптимизации и распознавания образов в задачах планирования. Свердловск: УНЦ АН СССР, 1980. С.92-98.

Оценка сниппетов в поиске Mail.ru: корреляция автоматических и ассессорских оценок

Андрей Кутузов

Mail.ru Group, НИУ ВШЭ, Москва, Россия. andrey.kutuzov@corp.mail.ru

Аннотация. В докладе описывается процедура автоматизированной оценки поисковых сниппетов в рамках проекта "Поиск@Mail.ru" (<http://go.mail.ru>), а также проводится сравнение результатов автоматизированных и ассессорских оценок. Анализируется возможность применения лингвистических метрик оценки сниппетов как индикаторов ошибок или регрессий в работе других компонентов большой системы веб-поиска. Кроме того, делается попытка предсказания ассессорской оценки сниппета на основе автоматизированных метрик.

Ключевые слова: поисковые сниппеты, информационный поиск, автоматическое реферирование

Введение

Постоянно усиливающийся запрос общества на информационный поиск требует от организаций, предоставляющих поисковые услуги, всё более высокого качества этих услуг. Вопросы оценки качества различных аспектов поисковой выдачи получили широкое освещение в научной литературе последних лет по тематике, связанной с компьютерной лингвистикой и computer science.

Большая часть этих работ посвящена оценке качеств поисковой выдачи с точки зрения ранжирования результатов, и вполне заслуженно: это, видимо, наиболее важная часть работы поисковой машины. Тем не

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок
менее, существуют и другие аспекты, влияющие на восприятие поискового сервиса конечным пользователем.

Среди них — поисковые сниппеты или краткие аннотации выдаваемых поисковой машиной документов. Исследования показывают, что сниппеты оказывают существенное влияние на выбор пользователем того или иного документа из выдачи. Сниппет низкого качества снижает вероятность перехода даже на релевантный запросу документ. Так, в [8] утверждается, что пользователи не переходят на 14% из высоко релевантных и 31% релевантных документов только из-за низкой оценки соответствующих сниппетов.

Субъективно «качество» сниппета воспринимается пользователем как сумма двух факторов — его «информативности» и «удобочитаемости», где «информативность» - это «индикативное качество» сниппета, его способность хорошо описать содержимое документа, а «удобочитаемость» - «перцептивное качество», лёгкость восприятия сниппета средним пользователем. [7]

[8] подтверждает, что удобочитаемость и информативность сниппета положительно коррелирует с количеством кликов на документ, представленный данным сниппетом. Поэтому для поисковых машин критически важно иметь процедуры регулярной оценки качества собственных сниппетов: как автоматизированные, работающие в реальном времени, так и основанные на данных от людей-оценщиков (ассессоров).

В данной работе мы описываем автоматизированные лингвистические метрики, применяемые для этой задачи на поисковом портале Mail.ru (<http://go.mail.ru>) и рассказываем о шаблонах их применения. Кроме того, мы сравниваем результаты автоматизированных и ассессорских оценок и делаем попытку построить модель автоматического выделения некоторой части сниппетов, которые могли бы быть оценены ассессорами как низкокачественные.

Метрики оценки сниппетов в поиске Mail.ru

Поиск@Mail.ru — это поисковая система, разрабатываемая и поддерживаемая компанией Mail.ru Group, в настоящее время является третьим по рыночной доле поисковиком в российском сегменте Интернета¹ и ежедневно обрабатывает около 40 миллионов запросов.

1 После Yandex и Google, согласно счётчикам Li.ru (<http://analyzethis.ru/?analyzer=from>)

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

Важно отметить, что по сравнению с конкурентами поисковый движок Mail.ru развивался меньшее количество времени: активная разработка и использование начались лишь в 2009 году, а собственная команда лингвистов работает только с 2012 года. Конструирование и оценка поисковых сниппетов относятся к сфере ответственности именно группы лингвистики.

До 2012 года в алгоритме формирования сниппетов лингвистические параметры практически не учитывались, углублённый анализ качества сниппетов не вёлся. Однако, в настоящее время сниппеты, выдаваемые поисковиком, систематически измеряются по 28 лингвистическим метрикам, приведённым ниже.

1. Индекс удобочитаемости. Вычисляется по формуле Флэш-Кинкэйда с весами, скорректированными для русского языка [10].
2. Количество знаков, за исключением пробелов.
3. Количество слов.
4. Количество непарных скобок (открывающая скобка не имеет соответствующей закрывающей и наоборот).
5. Количество непарных кавычек (учитываются только «ёлочки»).
6. Количество предложений. Разделение на предложения осуществляется посредством сплиттера из NLTK 2.0 [1].
7. Количество фрагментов. Границы фрагментов в сниппете обозначаются многоточием и возникают, когда высказывания, попавшие в сниппет, находятся в оригинальном документе не рядом. Например, в этом сниппете четыре фрагмента: *«31 дек 2012 ... На небе только и разговоров, что о море, и о закате...» ... закат фото. Закат на Филиппинах. Закат на ... Enniscrone Beach, Ирландия.»*.
8. Средняя длина предложения в знаках, за исключением пробелов.
9. Средняя длина предложения в словах.
10. Средняя длина фрагмента в знаках, за исключением пробелов.
11. Средняя длина фрагмента в словах.
12. Средняя длина слова.
13. Отношение количества инвектив к общему количеству слов в сниппете. Инвективы идентифицируются по списку длиной около 600 элементов (включая словоизменительные формы).
14. Лексическое разнообразие. Отношение количества словоформ в сниппете к количеству словоупотреблений.

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

15. Лексическое разнообразие выделенных слов. В нормальных условиях полужирным шрифтом выделяются в сниппете только те слова, которые присутствовали в запросе пользователя. Выделение других слов может быть признаком неисправности в генераторе сниппетов.
16. Доля слов из запроса, которые присутствуют и в сниппете. Например, если в сниппете присутствуют два из четырёх уникальных слов запроса, то этот параметр равен 0.5.
17. Отношение общего количества слов в сниппете к количеству слов из запроса в нём же.
18. Отношение общего количества слов в сниппете к количеству выделенных слов в нём же.
19. Количество неверных пробелов при знаках препинания. Сюда входят, например, отсутствующие пробелы перед скобкой, лишние пробелы перед запятой и т.д.
20. Отношение общего количества слов к количеству запятых. Если значение этого параметра приближается к единице, то немал шанс, что в качестве сниппета выдаётся список ключевых слов.
21. Отношение количества слов к количеству знаков препинания вообще.
22. Отношение количества слов, написанных полностью в верхнем регистре к общему количеству слов. При этом учитываются только слова длиннее 4 знаков, чтобы исключить наиболее распространённые аббревиатуры.
23. Отношение количества слов, в которых первый символ написан в верхнем регистре (Title Case) к общему количеству слов. Учитываются только слова от двух знаков длиной, чтобы исключить инициалы.
24. Отношение количества дат (в различных форматах) к общему количеству слов в сниппете.
25. Отношение количества цифр к общему количеству слов в сниппете.
26. Доля слов, написанных не латиницей и не кириллицей. Эта метрика ответственна за идентификацию сниппетов, содержащих символы из алфавитов, для обработки которых текущий движок поиска Mail.ru не предназначен. Возможно, в ближайшем будущем, с международным расширением поиска, она будет переработана.
27. Отношение количества слов, написанных латиницей, к общему количеству слов в сниппете.
28. Отношение количества «стоп-паттернов» к общему количеству слов в сниппете. В список «стоп-паттернов» входят регулярно

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

встречающиеся на веб-страницах строки, которые в сниппетах появляться заведомо не должны, и при этом не определяются детекторами навигационной обвязки. Это такие строки, как, например, «страница не найдена», «сейчас на сайте», «регистрация», «комментарии», «версия для печати» и т. д. Всего этот список в настоящее время насчитывает около ста позиций.

В целом, при составлении набора метрик оценки ставилась задача получить максимум параметризуемой лингвистической информации о выдаваемых поисковой системой сниппетах, не задействуя при этом глубокий морфологический и синтаксический анализ. В результате постоянного мониторинга этих данных появилась возможность выявлять узкие места алгоритма генерации сниппетов и оперативно их улучшать, а также быстро сравнивать и оценивать различные изменения в алгоритме. Не в последнюю очередь благодаря этому, уже с октября 2012 года по независимой оценке качества русскоязычных сниппетов Поиск@Mail.ru стабильно находится примерно на 1-2 пункта выше Google¹. Тем не менее, по-прежнему наблюдается отставание от Yandex на 4 пункта.

В Поиске@Mail.ru качество сниппетов контролируется отдельными тестовыми сетями по шести типам запросов:

1. Длинные запросы (длиннее четырёх слов)
2. Короткие запросы (короче трёх слов)
3. Навигационные запросы (пользователь хочет попасть на конкретную известную ему страницу)
4. Транзакционные запросы (пользователь хочет совершить некую операцию: покупку, продажу и т.д.)
5. Цитатные запросы (пользователь хочет найти источник конкретной цитаты)
6. Случайный набор запросов, включающий элементы из всех приведённых выше типов.

Отметим, что регулярный мониторинг параметров сниппетов позволяет своевременно детектировать нетипичное поведение и других компонентов системы поиска. Например, неожиданное увеличение количества инвектив в сниппетах по цитатным запросам привело в итоге к выявлению проблемы, связанной с попаданием в индекс новых документов и их ранжированием. Резкое уменьшение доли выделенных слов в сниппетах по навигационным запросам свидетельствует о пополнении базы каталожных описаний сайтов (сниппет

1 <http://analyzethis.ru/?analyzer=snippet&interval=year>

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок конструируется из каталожного описания, слова запроса не задействуются, следовательно, нет и выделенных слов). Увеличение количества непарных кавычек говорит о проблемах в модуле фильтрации навигационной обвязки (паттерны типа «*Главная » Регистрация » Новый пользователь*»), и так далее.

В дальнейшем планируется также внедрить в оценку сниппетов морфологическую и синтаксическую информацию.

Человеческая и автоматическая оценка поисковых сниппетов

Сами по себе автоматические метрики сниппетов ещё не говорят напрямую об их качестве (хотя некоторые предварительные выводы на их основании и можно сделать). Напрямую качество веб-сниппетов определяется их восприятием пользователями. Поскольку невозможно осуществить опрос всех пользователей крупной поисковой машины, наилучшее возможное приближение к этому — выборочная оценка сниппетов людьми-ассессорами.

Однако, у ассессорской оценки есть два недостатка: она дорогостояща финансово и затратна по времени [5]. Поэтому её невозможно производить в реальном времени или по отношению ко всем порождаемым сниппетам (или хотя бы относительно большей их части).

Таким образом, имеется типичная проблема автоматической классификации: небольшой массив наблюдений, точно категоризованных вручную, и большой массив, который требуется категоризовать по аналогии с первым на основании набора параметров. В [5] эта задача успешно решается для англоязычных сниппетов при помощи повышенных деревьев решений, а в [9] — для русскоязычных сниппетов при помощи составленного вручную бинарного классификатора.

В данной работе мы применяем этот же подход к массиву оцененных сниппетов Поиска@Mail.ru, обращая при этом особое внимание на точность и полноту выявления некачественных сниппетов. Основная цель обучения классификатора — это относительно надежно детектировать такие сниппеты в общем потоке и принимать необходимые меры. Детектирование качественных сниппетов в данном случае выступает в качестве вторичной задачи.

В следующем разделе будет рассказано о постановке эксперимента по обучению классификатора.

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

Постановка эксперимента

Мы исследовали оцененную ассессорами выдачу Поиска@Mail.ru по 150 случайным запросам. Выдача по каждому запросу обычно включала 10 сниппетов, которые оценивались тремя независимыми ассессорами. Таким образом, общее количество оценок составило 4260 (иногда выдача была меньше 10 позиций, либо ассессоры оценивали не все сниппеты, поэтому общее количество меньше, чем $150 \times 10 \times 3 = 4500$).

Перед ассессорами не ставилась задача отдельно оценивать информативность и удобочитаемость сниппета. Это обусловлено двумя причинами. Во-первых, эти две характеристики субъективно сложно разделить и зачастую ассессор не понимает, что именно ему/ей нужно оценивать. Во-вторых, как показано в [7], между информативностью и удобочитаемостью поисковых сниппетов существует устойчивая корреляция. Поэтому мы рассматривали их как одну интегральную характеристику «качества сниппета».

Сниппеты оценивались по трёхбалльной шкале от 0 до 2. Ассессоры были проинструктированы выставлять оценки следующим образом.

Оценка 0 выставляется, если сниппет «**плохой**» – слабо соответствует запросу. Текст представляет собой набор логически не связанных слов и словосочетаний. Наличие непонятных знаков. Многократная повторяемость выделенных слов.

Оценка 1 выставляется, если сниппет «**хороший**» – соответствует запросу, но текст тяжело читается, присутствуют слова и фразы логически не связанные с текстом запроса. Избыток выделенного текста.

Наконец, **оценка 2** выставляется если сниппет «**отличный**» – соответствует запросу и легко читаем. Текст в сниппете связный, отсутствуют непонятные символы и фразы, нет избытка выделенного текста и так далее.

В результате, как уже говорилось выше, были получены 4260 оценок для 1420 сниппетов (по три оценки на каждый). Тройная оценка была применена, чтобы минимизировать влияние субъективных склонностей конкретных ассессоров. На (Рис. 1) представлено распределение согласия среди ассессоров.

На 129 (9%) сниппетах наблюдался полный разнобой в оценках: все три ассессора выставили разное значение. Пример подобного сниппета по одному из документов выдачи по запросу «ремень для стиральной машины цена»:

Ремень для стиральной машины служит для передачи вращательного движения от электрического двигателя стиральной

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

машины к барабану. Различаются длиной, количеством клиньев (ручейков, дорожек)

По всей вероятности, один ассессор высоко оценил этот сниппет, поскольку он состоит из естественных фраз на русском языке, и при этом содержит слова из запроса. Второй ассессор посчитал, что сниппет слабо соответствует запросу, поскольку не содержит никакой информации о цене ремней. Третий же попытался совместить эти два подхода и выставил оценку «1».

Ещё для 71 сниппета (5%) наблюдалось серьёзное (на 2 пункта) отклонение оценки одного ассессора от оценки двух других. Такие расхождения могут быть обусловлены либо некачественной работой «отклоняющегося» ассессора, либо его специфическими знаниями о предмете поиска, которые позволяют ему выносить суждения, отличающиеся от прочих ассессоров.



Рисунок 1: Степень согласия ассессоров

Для большинства остальных сниппетов (896 из 1420, 63%) наблюдалось лёгкое разногласие ассессоров — например, два ассессора выставляют сниппету оценку «2», а третий - «1» Ещё для 324 сниппетов (23%) согласие ассессоров было полным, и они выставляли одинаковые оценки.

Здесь наблюдается меньшее согласие между ассессорами, чем в эксперименте для англоязычных сниппетов, описанном в [5]. Там 84.5% сниппетов были оценены при лёгком разногласии ассессоров и в 46.4% случаев наблюдалось полное согласие. Обе цифры существенно выше, чем в нашем случае. Вероятно, в первую очередь это связано с тем, что в указанном выше эксперименте больший упор делался на оценку именно удобочитаемости, и поэтому ассессоры могли частично

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок
абстрагироваться от информативности сниппетов, оценка которой зачастую более трудна и субъективна.

Сниппеты с серьёзным разногласием оценок или с тремя разными оценками мы считали «шумом», отфильтровали и не использовали в дальнейшем построении модели. Логика этого решения заключается в том, что если даже люди-ассессоры затрудняются точно определить качество подобных сниппетов, то, вероятно, объективного ответа на этот вопрос просто не существует, и такой сниппет будет «качественным» или нет в зависимости от внешних обстоятельств (например, эрудиции пользователя). Наша же задача состояла в выявлении объективно некачественных сниппетов, при оценке которых у ассессоров не возникало больших сомнений. Кроме того, как показали дальнейшие эксперименты, включение в состав модели оценок для подобных «нечётких» сниппетов приводит лишь к уменьшению её точности.

Таким образом, после фильтрации 200 сниппетов, вызвавших серьёзные разногласия ассессоров, мы получили 1220 сниппетов с тремя оценками, различающимися не более, чем на 1 пункт (всего 3660 оценок). Эти оценки были приведены к среднему значению. Распределение итоговых оценок показано на (Рис.2).

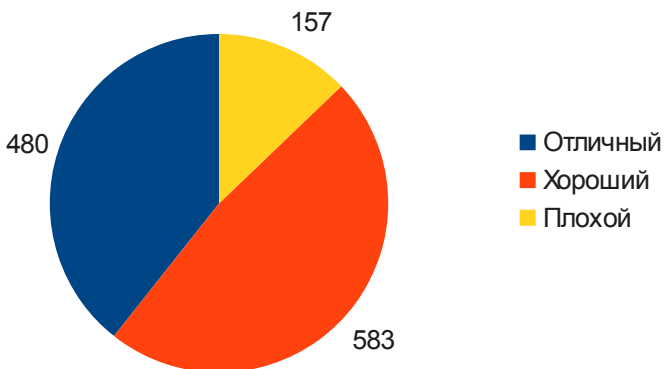


Рисунок 2: Распределение оценок

Кроме того, мы создали второй вариант этого же набора оценок с объединёнными оценками для «отличный» и «хороший». В этом случае все сниппеты делились на «качественные» и «некачественные», без дальнейшей детализации. Соответственно, в этом наборе количество «качественных» сниппетов составило $480+583=1063$, количество «некачественных» осталось 157.

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

Затем мы попытались подобрать к этим двум наборам данных алгоритм, наилучшим образом отделяющий сниппеты с плохими оценками от сниппетов с хорошими на основании приведённых выше 28 метрик. Анализ производился в ПО Weka 3.6.9. [2] Наборы оценок были предварительно случайным образом перемешаны, чтобы исключить влияние схожести сниппетов для одних и тех же запросов.

Результаты

Как уже говорилось выше, особое внимание мы уделяли точности и полноте выявления «плохих» сниппетов. Сниппетов, оцененных положительно, среди наших данных больше и они выделяются относительно надёжно, в отличие от сниппетов, оцененных отрицательно. Хорошие сниппеты хороши одинаково, плохие же плохи по-разному — так можно выразить эту дифференциацию.

Оценка эффективности работы классификаторов производилась на тестовом сете, составляющем 10% от тренировочного, то есть, 122 оценки.

Наибольшей эффективности классификации для «тернарного» набора данных (с тремя типами оценок) удалось достигнуть при использовании алгоритма Rotation Forest [6]. При этом точность выявления некачественных сниппетов составила **0.5**, полнота — **0.462**, F-мера **0.48**. Для классификатора в целом эти показатели равны **0.608**, **0.607** и **0.602** соответственно, что ещё раз подтверждает тезис о том, что некачественные сниппеты детектировать сложнее, чем качественные.

В поисках улучшения явно неудовлетворительных показателей классификатора была предпринята попытка уменьшить размерность пространства параметров путём применения алгоритма Correlation-based Feature Subset Selection [3]. Наилучшим образом коррелируют с ассессорской оценкой сниппета следующие параметры (в порядке уменьшения корреляции):

1. доля слов из запроса, присутствующих в сниппете;
2. доля выделенных слов;
3. лексическое разнообразие выделенных слов («прокрашенной» части);
4. длина фрагмента в знаках;
5. отношение количества знаков пунктуации к количеству слов;
6. индекс удобочитаемости;
7. отношение количества запятых к количеству слов;
8. количество слов с первым символом в верхнем регистре;
9. количество слов в верхнем регистре.

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

Здесь можно видеть некоторые различия по сравнению с распределением влияния параметров в эксперименте [5], где три самых влиятельных параметра — это доля больших букв, доля знаков пунктуации и доля стоп-слов, на четвертом месте длина слова, а на пятом — удобочитаемость. Количество фрагментов в сниппете — на девятом месте. Мы видим, что в нашем случае количество слов в верхнем регистре или в Title-Case также входит в 9 наиболее важных признаков, но удобочитаемость стоит по важности выше. Появление наверху нашего списка параметров, имеющих отношение к запросу, объясняется уже упоминавшимся фактом интегральной оценки нашими ассессорами информативности и удобочитаемости одновременно. Интересно, что в отличие от распределения в [5], в нашем случае более важной оказалась длина фрагментов, а не их количество: у «отличных» сниппетов средняя длина фрагмента (83 знака) значительно выше, чем у «плохих» и «хороших» (67 и 70 знаков соответственно).

Учёт только этих девяти параметров и применение гибридного алгоритма на основе таблиц решений и наивного байесовского классификатора [4] позволило повысить точность определения некачественных сниппетов с 0.5 до **0.667**, но не более. Полнота не изменилась.

Низкая эффективность классификатора связана, в том числе, с «тернарностью» классификации данных, то есть, с необходимостью различать не только «плохие» и «отличные» сниппеты, но и просто «хорошие». Результаты по второму варианту коллекции, где «хорошие» и «отличные» сниппеты были объединены в одну категорию «качественные», более обнадеживающие.

Применение к этому «бинарному» варианту радиально-базисной нейронной сети (RFBNetwork) с учётом только упомянутых выше девяти параметров позволило выявлять некачественные сниппеты с полнотой **0.667**, при этом с точностью не хуже классификатора для «тернарной» коллекции. Таким образом, F-мера выявления некачественных сниппетов также составила **0.667**. Характеристики всего классификатора в целом представлены в (Табл.1) и (Табл.2)

Таблица 1: Характеристики классификатора

Верно классифицированные сниппеты	114 (93.4426 %)
Неверно классифицированные сниппеты	8 (6.5574 %)
Каппа-коэффициент	0.6303
Средняя абсолютная погрешность	0.1695

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

Общее количество сниппетов	122
----------------------------	-----

Таблица 2: Разбивка по классам

Точность	Полнота	F-мера	Класс
0.964	0.964	0.964	Качественные
0.667	0.667	0.667	Некачественные
0.934	0.934	0.934	Средняя по всей коллекции

В [10] описан бинарный классификатор качества сниппетов с точностью 0.953, полнотой 0.718 и F-мерой 0.82. Наш классификатор характеризуется значительно более высокой полнотой при незначительной потере точности.

Обсуждение и будущая работа

Прежде всего, следует отметить, что описанный выше эксперимент проведён на ограниченном наборе данных и является пилотным. Для получения более надёжных результатов, его следует проводить на коллекциях из десятков или сотен тысяч оценок, что требует продолжительной загрузки ассессоров.

Только значительное увеличение объёма анализируемых данных позволит справиться с проблемой их разрежённости. В данном случае она выражается в том, что многие признаки обладают ненулевыми значениями лишь у небольшого количества сниппетов. Так, в нашей коллекции из 1220 наблюдений стоп-паттерны наблюдались лишь в 141 сниппете, непарные кавычки — в 58, непарные скобки - в 34, даты — в 23, а инвектив не было вообще. Между тем, интуитивно очевидно, что эти параметры влияют на восприятие сниппета пользователями, и, вероятно, на увеличенном массиве это будет заметно. Поэтому в ближайшем будущем мы планируем провести аналогичный эксперимент на объёме данных около ста тысяч сниппетов. Кроме того, планируется сравнение поведенческих характеристик различных категорий пользователей для одних и тех же сниппетов с целью найти возможную корреляцию между пользовательским поведением и лингвистическими характеристиками сниппета.

Интересным результатом нашего эксперимента стала высокая корреляция между долей слов из запроса, присутствующих в сниппете и оценкой сниппета. Визуализация этой зависимости представлена на (Рис.3). Красным цветом отмечены «плохие» сниппеты. Хорошо видно,

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

что, начиная со значения этой метрики в 66%, ассессоры уже чрезвычайно редко считают такой сниппет плохим, а при значениях выше 90% не делают этого практически никогда.

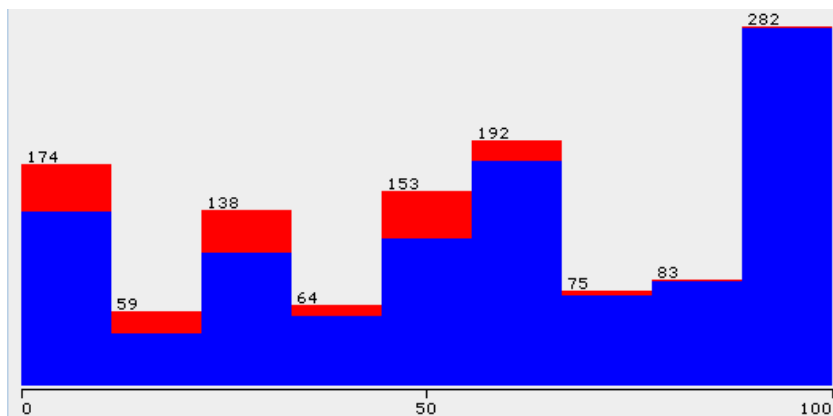


Рисунок 3: Зависимость оценки от доли слов запроса в сниппете

Так, в наших данных есть лишь 6 сниппетов (напомним, из всего 157 «плохих»), которые были отмечены как «плохие» при доле слов из запроса 75%, три «плохих» сниппета с долей слов из запроса 80 и всего один с долей 100. Вот единственный некачественный сниппет, в котором есть все слова из запроса (он выдан в ответ на запрос «смотреть бандитский петербург 2 сезон»):

*288 - Наруто 2 сезон аниме - Вы можете **посмотреть** этот кинофильм прямо сейчас в нашем кинотеатре - **Смотреть Бандитский Петербург 5 сезон сериал онлайн бесплатно - смотреть фильм онлайн***

Для такого положения вещей видятся следующие основания. Вероятно, редко встречаются сниппеты, где есть все слова из запроса, но при этом они разбросаны среди плохо читаемых и мало информативных фрагментов. Типичный сниппет с параметром «доля слов из запроса», равным 100%, оценивается ассессорами как качественный и выглядит так (запрос «жалюзи шторы на пластиковые окна»):

***Шторы и жалюзи на пластиковые окна** - Многие, кому приходилось задумываться о покупке **жалюзи** для **окон**, знают, как порой нелегко найти именно тот вариант, который необходим.*

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

То есть, он представляет из себя набор осмысленных фраз, а это улучшает субъективное восприятие пользователя. Между тем, сниппет с низкой долей слов из запроса зачастую воспринимается, как ведущий на нерелевантную запросу страницу, и, как следствие, оценивается низко. Таким образом, высокая предиктивность параметра «доля слов из запроса в сниппете» при предсказании ассессорской оценки не случайна.

Этот и прочие параметры с высокой корреляцией, перечисленные выше, могут быть использованы при обработке потока сниппетов поисковой выдачи в реальном времени или оффлайн. Найденные потенциально «плохие» сниппеты необходимо анализировать для выявления ошибок и потенциальных улучшений в алгоритме их формирования, а также типичных паттернов, ухудшающих восприятие сниппетов пользователями.

Безусловно, следует продолжать и поиск дополнительных параметризуемых характеристик сниппетов, которые могут иметь значение для предсказания их оценки. Вместе с тем, нужно отметить, что «качество» поискового сниппета зачастую может оценить лишь человек, привлекая весь присущий ему аппарат глубинного семантического анализа. Например, сниппет

Кыргызстан может потерять до 70 процентов... Хотя подобные озера в Кыргызстане и единичны, их изучение представляет важный научный и практический интерес. К числу таких озер можно отнести и озеро...,

выданный по запросу «**исчезнувшие озера в кыргызстане**», представляет собой вполне связный и легко читаемый текст, содержит 75% слов из запроса, и с точки зрения любых чисто статистических параметров должен считаться «качественным». Но ассессоры единодушно выставили ему оценку «0» — по всей вероятности, из-за отсутствия слова «исчезнувшие», которое они справедливо посчитали ключевым в запросе. Без применения глубокого семантического анализа учитывать подобные свойства сниппетов невозможно. Поэтому до момента внедрения таких технологий в процесс выявления некачественных сниппетов, определённая их доля всегда будет оставаться «невидимой» для автоматизированных процедур оценки.

Выводы

1. Описаны лингвистические метрики сниппетов, применяющиеся в Поиске@Mail.ru

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

2. Подтверждена возможность создания модели автоматизированной оценки качества сниппетов на основе лингвистических метрик и обучения на ассессорских оценках
3. Определены параметры, наиболее пригодные для выделения некачественных сниппетов
4. Применение радиально-базисной нейронной сети (RFBNetwork) позволяет выделять некачественные сниппеты с точностью и полнотой 0.667, при F-мере всего классификатора в целом 0.934 (хорошие сниппеты выделять проще, чем плохие).
5. Наблюдается высокая корреляция между долей слов из запроса в сниппете и его оценкой ассессорами.
6. Для построения более надёжного классификатора необходим набор на порядок большего количества ассессорских оценок по сниппетам, что будет осуществлено в ближайшем будущем.

Благодарности

Автор благодарен Игорю Андрееву и Максиму Ионову (группа прикладной лингвистики Поиска@Mail.ru) за ценные замечания, высказанные в ходе подготовки работы.

Список источников

1. Steven Bird, Ewan Klein, Edward Loper. Natural Language Processing with Python. O'Reilly Media Inc. 2009
2. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. 2009
3. M. Hall. Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand. 1998
4. M. Hall, E. Frank. Combining Naive Bayes and Decision Tables. In: Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), 318-319. 2008.
5. Taras Kanungo, David Orr. Predicting the readability of short web summaries. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 202-211, ACM New York, NY, USA, 2009

Оценка сниппетов в Поиске@Mail.ru: корреляция автоматических и ассессорских оценок

6. J. Rodriguez, L. Kuncheva, C. Alonso. Rotation Forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28(10):1619-1630. 2006
7. Savenkov D., Braslavski P., Lebedev M. Search Snippet Evaluation at Yandex: Lessons Learned and Future Directions. In: CLEF 2011.
8. A. Turpin, F. Scholer, K. Jarvelin, M. Wu, J. Culpepper. Including summaries in system evaluation. In: SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
9. Киселев Ю.А. Улучшение читаемости сниппетов поисковых систем. // *Scientific research and their practical applications. Modern state and ways of development*, 2012
10. Оборнева И.В. Математическая модель оценки учебных текстов // *Вестник МГПУ. Серия «Информатика и информатизация образования»*. – М.: МГПУ, 2005. No1 (4). – С.141-147.

Разработка системы видеодетектирования транспортных средств

Валентина Кустикова¹, Николай Золотых², Евгений Козин³,
Юсиф Мееров⁴, Алексей Половинкин⁵

¹Нижегородский государственный университет им. Н.И. Лобачевского,
Н. Новгород, Россия. valentina.kustikova@gmail.com

²Нижегородский государственный университет им. Н.И. Лобачевского,
Н. Новгород, Россия. nikolai.zolotykh@gmail.com

³Нижегородский государственный университет им. Н.И. Лобачевского,
Н. Новгород, Россия. evgeniy.kozinov@gmail.com

⁴Нижегородский государственный университет им. Н.И. Лобачевского,
Н. Новгород, Россия. meegov@vmk.unn.ru

⁵Нижегородский государственный университет им. Н.И. Лобачевского,
Н. Новгород, Россия. alexey.polovinkin@gmail.com

Аннотация. Ставится задача поиска транспортных средств на потоке видеоданных. Предлагается общая схема решения задачи, основанная на детектировании и последующем сопровождении объектов. Описывается прототип разработанной системы. Проводится анализ текущих результатов, а также формулируются дальнейшие направления исследований.

Ключевые слова: компьютерное зрение; машинное обучение; поиск объектов на изображении; сопровождение объектов.

Введение

В связи с ростом транспортной нагрузки остро встают вопросы планирования строительства дорожных развязок и эффективного распре-

ления транспортных потоков с целью обеспечения бесперебойного движения на отдельных дорожных участках и магистралях.

Видеодетектирование – один из возможных подходов к решению проблемы анализа транспортных потоков. Задача видеодетектирования состоит в том, чтобы определить положение транспортных средств (ТС) на каждом кадре потока данных и построить траекторию движения обнаруженных объектов. Построение траектории подразумевает выделение на наборе последовательно идущих кадров совокупности положений, отвечающих каждому объекту, который попадает в зону видимости камеры.

Обзор научных работ последних лет показал, что задача видеодетектирования ТС изучается многими исследователями [1 – 4]. Основная цель рассмотренных публикаций – повышение качества поиска ТС за счет модификации алгоритмов компьютерного зрения. Однако подавляющее большинство работ посвящено детектированию транспорта посредством обнаружения регистрационных номеров, т.к. данный объект с точки зрения алгоритмов распознавания является наиболее простым (контрастность фона и символов, конечность множества символов для распознавания). В результате возникают ограничения, связанные с расположением камеры и ее разрешением.

В настоящей работе предлагается использовать подход, основанный на идеях, предложенных в [5]. Данный подход предполагает поиск ТС как самостоятельных объектов. С одной стороны, задача поиска транспортных единиц, а не их отдельных частей, является алгоритмически более сложной (требуется инвариантность относительно ракурса и масштаба объекта, устойчивость к частичным перекрытиям), но с другой – ее решение позволит увеличить объем информации, которая извлекается из потока видеоданных, и смягчить требование фиксированного расположения камеры относительно дороги.

Постановка задачи

Метод видеодетектирования ТС работает с потоком видеоданных V , который можно представить в виде последовательности изображений $I_0, I_1, I_2, \dots, I_{N-1}$, где N – количество кадров видео. *Положение ТС* определяется расположением прямоугольника, его окаймляющего [5, 6]. Задача видеодетектирования ТС состоит в том, чтобы каждому кадру I_k исходного видео V поставить в соответствие совокупность положений объектов B_k . Задача сводится к построению отображения φ (1).

$$\varphi: \{I_k, k = \overline{0, N-1}\} \rightarrow \{B_k, k = \overline{0, N-1}\}, \quad (1)$$

где $B_k = \{b_l^k, l = \overline{0, s_k - 1}\}$ – множество окаймляющих прямоугольников, обнаруженных на кадре I_k . При этом каждый прямоугольник b_l^k определяется набором следующих компонент:

$$b_l^k = \left((x_1^l, y_1^l), (x_2^l, y_2^l) [s^l, c^l] \right), \quad (2)$$

где $(x_1^l, y_1^l), (x_2^l, y_2^l)$ – координаты левого верхнего и правого нижнего углов прямоугольника, $s^l \in \mathbb{R}$ – достоверность того, что объект обнаружен правильно, а c^l – класс, которому принадлежит ТС (автомобиль, автобус, трамвай и др.).

Метод решения задачи видеодетектирования

Подробный обзор существующих методов видеодетектирования ТС на потоке видеоданных был приведен в работе [7]. В целом можно выделить два основных подхода к решению задачи:

1. Поиск и сопровождение областей движения с последующей идентификацией ТС внутри полученного набора областей [12].

2. Поиск ТС и их последующее сопровождение [5].

Принципиальное отличие приведенных подходов состоит в том, что согласно первому подходу распознавание ТС происходит в момент, когда имеется полный набор областей движения, а согласно второму – в момент появления ТС в области обзора камеры.

В настоящее время большинство систем видеодетектирования использует схему, основанную на поиске и сопровождении областей движения. Отметим, что такая схема работает лишь в случае неподвижной камеры и относительно постоянного фона, что в естественных условиях получить достаточно сложно.

В данной работе предлагается использовать второй подход. Схема решения задачи видеодетектирования ТС в соответствии с данным подходом включает несколько действий:

1. Выделение кадра видео.

2. Определение положения ТС на текущем кадре и оценка достоверности нахождения ТС в полученной области – *детектирование*. На текущем этапе развития проекта детектирование выполняется посредством алгоритма Latent SVM [6, 8].

3. *Сопровождение* (трекинг) выделенных ТС на следующих кадрах последовательности. Поскольку в реальных системах видео зачастую имеет небольшую характеристику FPS (~5 кадров/с), то алгоритмы сопровождения, основанные на вычислении оптического потока, не применимы. В подобных условиях предлагается использовать алгоритмы сопоставления (matching) особых точек [7].

4. Анализ результатов (определение направления движения ТС, подсчет ТС, классификация ТС и т.п.).

Архитектура системы

Модули системы можно разделить на 5 групп (рис. 1):

1. *Модули подготовки тестовых данных*: модуль разметки окаймляющих прямоугольников; модуль формирования траекторий движения.

2. *Модули детектирования и сопровождения*:

- Модуль детектирования ТС на отдельном кадре с помощью параллельной реализации алгоритма Latent SVM [9]. Классификатор натренирован на данных PASCAL Visual Object Classes Challenge 2007 (VOC 2007, [10]), модель состоит из двух компонент.

- Модуль сопровождения ТС (алгоритм Лукаса-Канаде, сопоставление ключевых точек).

- Приложения, содержащие реализации различных схем видеодетектирования ТС.

3. *Модули апробации результатов видеодетектирования транспортных средств*: модуль вычисления показателей качества детектирования и сопровождения ТС; сторонний модуль VOCdevkit [11].

4. *Служебные модули визуализации*.

5. *Служебные модули для автоматизации сбора показателей качества видеодетектирования*.



Рис. 1. Архитектура системы видеодетектирования

Приведенные модули разработаны на базе библиотеки OpenCV [12].

Текущие результаты

Тестовые данные

Для проведения экспериментов по оцениванию качества видеодетектирования транспортных средств было снято несколько треков (25FPS, 720x405 пикселей), из которых выбраны отдельные последовательности кадров:

- *track_09_0-2000* (2000 кадров = 80 секунд) содержит большое количество мелких транспортных средств, движущихся в двух противоположных направлениях (рис. 2, слева);
- *track_10_5000-7000* (2000 кадров = 80 секунд) – видео с транспортными средствами, которые двигаются в 4 полосы одного направления (рис. 2, справа);
- *track_10_7000-8000* (1000 кадров = 40 секунд) – видео, снятое при тех же условиях, что и *track_10_5000-7000*.

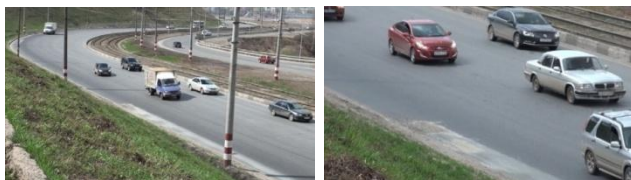


Рис. 2. Кадры треков *track_09_0-2000*, *track_10_5000-7000*

Критерии оценки качества видеодетектирования

В работах [5, 13] используются количественные показатели, которые вычисляются на основании информации о числе правильно и неправильно продетектированных объектов:

1. *Истинноположительный показатель* (the true positive rate [5] или detection rate [13], *TPR*) – отношение количества правильно продетектированных транспортных средств к общему числу транспортных средств.

2. *Показатель числа ложных срабатываний* (the false detection rate, *FDR*) – отношение количества ложных срабатываний к общему числу срабатываний детектора. При этом считается, что объект продетектирован правильно, если перекрытие продетектированного и размеченного окаймляющего прямоугольника превышает некоторое пороговое значение (в [13] в качестве порога выбирается 80%).

Наряду с указанными метриками авторы [5] вводят несколько относительных показателей качества:

1. *Количество ложных срабатываний, которое в среднем приходится на один кадр* (the average false positives per frame, *FPperFrame*) – отношение количества ложных срабатываний детектора к общему числу кадров видео.

Разработка системы видеодетектирования транспортных средств

2. Среднее количество ложных срабатываний, приходящееся на объект (the average false positives per object, $FPperObject$) – отношение количества ложных срабатываний детектора к общему числу объектов, содержащихся на видео.

Результаты экспериментов

Результаты видеодетектирования с помощью разработанных схем ниже сведены в общей таблице (табл. 1):

- 1 – схема полного покадрового детектирования.
- 2 – схема полного детектирования с отсечением прямоугольников, находящихся вне областей интереса (области интереса задаются бинарным изображением, полученным в результате разметки одного кадра видео (рис.1)).
- 3 – схема детектирования с одношаговым трекингом и отсечением. Идея применения трекинга на один кадр вперед состоит в том, чтобы обеспечить дополнительный уровень проверки корректности работы детектора с целью отсечения ложных срабатываний.
- 4 – схема детектирования с одношаговым трекингом без отсечения.
- 5 – схема детектирования с трекингом на 5 кадров вперед и отсечением.
- 6 – схема детектирования с трекингом на 5 кадров вперед без отсечения.
- 7 – схема детектирования с трекингом на 3 кадра вперед и отсечением.

Табл. 1. Сводные результаты видеодетектирования

№ схемы	Название тестового видео	Значение показателя TPR		Значение показателя FDR		Значение показателя $FPperFrame$		Значение показателя $FPperObject$	
		BUS	CAR	BUS	CAR	BUS	CAR	BUS	CAR
1	track_09_0-2000	0.291	0.201	0.924	0.786	1.627	5.898	3.530	0.738
	track_10_5000-7000	-	0.728	-	0.735	-	2.942	-	2.015
	track_10_7000-8000	0.657	0.745	0.910	0.813	0.660	3.567	6.677	3.243
2	track_09_0-2000	0.287	0.114	0.802	0.663	0.532	1.792	1.168	0.224
	track_10_5000-7000	-	0.726	-	0.676	-	2.209	-	1.513
	track_10_7000-8000	0.657	0.745	0.905	0.784	0.619	2.980	6.263	2.709
3	track_10_5000-7000	-	0.680	-	0.437	-	0.772	-	0.529
	track_10_7000-8000	0.596	0.697	0.879	0.631	0.427	1.313	4.313	1.193
	track_10_5000-7000	-	0.683	-	0.518	-	1.074	-	0.735
4	track_10_7000-8000	0.596	0.697	0.884	0.674	0.448	1.583	4.525	1.440
	track_10_5000-7000	-	0.586	-	0.508	-	0.885	-	0.606
	track_10_7000-8000	0.606	0.596	0.869	0.667	0.340	1.134	4.04	1.194
5	track_10_5000-7000	-	0.622	-	0.605	-	1.390	-	0.952
	track_10_7000-8000	0.606	0.621	0.879	0.721	0.436	1.770	4.404	1.609
	track_10_5000-7000	-	0.644	-	0.521	-	1.023	-	0.700
7	track_10_7000-8000	0.606	0.667	0.875	0.659	0.419	1.416	4.242	1.287

Количественные показатели качества (столбцы 3 – 6) приведены для порогового значения процента пересечения размеченных и протектированных окаймляющих прямоугольников, равного 50%.

Введение отсечения по областям интереса позволило незначительно снизить количество ложных срабатываний детектора (в среднем на одно

срабатывание на каждом кадре, столбец 5). Применение алгоритмов сопровождения обеспечило снижение числа ложных срабатываний. При этом очевидно прослеживается уменьшение правильно протектированных объектов. Если для реализаций, использующих одношаговый трекинг (схемы 3 и 4), это сказывается в сотых долях, то с многошаговым сопровождением (схемы 5, 6, 7) падение точности наблюдается с увеличением шага для выполнения очередного детектирования. Причина состоит в том, что при увеличении шага увеличивается вероятность перекрытия транспортных средств другими объектами или частичный выход из кадра. Как следствие, алгоритм детектирования может не обнаружить данный объект, и ошибка распространится на несколько предыдущих кадров.

Заключение

В работе представлен прототип системы видеодетектирования транспортных средств, обладающей следующей функциональностью:

- 1. Поиск ТС на отдельных кадрах видео.*
- 2. Классификация ТС (автомобили, автобусы).*
- 3. Сопровождение обнаруженных ТС.*

Анализ работы прототипа на модельных задачах показал, что среднее время обработки кадра (720×405) составляет 4с, для наиболее перспективной схемы количество правильно обнаруженных ТС примерно составляет 68% от общего числа ТС при одном ложном срабатывании на кадр.

Дальнейшее развитие системы планируется вести в следующих направлениях:

- 1. Повышение скорости обработки данных за счет применения каскадной модификации алгоритма Latent SVM (время обработки кадра ~1с).*
- 2. Повышение качества поиска транспортных средств вследствие модификации имеющейся реализации алгоритма поиска: выбор новых признаков, адаптация других методов машинного обучения. Применение методов сопоставления вместо сопровождения.*
- 3. Нарращивание функциональности: более «мелкая» классификация ТС, построение траекторий движения ТС, подсчет ТС с учетом направления движения.*

СПИСОК ИСТОЧНИКОВ

1. R. Rad, M. Jamzad Real time classification and tracking of multiple vehicles in highways // Journal Pattern Recognition Letters. 2005. V. 26. Issue 10. P. 1597-1607.
2. M. Kafai, B. Bhanu Dynamic Bayesian Networks for Vehicle Classification in Video // IEEE Transactions on Industrial Informatics. 2012. V.8. No.1. P.100-109.
3. N. Buch, J. Orwell, S. A. Velastin A Review of Computer Vision Techniques for the Analysis of Urban Traffic // IEEE Transactions on intelligent transportation systems. 2011. V.12. No.3. P. 920-939.
4. S. Gauglitz, T. Holerer, M. Turk Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking [<http://cs.iupui.edu/~tuceryan/pdf-repository/Gauglitz2011.pdf>].
5. S. Sivaraman, M.M. Trivedi A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking //IEEE Transactions on Intelligent Transportation Systems. 2010. V.11. No.2. P. 267-276.
6. P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan Object Detection with Discriminatively Trained Part Based Models // Transactions on Pattern Analysis and Machine Intelligence. 2010. V.32. No.9. P. 1627–1645.
7. Н.Ю. Золотых, В.Д. Кустикова, И.Б. Мееров Обзор методов поиска и сопровождения транспортных средств на потоке видеоданных // Н.Новгород: Вестник ННГУ им. Н.И. Лобачевского. 2012. № 5(2). С. 346-357.
8. P. F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan Cascade object detection with deformable path model // Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'10). 2010. P. 2241-2248.
9. P.N. Druzhkov, V.L. Eruhimov, E.A. Kozinov, V.D. Kustikova, I.B. Meyerov, A.N. Polovinkin, N.Yu. Zolotykh. On some new object detection features in OpenCV library // Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications. 2011. V.21. No.3. P. 384-386.
10. The PASCAL Visual Object Classes Challenge 2007 [<http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/index.html>].
11. PASCAL Visual Object Challenge Development Kit [<http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/#devkit>].
12. Официальная страница библиотеки OpenCV [<http://opencv.org/>].
13. Song X., Netavia R. A Model-based Vehicle Segmentation Method for Tracking // Proceedings of the IEEE ICCV. 2005. V.2. P. 1124-1131.

Фильтрация ложных соответствий описателей особых точек изображений

Сергей Белоусов¹, Александр Шишков²

¹ННГУ им. Н.И. Лобачевского, Нижний Новгород, Россия. belbes122@yandex.ru

²Itseez, Нижний Новгород, Россия. shishkov.alexander@gmail.com

Аннотация. Статья посвящена алгоритму фильтрации ложных соответствий между описателями особых точек изображений на основе эвристических методов. В работе рассматриваются различные существующие подходы к решению этой задачи. Приводится вариант объединения нескольких методов, дающий более качественные результаты, нежели каждый из них по отдельности.

Ключевые слова: дескриптор; описатель; особый регион; особая точка; соответствие; фильтрация.

Введение

Многие современные методы распознавания изображений строятся на анализе соответствий особых точек визуального объекта, которые описываются посредством дескрипторов или описателей [1]. Особая точка p состоит из двух частей: $p = (p_d, p_c)$, где $p_d \in R^n$ – вектор вещественных чисел размерности n , представляющий собой дескриптор особой точки (например, SIFT-дескриптор [1]); а $p_c = (x, y)$ содержит пространственные координаты особой точки в плоскости изображения. Применение особых точек позволяет решать задачу распознавания в условиях неполной информации об анализируемых объектах и устранять ложные элементы описания.

Пусть U – некоторый универсум особых точек $p_i \in U$. Обозначим за P^1, P^2 конечные множества особых точек $P^1, P^2 \subset U$, $P^1 = \{p_i^1\}$, $P^2 = \{p_k^2\}$, $\|P^1\| = \mu_1$, $\|P^2\| = \mu_2$. Пусть $\Omega: P^1 \rightarrow P^2$ некоторое отображение из множества P^1 в множество P^2 . В результате применения Ω получим конечное множество $\Theta = \{\theta_j\}$ ($\theta \in \Theta$, $\|\Theta\| = m$, $m \leq \mu_1 * \mu_2$) парных соответствий $p_i^1 \rightarrow p_k^2$, $p_i^1 \in P^1$, $p_k^2 \in P^2$ (Θ – универсум соответствий).

Одним из наиболее распространенных методов построения множества Θ является вычисление дистанции $d(p_{id}^1, p_{kd}^2)$ между особыми точками из U , например, евклидоваго расстояния между дескрипторами. В задачах распознавания визуальных объектов принято использовать одно из множеств P^1 как эталонное и искать для каждого элемента из этого множества одно или несколько ближайших соответствий в множестве P^2 .

Переход к анализу особых точек существенно сокращает объем информации о визуальных объектах, часто обеспечивая при этом достаточно высокий уровень надежности при распознавании, так как посредством особых точек удастся сохранить важную для распознавания информацию. Однако, в связи с потерей информации часто возникают ложные соответствия, что приводит к неправильным результатам при распознавании изображений. Основная причина состоит в том, что подход, состоящий в построении соответствий как наиболее близких между собой точек, не отображает физических и пространственных свойств визуальных объектов. Так, например, даже когда на сцене нет эталонного объекта, мы все равно получаем некоторое количество соответствий (рис. 1).

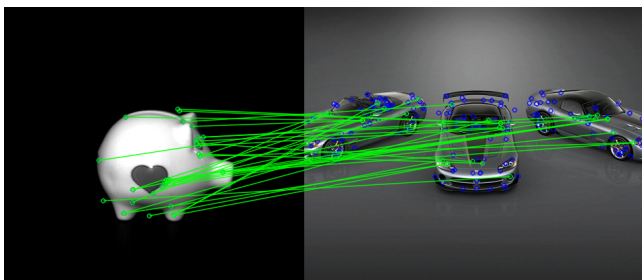


Рис. 1. Ложные соответствия

В данной работе предложен метод фильтрации ложных соответствий между особыми точками объекта и сцены, основанный на эвристических предположениях относительно свойств особых точек. Результаты работы данного алгоритма представлены на рисунке 2. Как видно

из этой иллюстрации, метод отфильтровал все ложные соответствия между изображениями.

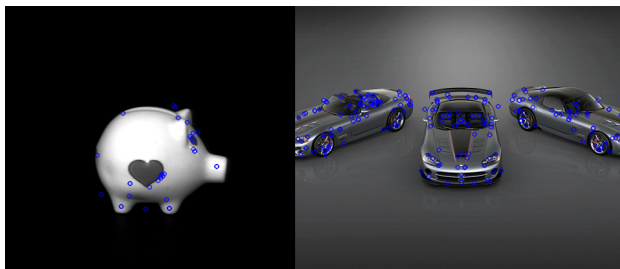


Рис. 2. Пример работы предложенного метода фильтрации

Фильтрация соответствий

В общем виде фильтрацию соответствий можно определить как отображение $F: \Theta \rightarrow \Theta$, которое для конкретного набора соответствий имеет вид $F: \vartheta \rightarrow \vartheta^f$, где элементы подмножества $\vartheta^f \subseteq \vartheta$ обладают заданными свойствами, причем $\vartheta, \vartheta^f \in \Theta$. Применительно к методам предлагаемым в данной работе процесс фильтрации соответствий представляет собой выбор только тех соответствий, которые удовлетворяют эвристическим предположениям о свойствах правильных соответствий.

Предлагаемый в данной работе метод фильтрации ложных соответствий представляет собой последовательное применение эвристических предположений относительно свойств особых точек и пар соответствий. Далее приводится описание используемых в данном исследовании эвристик в том порядке, в котором их последовательное применение дает наилучшее соотношение числа правильных соответствий к ложным соответствиям до фильтрации и после.

В целях качественного анализа предложенного в статье подхода проведены компьютерные эксперименты на тренировочном наборе входных данных, взятых из библиотеки OpenCV [3]. В качестве детектора особых точек использовался детектор SIFT [1], а в качестве описателя особой точки использовался RootSIFT [4]. Соответствия были построены на основе евклидова расстояния между дескрипторами особых точек.

Сравнение различных алгоритмов фильтрации соответствий было проведено в пространстве *recall/precision*. Где $recall = \frac{tp}{tp+fn}$, $precision = \frac{tp}{tp+fp}$: tp (true positive) – количество корректно определенных правильных соответствий; fp (false positive) – количество ложных

соответствий, определенных как правильные; fn (false negative) – количество отсеянных правильных соответствий. Иными словами, *precision* характеризует долю правильных соответствий, а *recall* характеризует долю оставшихся правильных соответствий от их общего числа после применения предложенного в статье алгоритма.

Фильтрация на основе соотношения дистанций

Для каждой особой точки рассмотрим два ближайших по дистанции описателя из другого множества особых точек. Если расстояние до первого соответствия значительно меньше чем до второго, предполагаем что такое соответствие правильное [1].

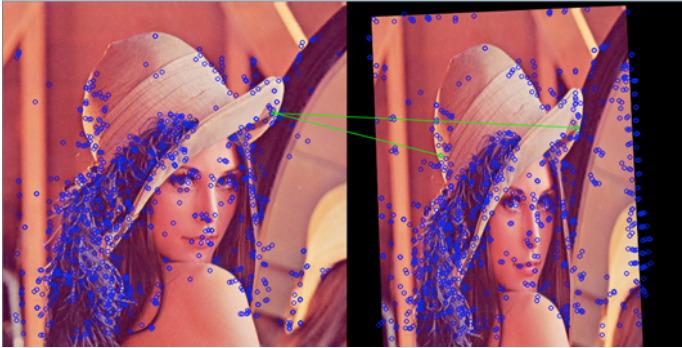


Рис. 3. Пример соответствия, удовлетворяющего данной эвристике

Другими словами, множество ϑ^f выбираем как $\vartheta^f = \left\{ \vartheta_j: p_i^1 \rightarrow p_{k_1}^2 \in \vartheta, \frac{d(p_i^1, p_{k_1}^2)}{d(p_i^1, p_{k_2}^2)} < r \right\}$, где r – пороговый коэффициент.

Результаты применения этой фильтрации для оптимального значения порога представлены на рисунке 4. Каждой точке на графике соответствует один из наборов тестовых входных данных.

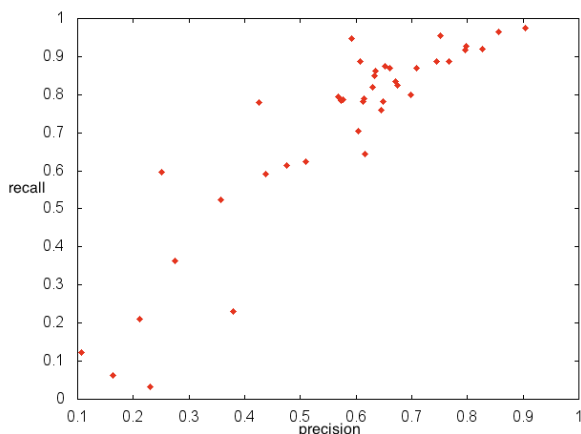


Рис. 4. Результаты работы фильтрации на основе соотношения дистанций

Обратное отображение

Строим вспомогательное отображение $\bar{\Omega}: P^2 \rightarrow P^1$, ставящее в соответствие каждой особой точке сцены точку эталона. Предполагается, что если существует соответствие для точки p_i^1 вида $\vartheta_j: p_i^1 \rightarrow p_k^2$, где $\vartheta_j \in \vartheta$, тогда если существует такое соответствие $\bar{\vartheta}_q: p_k^2 \rightarrow p_i^1$, где $\bar{\vartheta}_q \in \bar{\vartheta}$, то считаем соответствие ϑ_j правильным. На следующих рисунках изображен пример соответствия, удовлетворяющего данному правилу, и соответствующее ему обратное отображение.

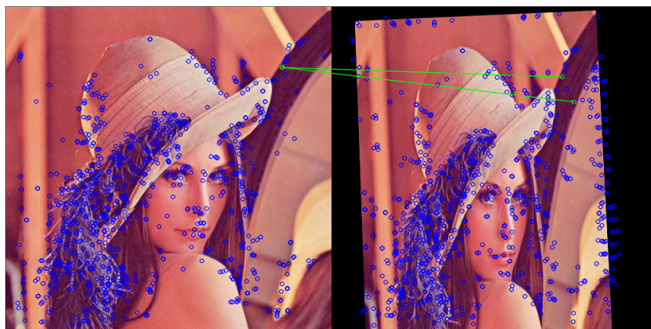


Рис. 5. Пример соответствия $\vartheta_j: p_i^1 \rightarrow p_k^2$, удовлетворяющего обратному отображению

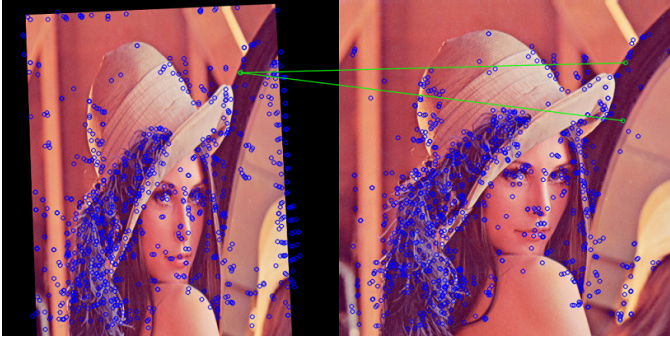


Рис. 6. Соответствие $\bar{\vartheta}_q: p_k^2 \rightarrow p_i^1$ обратного отображения

Формально множество ϑ^f выбираем как $\vartheta^f = \{\vartheta_j: p_i^1 \rightarrow p_k^2 \in \vartheta, p_i^1 \in \bar{\Omega}(\Omega(p_i^1))\}$.

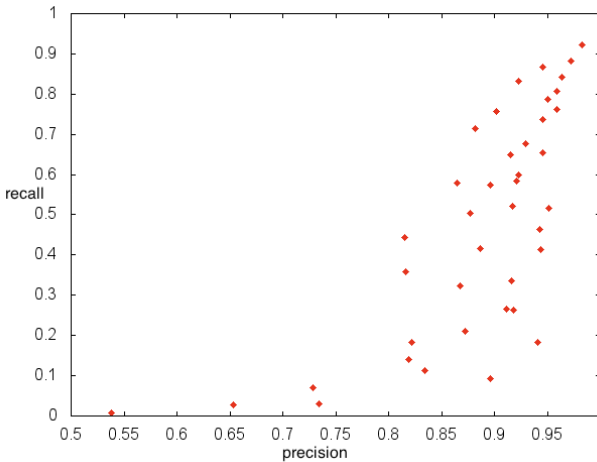


Рис. 7. Результаты работы фильтрации на основе обратного отображения

Фильтрация узловых соответствий

Предположим, что есть несколько точек из P^1 , которым соответствует один и тот же образ из P^2 (таким образом эти соответствия образуют в некотором смысле «узел»), тогда все эти точки не являются достаточно уникальными, и их соответствия не рассматриваются как правильные. Однако такие соответствия могут возникать в результате сла-

Фильтрация ложных соответствий описателей...

бой текстурности изображения объекта, и в таком случае «узел» может содержать в себе правильные соответствия. Поэтому из каждого «узла» выбирается то соответствие, для которого мера различия между эталоном и образом минимальна и удовлетворяет некоторому пороговому значению, а остальные соответствия из «узла» считаются ложными. Другими словами, множество ϑ^f выбирается как

$$\begin{aligned} \vartheta^f &= A \cup B \\ A &= \{\vartheta_j \in \vartheta : \forall p_k^2, \exists ! p_l^1 : \Omega(p_l^1) = p_k^2\} \\ B &= \{\vartheta_i : p_c^1 \rightarrow p_d^2 \in \vartheta : \forall \vartheta_o : p_c^1 \rightarrow p_d^2 : d(p_c^1, p_d^2) \leq d(p_o^1, p_o^2), d(p_c^1, p_d^2) \leq \text{distance}\} \end{aligned} \quad (1)$$

, где *distance* – пороговое значение.

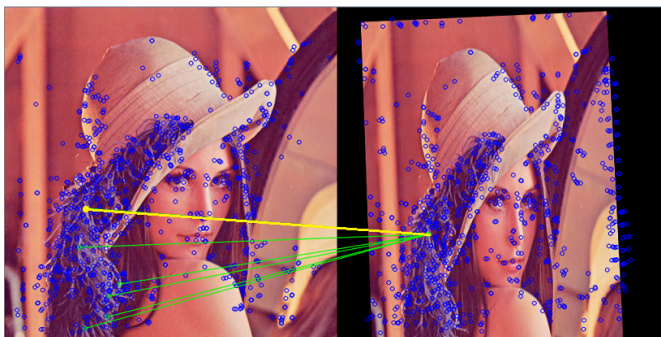


Рис. 8. Пример "узла", содержащего удовлетворяющее пороговому значению соответствие

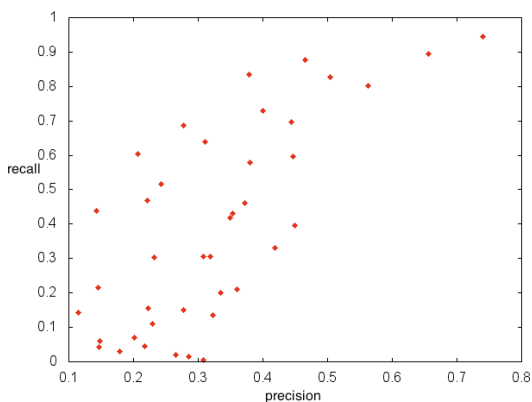


Рис. 9. Результаты работы фильтрации узловых соответствий

Геометрическая фильтрация

Предположим, что две особые точки лежат в некоторой малой окрестности друг от друга, тогда, если эти соответствия являются верными, образы этих точек должны также располагаться в некоторой окрестности друг от друга. Для проверки этой гипотезы необходимо для каждой особой точки взять некоторую окрестность (например, масштаб в котором точка является особой). Возьмем некоторое соответствие $\vartheta_j: p_i^1 \rightarrow p_k^2$, будем считать его правильным в том случае, если существует такое соответствие $\vartheta_t: p_c^1 \rightarrow p_d^2$, что окрестности точек p_i^1 и p_c^1 , для которых они являются особыми, пересекаются, а также пересекаются окрестности соответствующих им образов p_k^2 и p_d^2 .

Другими словами, множество ϑ^f выбираем как $\vartheta^f = \{\vartheta_j: p_i^1 \rightarrow p_k^2, \exists \vartheta_t: p_c^1 \rightarrow p_d^2: \text{overlap}(p_i^1, p_c^1) \rightarrow \text{overlap}(p_k^2, p_d^2)\}$, где $\text{overlap} = \text{true}$, если окрестности точек пересекаются, иначе $\text{overlap} = \text{false}$.

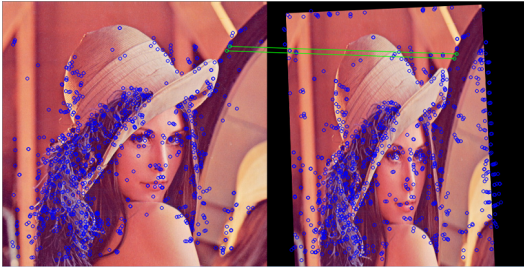


Рис. 10. Пример пары соответствий с перекрывающимися окрестностями

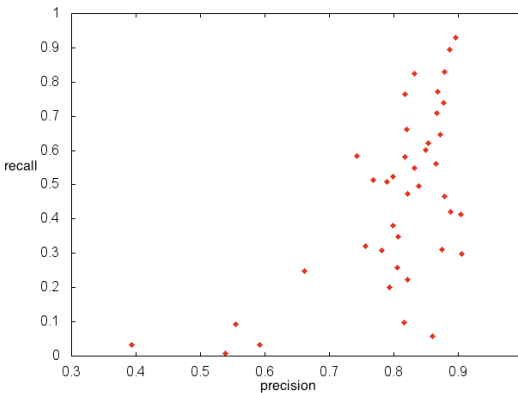


Рис. 11. Результаты работы геометрической фильтрации

Объединенный алгоритм

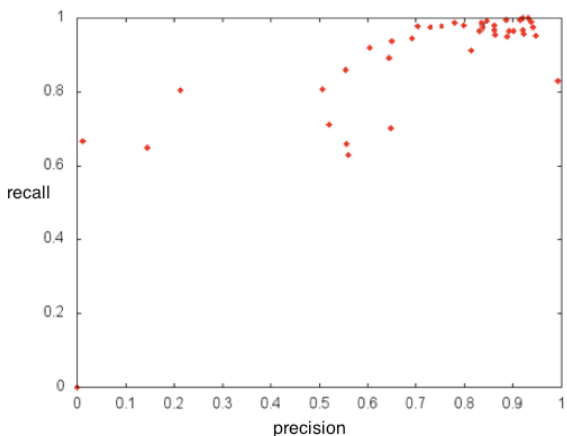


Рис. 12. Результаты работы алгоритма на тестовом наборе входных данных

Как видно из приведенного графика, для большинства входных наборов данных, на выходе получается набор, содержащий более 80% правильных соответствий, при этом доля *false negative* не превышает 50%.

Точкам для которых $recall < 0.7$ или $precision < 0.8$ соответствуют случаи, для которых в исходных данных доля ложных соответствий превышает 95%.

Типичный пример набора соответствий, полученного в результате фильтрации предложенным в данной статье методом, изображен на рис. 13.

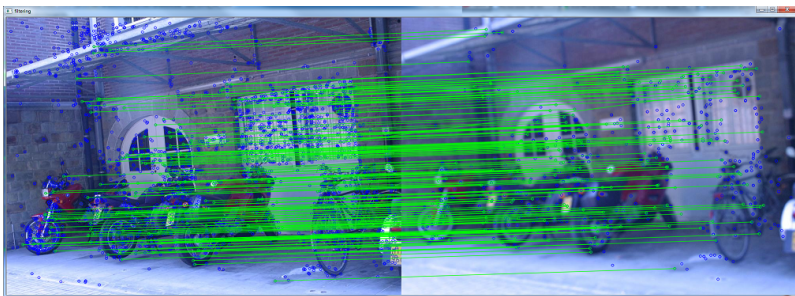


Рис. 13. Соответствия особых точек после фильтрации

Выводы

- 1. Фильтрация соответствий особых точек, полученных при сопоставлении изображений, позволяет улучшить достоверность установления их связи, и как результат – увеличить вероятность распознавания объектов в условиях зашумленных исходных данных.*
- 2. Было показано, что фильтрация соответствий особых точек изображений на основе эвристических предположений относительно свойств особых точек является эффективным способом устранения ложных соответствий, связанных с зашумленностью исходных данных.*
- 3. Практически важными являются данные, полученные экспериментальным путем на тестовых данных, подтверждающие целесообразность применения предложенного алгоритма в задачах компьютерного зрения.*

Список источников

1. Lowe D. Distinctive image features from scale-invariant keypoints // International Journal of Computer Vision. 2004. Vol. 60(2). P. 91-110.
2. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // Computer Vision and Image Understanding (CVIU). 2008. Vol. 110 (3). P. 346-359.
3. Тестовые данные для описателей. [Электронный ресурс]. URL: https://github.com/Itseez/opencv_extra/tree/master/testdata/cv/detectors_descriptors_evaluation/images_datasets (дата обращения: 10.02.2013).
4. Arandjelovic R., Zisserman A. Three things everyone should know to improve object retrieval // CVPR. 2012. P. 2911-2918.

Некоторые аспекты задачи исследования распространения информации в социальной сети ВКонтакте

Евгений Рабчевский, Артем Цукерман

ПГНИУ, Пермь, Россия. evgeny@rabchevsky.name, azholy@gmail.com

Аннотация. Статья посвящена некоторым техническим аспектам обширной задачи по исследованию особенностей распространения информации в социальной сети ВКонтакте. В статье описывается работа по сбору информации со страниц пользователей, индексация и поиск по полученным данным, а также анализ топологии Сети, который говорит о безмасштабности исследованного сегмента.

Ключевые слова: поиск в социальных сетях; анализ и мониторинг социальных медиа; аудит сообществ, исследование общественного мнения в социальных сетях.

Введение

На сегодняшний день социальные сети представляют собой коммуникативную площадку, которая активно используется для формирования и манипулирования общественным мнением. Так, например, сегодня не редкость, когда "ВКонтакте" создаются группы для помощи людям, попавшим в беду, или группы любителей определенных брендов, которые продвигают соответствующую продукцию, и так далее. В данном случае социальную сеть можно рассматривать, как общественную среду, на которую пытаются каким-то образом повлиять для дос-

тижения определенной цели, например формирования определенного общественного мнения по определенному вопросу. Для решения этой задачи, лицо, заинтересованное в формировании определенного мнения (далее будем называть его автором), должно знать, каким образом максимально эффективно привести сеть в необходимое состояние. А для этого требуется определить максимально эффективные каналы распространения информации по сети, пользователей, потенциально готовых с большей вероятностью принять необходимую позицию (например, тех, кто разделяет некоторые взгляды автора), и так далее.

Помимо формирования общественного мнения актуальны такие прикладные задачи как оценка успешности любого нововведения, будь то сервис, сообщество или рекламная компания. Такая оценка дает возможность разрабатывать сервисы не наугад, а целенаправленно повышать пользовательскую активность. Также актуален анализ аномально-го изменения пользовательской активности, поиск целевой аудитории бренда и маркетинговые исследования, связанные с получением данных о позиции пользователей по отношению к услугам какой-либо компании, что требует, однако, более глубокого лингвистического анализа.

Все это требует полноценного анализа сегмента социальной сети, с которым придется работать автору. Более того, этот самый сегмент еще нужно каким-то образом выделить. Также следует принять во внимание тот факт, что такая задача уникальна для каждого автора, и при этом очень востребована в целом для сети.

В этой связи, глубокое понимание особенностей распространения информации в социальных сетях в целом, и ВКонтакте в частности, на практике, например, позволит осуществлять более эффективные маркетинговые мероприятия в социальных сетях.

Постановка задачи

Задача исследования особенностей распространения информации в социальных сетях представляется достаточно сложной. Она включает в себя задачи по

- автоматическому сбору информации из Сети
- анализу собранной информации с учетом изменения по времени
- идентификации источников данных, представляющих информацию по определенной теме (тема, распространение которой исследуется)
- непосредственно анализ полученных на первых этапах данных с целью выявления определенных закономерностей и особенностей.

Данная статья представляет собой отчет о промежуточных результатах решения указанной комплексной задачи, которые на данный момент нельзя считать законченным исследованием, а стоит рассматривать лишь как некую подготовительную работу по решению указанной сложной задачи.

Согласование исследований с правилами пользования сайта ВКонтакте

Для сбора информации из Сети вместо стандартного API ВКонтакте использовалась имитация поведения пользователя при просмотре страницы. Согласно пункту 4.2. правил, пользователем должно быть физическое лицо, а в случае сбора информации при помощи программ, согласно пункту 5.3.9. правил, необходимо получить согласие администрации Сайта.

На данный момент, запрос к администрации Сайта на получение разрешения использования программ для сбора информации из Сети не выполнялся. Это связано с тем, что на данный момент сбор информации проводился единоразово, и не представлял собой существенной нагрузки на Сеть. В последующем, такой запрос будет сделан.

Краулер - программа для сбора данных со страниц Сети

Для сканирования страниц пользователей социальной сети ВКонтакте, далее Сети, на интерпретируемом высокоуровневом языке программирования Ruby была разработана программа – краулер.

Особенность работы краулера состоит в том, что информация в Сети представляется с помощью динамических технологий. И для сканирования Сети краулеру требуется полностью имитировать работу реального пользователя, это достигается формированием определенной последовательности запросов типа POST (например, полностью имитирующей вертикальную прокрутку страницы), отправляемой на HTTP сервер Сети. Ответы сервера Сети, которые являются сообщениями со страниц пользователей, сохраняются краулером в базу данных СУБД MySQL.

Еще одной сложностью при разработке краулера стала особенность поведения пользователей, которая заключается в том, что около 2/3 всех пользователей (по нашей статистике) Сети закрывают свои страницы для неавторизованных пользователей. Для обхода этой проблемы краулер авторизуется на сервере vk.com как зарегистрированный пользователь.

Однако, при чрезмерной активности пользователь блокируется сервером vk.com, поэтому перед переходом на следующую сканируемую страницу краулер делает паузу в 1 секунду, что заметно снижает скорость работы. Чтобы ускорить работу, программа использует многопоточность, где каждый поток обращается к серверу как отдельный зарегистрированный пользователь, а список страниц распределяется между потоками. Программа может работать в нескольких режимах, с использованием авторизации и без, а также из-под прокси-сервера.

Режим без авторизации используется для предварительного сканирования страниц пользователей, которые открыли свои страницы, что позволяет сэкономить время.

В первой версии программы список пользователей был получен парсингом выдачи стандартного поиска vk.com/search (в качестве выборки использовалось местоположение город Пермь). Оказалось, что стандартный поиск vk.com ограничен выдачей в 1000 человек.

Во второй версии это ограничение было преодолено и количество пользователей расширено по требованию задачи. Ограничение было преодолено за счет того, что краулер получал список друзей пользователя, которые проживают в городе Перми, сохранял их в базу данных и заносил в очередь, после чего процедура повторялась с каждым из них, таким образом, удалось выявить необходимое количество пользователей.

При сканировании лишь сообщений со стены пользователей мощности краулера сейчас хватает для того, чтобы охватить один не крупный город. Время обработки линейно зависит от количества параллельных потоков, имитирующих работу пользователя.

Индексация и полнотекстовый поиск

В первой версии краулера, собранная им коллекция документов составила 23318 сообщения, принадлежащих 1000-ти пользователям, что занимает 9.6 МВ в базе MySQL.

Полученная таким образом коллекция проиндексировалась системой полнотекстового поиска Sphinx [1]. В результате чего к коллекции добавился индекс, который с помощью системы Sphinx позволяет очень быстро осуществлять полнотекстовый поиск по всем сообщениям коллекции. Система Sphinx была выбрана вследствие ее эффективности при работе с небольшими сообщениями, каковыми являются сообщения со стены пользователей.

Время индексации составило 11.966 секунды, а скорость соответственно 804320 байт/сек или 1948.54 документов/сек.

Во второй версии коллекция была расширена до 1152586 сообщений 63770 пользователей, и занимает 493.6 МВ. Время индексации составило 2163.567 секунд, а скорость 228142 байт/сек или 532.72 документов/сек.

Доступ к системе поиска по данным сообщениям может быть осуществлен по адресу <http://78.47.43.6/> [2]. Представленная система поиска целенаправленно ограничена по количеству результатов поиска и функционалу работы с ними.

Обработка больших объемов данных

При дальнейшей работе была поставлена задача получения данных обо всех пользователях пермского сегмента Сети. При этом требовалось получить не только сообщения со стены пользователя, а полную информацию с его страницы, включая "лайки", комментарии, интересы и так далее.

Эта задача потребовала БОльших ресурсов и более сложной архитектуры. Поэтому требует и более интеллектуального подхода к сканированию информации со страниц пользователей, нежели одномоментного индексирования. В связи с этим, для решения задачи анализа динамического состояния социальной сети разработана четырехуровневая модель сканирования.

На первом, низшем, уровне модель включает модель данных социальной сети ВКонтакте и библиотеку для извлечения данных со страниц пользователей. На втором уровне модель включает средства, позволяющие на базе данных первого уровня получить анализ характеристик пользователей и материалов с учетом их динамики. Третий уровень включает средства, позволяющие на базе данных второго уровня получить данные о пользователях, сообществах и материалах, обобщенные с точки зрения масштабов всей сети. И, наконец, четвертый уровень включает модели и соответствующие алгоритмы, определяющие пути и способы сканирования сети.

Указанная модель сканирования полностью реализована лишь на первом уровне, остальные уровни реализованы частично. В частности, разработана модель данных социальной сети ВКонтакте. Она предоставляет возможности для последующего анализа индексированных данных в различных разрезах. Данная модель оперирует простейшими фактами, такими как: отправка сообщения, лайк, репост и т.д. Факты складываются в триплеты: «пользователь А» написал «текст сообщения»; «текст сообщения» находится на «странице пользователя А». Таким образом, можно описать все действия пользователей. Удобство такого

представления заключается в возможности получать данные по сложным запросам, например: получить все сообщения, которые «А» написал на стене «Б» и у которых имеются лайки. Модель связывает понятия, с которыми система оперирует при навигации по персональной странице одного пользователя (например, ссылки «Мне нравится», «Друзья», и т.д.), с характеристиками этих элементов навигации, которые определяют, какую часть страницы пользователя и каким образом, следует обрабатывать, для того чтобы извлечь из его страницы необходимую информацию.

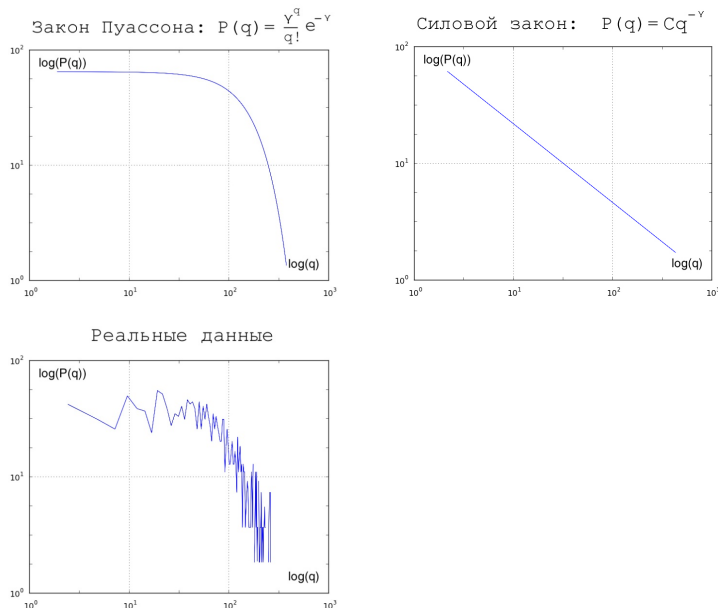
Вопрос о физическом хранении RDF графа, хранящего в себе такого рода информацию, которая указаны выше, на данный момент остается открытым.

Создана библиотека, которая в совокупности с моделью данных социальной сети, позволяет в режиме реального времени получать полную информацию со страницы одного пользователя.

При обработке таких больших объемов данных будут достигнуты ограничения СУБД MySQL по производительности. Поэтому было приятно решение перейти на распределенную базу данных в системе PostgreSQL.

Исследование общественного мнения в сети

Используя возможности программы – краулера для фрагмента Сети, состоящего из 1000 пользователей, были проведены исследования, в ходе которых получены данные о структуре фрагмента Сети. А также определена его топология. Исследованный фрагмент Сети был представлен в виде графа. На графике ниже с меткой «реальные данные» представлено распределение степеней $P(k)$, где k является числом связей, выходящих из данного узла графа (пользователя), а $P(k)$ - это вероятность того, что степень (число связей) случайно выбранного узла равняется k . Видно, что график распределения для исследуемого фрагмента Сети близок к распределению Пуассона. Если увеличить объем исследуемого сегмента, то можно предположить, что в предельном случае распределение совпадет с Пуассоновским. Таким образом, Сеть, как и ожидалось, является безмасштабной [4].



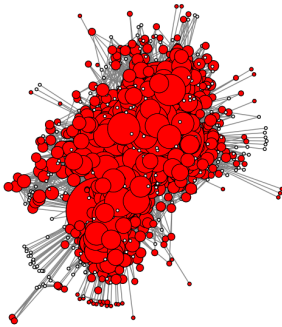
На полученном фрагменте Сети Проведено имитационное моделирование, в ходе которого показано, что мнения агентов стабилизируются [5].

На начальном этапе моделирования мнения пользователей распределялись случайным образом. Каждому пользователю присваивалось значение в диапазоне $[-50..50]$. На последующих шагах пользователи обменивались мнениями, в результате чего мнения стабилизировались и приняли значение равное 50.

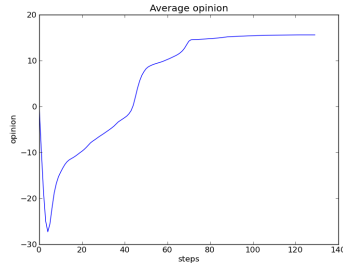
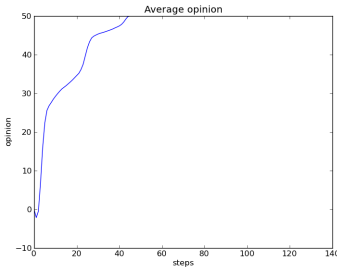
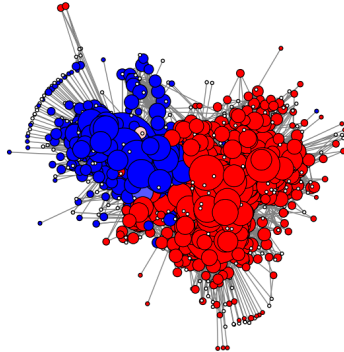
Во втором варианте моделирования мнения пользователей устанавливались также как и в первом, но у выбранной подгруппы мнение было целенаправленно установлено в значение (-50). Ниже представлен конечный результат моделирования двух сценариев – со случайным первоначальным распределением мнения, и с целенаправленно смещенным распределением. Результат представлен на изображении цветом.

На графиках представлены зависимости значения среднего мнения от шага.

случайное
распределение



смещение мнения
подгруппы



Данное имитационное моделирование показывает, что, манипулируя начальными мнениями группы агентов, можно эффективно влиять на итоговое среднее мнение всех членов сети.

В ходе экспериментов выявлено, что сеть устойчива к случайным воздействиям, а целенаправленные воздействия, могут вызвать лавинообразный процесс распространения информации [6]. Данные результаты соответствуют теории безмасштабных сетей.

Для проведения реальных экспериментов по анализу изменения общественного мнения в Сети по определенному вопросу, потребуются формальный механизм преобразования вербальных оценок пользователей Сети определенных тезисов в точное количественное представление. Что планируется реализовывать в контексте определенной предметной области.

Используя систему поиска, расширенную критериями поиска по времени и граф сети, можно будет производить оценку скорости рас-

пространения информации и ее охват, выявлять характерные пути распространения информации, и т.д.

Выводы

- 1. Исследование топологии фрагмента Сети показало, что Сеть можно считать безмасштабной.*
- 2. Для исследования распространения информации в Сети для больших объемов данных требуется согласовать свои действия с администрацией сайта ВКонтакте.*
- 3. При расширении объема хранимой информации требуется переход с MySQL на систему, обладающую более высоким быстродействием и дающую широкие возможности для хранения, доступа и архивирования информации в распределенном виде, например PostgreSQL.*
- 4. Для исследования распространения реальной информации необходимо выбрать предметную область и сформулировать методiku по количественной формализации вербальных оценок пользователей.*

Список источников

1. Платформа полнотекстового поиска <http://sphinxsearch.com/>
2. Демонстрация поиска по пермскому сегменту социальной сети Вконтакте <http://78.47.43.6/>
3. Бизнес-анализ в социальной сети Одноклассники <http://habrahabr.ru/company/odnoklassniki/blog/149391/>
4. Barabasi A.L. Scale Free Networks : Scientific American - http://www.barabasilab.com/pubs/CCNR-ALB_Publications/200305-01_SciAmer-ScaleFree/200305-01_SciAmer-ScaleFree.pdf
5. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети модели информационного влияния, управление и противоборство: Под ред. чл.-корр. РАН Д.А. Новикова. – М.: Издательство физико-математической литературы, 2010. – 288 с. ISBN 9785-94052-194-5
6. Barabasi A.L. Social consensus through the influence of committed minorities - http://arxiv.org/PS_cache/arxiv/pdf/1102/1102.3931v2.pdf

Методы распараллеливания алгоритма сравнения дактилоскопических изображений

Владимир Гудков¹, Дарья Лепихова²

¹ЧелГУ, Миасс, Россия. diana@sonda.ru

²ООО «Сонда Про», Миасс, Россия. daria.lepikhova@yandex.ru

Аннотация. В статье рассмотрены возможности для распараллеливания алгоритма сравнения дактилоскопических изображений, основанного на модели топологических векторов. Рассматривается параллелизм на уровне данных, а также реализация алгоритма на базе графических процессоров

Ключевые слова: отпечатки пальцев, параллельный алгоритм, OpenMP, CUDA

Введение

Цель работы: рассмотреть возможные варианты для организации параллельных вычислений на различных этапах алгоритма сравнения дактилоскопических изображений (далее – ДИ), а также произвести сравнительный анализ эффективности рассмотренных вариантов.

В работе рассматривается алгоритм сравнения ДИ, основанный на модели топологических векторов. Сравнение дактилоскопических изображений выполняется путем сравнения их шаблонов, в которых хранится информация о признаках изображения и связанных с ними топологи-

Методы распараллеливания алгоритма сравнения ДИ

ческих векторах [7]. Топологический вектор представляет собой нумерованный набор связей с упорядоченными парами (e_l, n_l) , где e_l – событие, сформированное на связи частным признаком с номером n_l . Несмотря на то, что использование топологических векторов в качестве дополнительного критерия при сравнении увеличивает размер шаблона, также обеспечивает и значительно большую устойчивость алгоритма к дефектам изображений. Подробно математическая модель сравнения ДИ, основанная на топологических векторах, и алгоритм, использующий данную модель, описаны в работах [4, 7].

С развитием средств параллельного программирования предпринимались попытки ускорить работу идентификационных систем за счет параллельной обработки больших массивов изображений. В работе [6] рассматриваются общие вопросы распараллеливания биометрических вычислений на базе вычислительного кластера.

Вопросы распараллеливания алгоритма, использующего модель топологических векторов, рассмотрены в работе [4]. Приведена параллельная версия алгоритма, основанная на распараллеливании выполнения оценок по топологии и по геометрии в рамках одного сравнения.

Параллельные реализации

Структура существующего параллельного алгоритма позволяет использовать следующие подходы к организации параллельных вычислений.

- Параллелизм на уровне задач. Подробное описание и схема данного уровня параллелизма содержатся в работе [1], поэтому в статье приведена только его краткая характеристика. Суть метода заключается в организации параллельной обработки наборов связей при выполнении оценок подобия по геометрии и по топологии с помощью директив OpenMP. Такой подход позволяет добиться практически линейного ускорения на небольшом количестве ядер. На рис. 1 представлен график зависимости ускорения параллельного алгоритма от числа задействованных потоков. На 4 потоках данная версия демонстрирует двукратное ускорение по сравнению с последовательной версией. Эксперименты проводились на вычислительной системе со следующими параметрами:
 - количество процессоров: 4;
 - тип процессоров: Intel Atom 1,6 ГГц;
 - оперативная память 1 Гб.

Методы распараллеливания алгоритма сравнения...

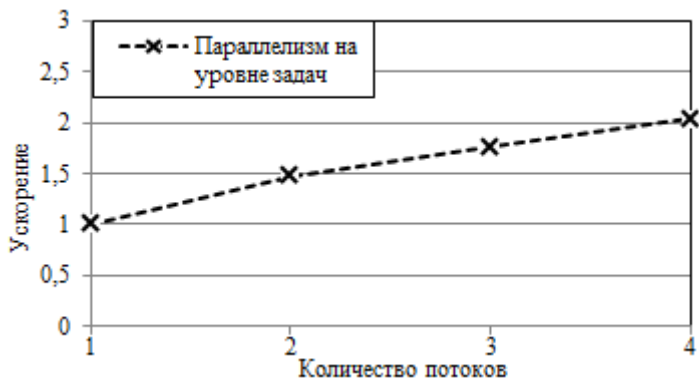


Рис. 1. Ускорение при использовании параллелизма на уровне задач

Под ускорением алгоритма понимается следующее. Пусть дан исходный последовательный алгоритм, который выполняет сравнение шаблонов за время t_0 . Тогда ускорение параллельного алгоритма на n потоках определяется формулой $q_l = \frac{t_0}{t_n}$, где t_n — время работы алгоритма на n потоках.

Для проведения испытаний использовались тестовые базы Db1 и Db2 из наборов FVC2002 [1] и FVC2004 [2]; примеры используемых изображений показаны на рис. 2. Время работы алгоритма замерялось без учета времени, затрачиваемого на чтение данных с диска и размещение их в оперативной памяти.



Рис. 2. Примеры изображений

- Параллелизм по данным.

Методы распараллеливания алгоритма сравнения ДИ

Этот уровень параллелизма предполагает, что каждый из параллельных процессов решает одну и ту же задачу сравнения двух ДИ для своего набора сравниваемых пар признаков. При этом задача сравнения может решаться по любой из схем, но в данной статье рассматривается схема, представленная на рис. 3, поскольку данная схема предполагает более прозрачное разделение процесса сравнения на отдельные этапы и больше возможностей для исследования влияния каждого из этих этапов на качество сравнения.

- Использование графических процессоров. Современные видеокарты позволяют организовать обработку базы шаблонов так, что каждая нить, как и в случае параллелизма по данным, выполняет одно сравнение.

Параллелизм на уровне данных

Основная идея подхода параллелизма по данным заключается в том, что для всех элементов исходного массива данных выполняется одна и та же операция. Т.е. каждый из запущенных параллельных потоков будет выполнять сравнение двух шаблонов по одной и той же схеме.

Сравнение шаблонов может проводиться, например, при помощи исходного последовательного алгоритма. Помимо этого можно также добавить новый уровень параллелизма – одновременное выполнение оценок по топологии и по геометрии. Такое разделение оценок позволит, помимо всего прочего, изучить влияние топологических и геометрических характеристик на скорость и качество сравнения.

Для реализации такого подхода можно использовать сочетание с OpenMP различных технологий параллельного программирования. Схема параллельного алгоритма, реализующего предложенную модель, приведена на рис. 3. Необходимо, однако, учитывать, что накладные расходы на создание и поддержку процессов не должны превышать выигрыш по времени от использования параллельных вычислений. Поэтому использование такого подхода в рамках одного сравнения может оказаться нецелесообразным, но в случае одновременного выполнения большого количества сравнений по предложенной схеме его применение оправдано.

Будем считать, что база шаблонов полностью помещается в оперативной памяти.

Методы распараллеливания алгоритма сравнения...

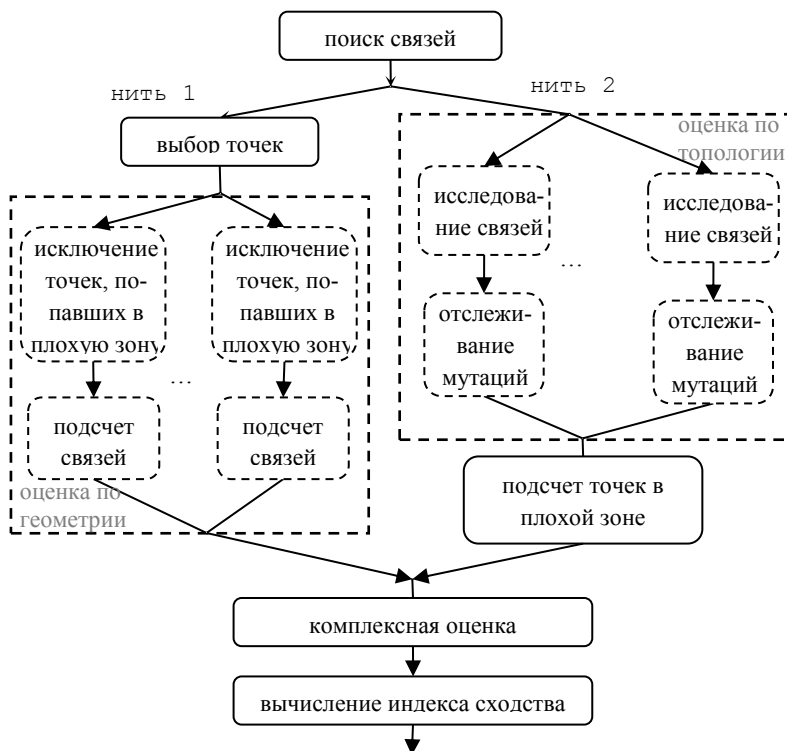


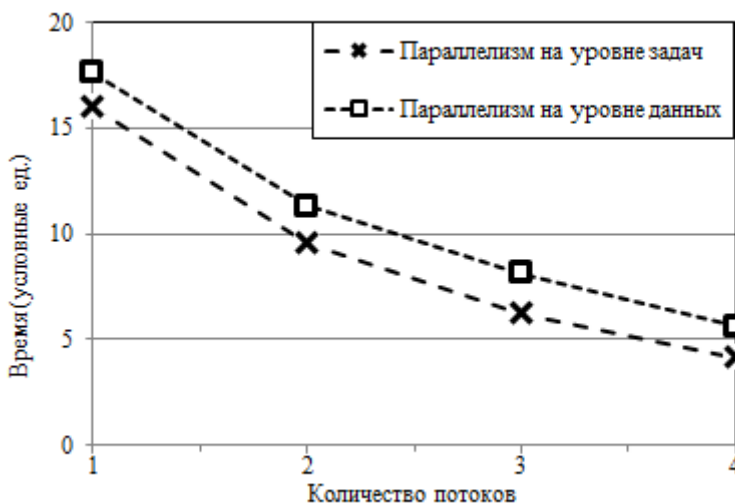
Рис. 3. Схема одновременного выполнения оценок

Представленный подход использует концепцию общей памяти. Каждый из запущенных параллельных процессов выполняет сравнение всей базы шаблонов с некоторой его частью. Каждому из параллельных потоков необходим доступ к переменной *Templates* для получения шаблонов, а также к переменным *Gen* и *Imp*, в которых хранятся распределения «родных» и «чужих» индексов.

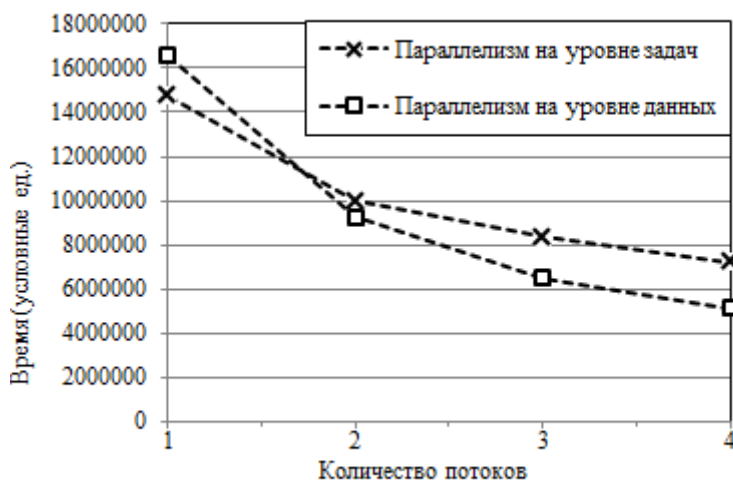
Matching() – сравнение шаблонов по схеме, приведенной на рис. 3.

На рис. 4 представлены графики зависимости времени работы алгоритма от числа параллельных процессов как в рамках одного сравнения, так и для обработки всей базы шаблонов. Для экспериментов использовалась вычислительная система с теми же характеристиками, что и в модели параллелизма на уровне задач, и та же самая база изображений.

Методы распараллеливания алгоритма сравнения ДИ



а) одно сравнение



б) общее время для всей базы шаблонов

Рис. 4. Сравнение времени работы алгоритма для случаев параллелизма на уровне задач и на уровне данных

Методы распараллеливания алгоритма сравнения...

На графиках видно, что, несмотря на увеличение времени, за которое выполняется одно сравнение, время обработки всей базы шаблонов значительно сокращается. Ускорение алгоритма при обработке всей базы шаблонов близко к линейному, что видно на графике на рис. 5.



Рис. 5. График_сравнительное_ускорение

На сегодняшний момент подходы параллелизма на уровне задач и на уровне данных реализованы и успешно применяются на практике.

Графические процессоры

В настоящее время возможности современных видеокарт позволяют производить вычисления над большими объемами данных [5], при этом GPU предназначены для интенсивных расчетов.

Можно предложить следующую схему работы алгоритма. В рамках блока решается задача сравнения для некоторого сегмента базы шаблонов. Такой подход позволяет масштабировать задачу путем увеличения количества выполняемых блоков. Предлагается внутри блока производить сравнение с одним конкретным шаблоном. При этом шаблон, для которого производятся сравнения, хранится в разделяемой памяти, что обеспечивает быстрый доступ к нему всех нитей внутри блока.

Каждая нить блока будет выполнять одно сравнение, сохраняя полученный в результате индекс сходства в локальной переменной. Далее необходимо будет собрать полученные индексы для всей базы шаблонов в единую матрицу, на основе которой затем формируются массивы *Gen* и *Imp* и вычисляются значения ошибок FAR и FRR, на основе которых делается вывод о качестве работы алгоритма. Общая матрица ин-

Методы распараллеливания алгоритма сравнения ДИ

дексов сходства шаблонов хранится в глобальной памяти, доступ к которой имеют все нити. Схема такого подхода показана на рис. 6.

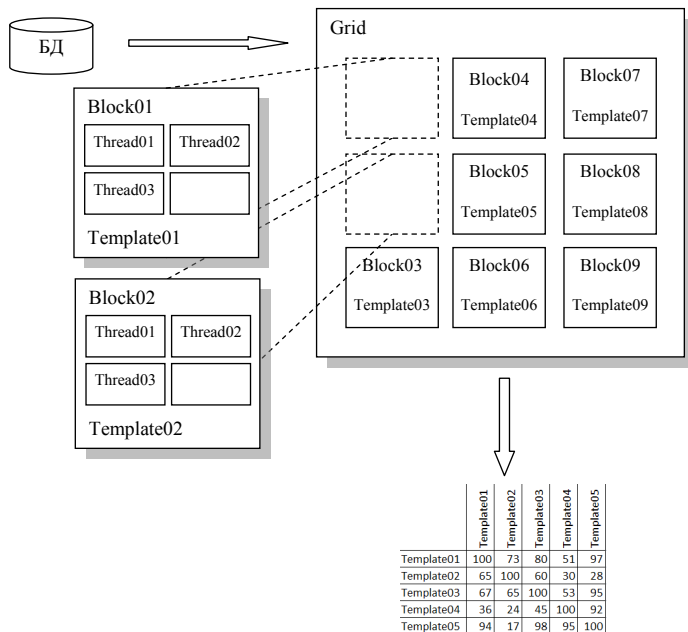


Рис. 6. Схема алгоритма для CUDA

Ядро программы выполняется параллельно для каждого набора элементов. В рассматриваемом алгоритме ядром будут являться функция вычисления индекса сходства шаблонов ДИ.

В качестве направления дальнейших исследований можно выделить изучение зависимости быстродействия алгоритма от размера блоков CUDA.

Заключение

В данной статье рассматривается применение различных подходов к организации параллельных вычислений в алгоритме вычисления индекса сходства дактилоскопических изображений. Рассматриваются уровни параллелизма по данным и по задачам, а также вычисления на графическом процессоре.

Методы распараллеливания алгоритма сравнения...

Выводы

Параллельная версия алгоритма, использующая модель параллелизма по данным и предполагающая одновременное выполнение оценок по геометрии и по топологии демонстрирует заметно более значительное снижение быстродействия, чем версия, использующая параллелизм на уровне задач. Это связано с тем, что данная версия фактически создает дополнительный параллельный регион в рамках каждого проводимого сравнения, что требует значительных накладных расходов.

Предлагаемая версия алгоритма для графических процессоров предполагает прозрачную масштабируемость задачи, позволяет легко выделять отдельные подзадачи, например, выбрать все «родные» шаблоны для шаблона с заданным номером.

Список источников

- 1 FVC2002 website [Электронный ресурс]. URL: <http://bias.csr.unibo.it/fvc2002> (дата обращения 16.05.2012).
2. FVC2004 website [Электронный ресурс]. URL: <http://bias.csr.unibo.it/fvc2004> (дата обращения 16.05.2012).
3. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. М.: ДМК-Пресс, 2010. 232 с.
4. Гудков В.Ю., Лепихова Д.Н. Параллельное программирование алгоритма идентификации дактилоскопических изображений // ГрафиКон'2012: Труды 22-й межд. конф. – 2012. – С. 275–277.
5. Славин О.А. Методы ускорения алгоритмов распознавания символов // Труды ИСА РАН «Технологии программирования и хранения данных», том 45, 2009. С. 287-299.
6. Ушмаев О.С. Проблемы распараллеливания сложных вычислений в крупномасштабных информационных системах // Информатика и ее применения, т. 3, вып. 1. – 2009. – С. 8-18.
7. Гудков В.Ю. Математические модели изображений отпечатка пальца на основе описания линий // Информатика и ее применения, 2010. Т. 4, вып. 1. С. 58-64.

Анализ тональности текста на русском языке при помощи графовых моделей

И. Л. Меньшиков

unkmas@gmail.com

УРФУ, Екатеринбург, Россия

Аннотация. Статья посвящена вопросу анализа тональности текста на русском языке при помощи графовых моделей. Описан и экспериментально исследован алгоритм решения данной задачи.

Ключевые слова: компьютерная лингвистика; обработка естественного языка; анализ тональности.

1 Введение

Количество генерируемого пользователями контента в интернете выросло экспоненциально за последнее десятилетие. Пользователи пишут на форумах, в блогах, оставляют комментарии на множестве страниц и пользуются социальными сетями. Согласно исследованиям Всероссийского центра изучения общественного мнения, количество россиян, регулярно (не реже раза в месяц) пользующихся интернетом выросло с 38% в 2010 г. до 55% в 2012 г. Число зарегистрированных в социальных сетях россиян за эти 2 года (с 2010 по 2012 гг.) также значительно возросло – с 53% до 82%. [1] Весь этот контент несет в себе огромное количество информации, которой мы регулярно получаем, анализируем и используем.

Для владельцев информационных ресурсов жизненно важно знать мнение пользователей — будь это оценка людьми нового про-

дукта в интернет магазине или отношение к свежей новости на новостном сайте. Для простого пользователя интернет-магазина будет интересна информация о том, насколько другим покупателям понравился или не понравился конкретный товар.[2] Однако, вся эта информация представляет собой большой объем текстовых данных. Для того, чтобы прочитать их и проанализировать требуется много времени. Для решения этой проблемы необходимы системы анализа тональности текста.

Анализ тональности текста — это класс методов, предназначенный для выявления эмоций в тексте. Он позволяет охарактеризовать текст по его эмоциональной окраске — положительный, отрицательный, нейтральный текст. Кроме того, возможно определение силы тональности, субъекта/объекта тональности и многих других характеристик текста.

2 Предпосылки

Исходными данными для этой работы послужило предположение о том, что не все слова в тексте равнозначны. Какие-то слова имеют больший вес, более значимы для данного текста. Какие-то слова — менее значимы. Очевидно, что более значимые слова будут оказывать более сильное влияние на общую тональность текста.

При этом слова, имеющие высокую силу тональности, могут оказывать большее влияние на тональность, нежели слова, имеющие больший вес, однако меньшую силу тональности.

3 Алгоритм

Анализ тональности происходит в несколько этапов:

- 1) Построение графа на основе текста
- 2) Ранжирование его вершин
- 3) Классификация найденных слов
- 4) Вычисление результата

Этапы 1 и 2 детально описаны в статье [3]. Этапы 3 и 4 описаны далее в данной статье.

4 Классификация слов

Для определения классов слов и определения силы их тональностей используется тональный словарь.

В данной работе, для слов использовалось два класса:

- Положительное слово

— Отрицательное слово

Оценка силы тональности проводилась по шкале от 1 до 5 для каждого класса.

5 Вычисление результата

Для получения итогового результата необходимо получить две оценки: оценку положительной составляющей текста и оценку его негативной составляющей.

Для оценки положительной составляющей необходимо подсчитать сумму тональностей всех найденных положительных терминов текста, с учетом их веса:

$$P = \sum_i TR(i) * Q(i), \quad (1)$$

где P — оценка положительной составляющей текста, $TR(i)$ - вес слова, $Q(i)$ - сила его тональности.

Аналогичным образом вычисляется значение отрицательной составляющей текста (N).

Для итоговой оценки тональности текста вычисляется отношение этих оценок:

$$T = P/N \quad (2)$$

Текст, в котором значение T близко к единице, считается нейтральным. Текст, в котором значение T больше или меньше единицы, считается положительным или, соответственно, отрицательным.

Кроме того, возможно разделение текстов на большее число классов — если T незначительно превосходит единицу, текст считается слабо-положительным, если же он намного больше единицы — сильно-положительным.

6 Результаты работы алгоритма

Для исследования работы алгоритма выполнена обработка ряда текстов, размеченных экспертами.

Было выбрано 40 текстов, каждый из которых был отмечен экспертами как положительный или отрицательный. Проверялось соответствие результатов, выданных системой результатам оценки экспертов. Система правильно обработала 66% текстов.

Для оценки результатов работы системы, тексты были разбиты на три группы, в соответствии с количеством найденных терминов.

Тексты, в которых обнаружено менее 30 терминов считаются мелкими, от 30 до 70 — средними и более 70 — крупными. Результаты приведены в Табл 1, где приняты следующие обозначения: С — количество обнаруженных в тексте терминов, А — отношение числа правильно проанализированных текстов данной группы к общему числу текстов данной группы, Т — доля терминов, несущих эмоциональную окраску среди всех найденных терминов. Для получения величины Т для расчетов брались средние значения среди группы, σ — стандартное отклонение Т.

Табл. 1. Результаты работы системы

С	σ	Т	А
< 30	18%	2,3%	75%
30 - 70	12%	5,8%	57%
> 60	14%	6,1%	71%

Как видно из вышестоящей таблицы, лучшие результаты система показала для коротких и крупных текстов, с небольшим снижением точности на средних.

В ряде случаев, система выдавала ошибку из-за отсутствия ряда эмоционально окрашенных слов в тональном словаре. При использовании более полного тонального словаря точность работы системы повысится.

Однако, система не способна корректно обработать некоторые тексты. Например, отрицательный текст, написанный положительными словами с большим количеством отрицаний, не будет правильно обработан, так как слова-отрицания (“не”, “никогда” и т.д.) в данном алгоритме не учитываются.

7 Заключение

Описанный алгоритм показал хорошие результаты, которые могут быть улучшены путем. У алгоритма выявлены недостатки, трудные для устранения, однако точность работы системы можно улучшить.

Пути улучшения работы системы:

- расширение тонального словаря
- учет не только самих слов, но и взаимосвязей между ними

Список источников

1. РИФ+КИБ: Тренды Рунета-2012: всегда и везде быть в сети [Электронный ресурс]: Всероссийский центр изучения общественного мнения. – Режим доступа: <http://wciom.ru/index.php?id=270&uid=112746> 28.11.2012
2. Bo Pang, Lillian Lee: Opinion Mining and Sentiment Analysis // Journal Foundations and Trends in Information Retrieval. 2008. С. 1–135.
3. Усталов Д. А. Извлечение терминов из русскоязычных текстов при помощи графовых моделей // Теория графов и приложения = Graphs theory and applications : материалы конференции. — Екатеринбург : Изд-во Урал. ун-та, 2012. — С. 62–69.

Ошибки первого и второго рода для проставки частных признаков на изображении отпечатка пальца

Дорофеев Константин Андреевич

Миасский филиал ЧелГУ, Миасс, Россия. Kostuan1989@mail.ru

Аннотация. Статья посвящена сравнению частных признаков дактилоскопических изображений, полученных автоматическим методом скелетизации и шаблонов, составленных экспертами в предметной области. Посчитаны ошибки первого и второго рода.

Ключевые слова: Дактилоскопия; отпечаток пальца; скелетизация; частные признаки; шаблон.

Введение

Цели исследования:

- предложить способ вычисления ошибок первого и второго рода, адекватно отражающий качество расстановки частных признаков дактилоскопических изображений [7];
- вычислить значения ошибок первого и второго рода для реализованного автоматического метода [1] для базы дактилоскопических изображений NIST 14;

Опишем алгоритм сравнения результатов автоматической поставки частных признаков и шаблона, составленного экспертом-криминалистом. Пусть $A = \{a_i\}$ – множество частных признаков из шаблона, $B = \{b_i\}$ - множество частных признаков, полученных некоторым автоматическим методом. B_f – количество “ложных” частных при-

Ошибки первого и второго рода для простановки частных признаков на изображении отпечатка пальца знаков из B , т.е. таких частных признаков, которые не находятся в некоторой окрестности (размер окрестности – меняющийся параметр) от какого-либо частного признака из A . A_f – количество “отсутствующих” частных признаков из A в B , т.е. таких частных признаков из A , для которых нет “пары” из B (находящейся в некоторой окрестности). Тогда некоторым аналогом ошибки первого рода для простановки частных признаков при сравнении с отметками эксперта можно считать:

$$\alpha = \frac{B_f}{|B|}$$

Аналогом ошибки второго рода можно считать:

$$\beta = \frac{A_f}{|A|}$$

Проблема “повернутых” вокруг центра изображений решается с помощью локального поля направлений. Выполняется процедура распознавания ориентации локальных окрестностей [6,8] и осуществляется “поворот” окрестности по данным этой ориентации.

Экспертами были составлены шаблоны, на которых были отмечены частные признаки (окончание и разветвление линии). Результаты для конкретного изображения представлены на рис. 1.а. Автор считает, что ошибки эксперта распределены по нормальному закону, поскольку эталоны изображений отпечатков пальцев из базы кодировали разные эксперты независимо. Реализованным автоматическим методом [1] также были отмечены частные признаки на изображениях отпечатков пальцев. Результаты для того же изображения представлены на рис. 1.б.

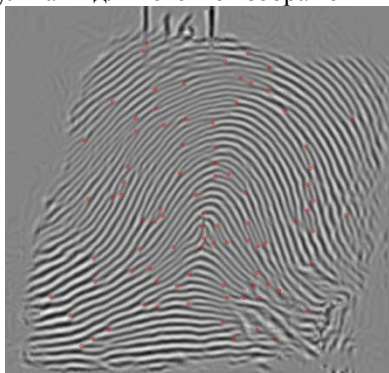


Рис. 1.а Шаблон



Рис. 1.б Метод скелетизации

Для сравнения частных признаков использовалась база NIST 14. Полученные результаты для ошибок первого и второго рода, в зависи-

Ошибки первого и второго рода для простановки частных признаков на изображении отпечатка пальца мости от порога (размера допускающей окрестности), приведены в таблице 1.

Порог (размер области)	Ошибки	
	α	β
14x14	0,05	0,07
12x12	0,09	0,1
10x10	0,11	0,15
8x8	0,17	0,35
6x6	0,43	0,51
4x4	0,65	0,75

Таблица 1 Ошибки I и II рода.

График зависимости ошибок I и II рода от порога представлен на рис. 2.

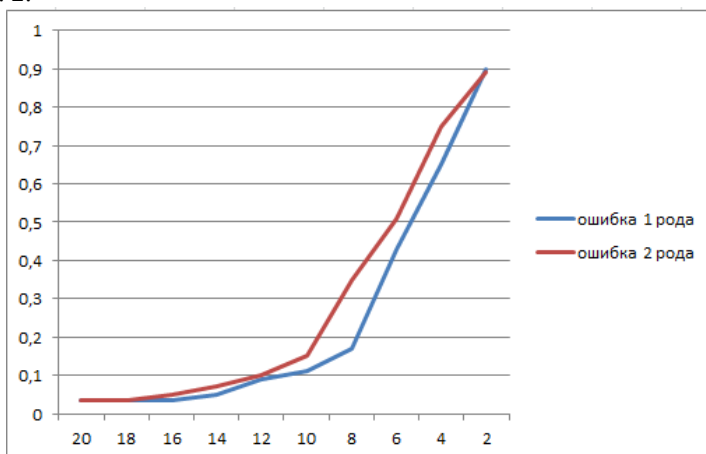


Рис. 2 Зависимость ошибок I и II рода от порога

Ограничение порога сверху значением в 14 пикселей связано со средним значением расстояния между соседними частными признаками в шаблоне, составленном экспертом.

Выводы

Вопрос оценки качества равенства признаков крайне актуален [2,3], особенно в криминалистике, из-за сильно загрязненных изображений следов отпечатков пальцев с мест преступлений. Задавая нулевой порог, требуется, чтобы частные признаки, полученные вручную и автоматическим методом, располагались точка в точке. Это идеальный случай, который не достигим в действительности. Существуют некоторые проблемы, связанные с описанным

Ошибки первого и второго рода для простановки частных признаков на изображении отпечатка пальца

способом подсчета ошибок первого и второго рода. Например – компактно расположенные ложные и истинные частные признаки [4,5]. Планы на будущее состоят в построении некоторой метрики для сопоставления частных признаков, решающей такие проблемы, а также в оценке точности автоматического распознавания частных признаков, основанной на такой метрике.

Список источников

1. Гудков В.Ю. Методы первой и второй обработки дактилоскопических изображений: монография / В.Ю. Гудков. – Миасс: изд-во “Геотур”, 2009. – 237 с. – ISBN: 978-5-8920-4151-5.
2. А.С. 1652984 СССР, МКИ G 06 K 9/00. Способ формирования признаков при распознавании изображений объектов / Г.Е. Баскин, В.И. Гордиенко, Л.С. Королюк, Б.П. Русын. - № 4468868/24; заявл. 01.08.88; опубл. 30.05.91, Бюл. № 20. – 9 с.
3. Bolle R.M. Guide to biometrics / R.M. Bolle, J.Y. Connel, S. Pankanti, N.K. Ratha. – New York: Springer-Verlag, 2004. – 368 с.
4. Гуревич И.Б. Проблемы распознавания изображений: распознавание, классификация, прогноз: математические методы и их применения / И.Б. Гуревич. – Москва: Наука, 1982. – Вып. 1. – 237 с.
5. Дактилоскопическая экспертиза: современное состояние и перспективы развития / В.Е. Корноухов, В.К. Анциферов, Г.П. Морозов и др.; под ред. Г.Л. Грановского. – Красноярск: Изд-во Красноярского университета, 1990. – 416 с.
6. Яне Б. Цифровая обработка изображений / Б. Яне. – Москва: Техносфера, 2007. – 584 с.
7. Самищенко С.С. Современная дактилоскопия: проблемы и тенденции развития: монография / С.С. Самищенко. – Москва: Типография Академии управления МВД России, 2002. – 132 с.
8. Bazen A.M. Fingerprint identification – feature extraction, matching, and database search / A.M. Bazen, 2002. – 187 с.

Проблемы применения классических методов распознавания для фотографических изображений пыльцевых зерен

Андрей Черных, Елена Замятина

Пермский государственный национальный исследовательский университет,
Пермь, Россия, subaromows@gmail.com, e_zamyatina@mail.ru

Аннотация. Рассматриваются проблемы применения классических методов распознавания фотографических изображений зерен пыльцы. Исследования относятся к области палинологии и предназначены для решения ряда задач, таких, например, как определение начала пыления тех или иных растений. Авторы доклада последовательно рассматривают этапы предварительной обработки фотографических изображений пыльцевых зерен (масштабирование, сегментация, бинаризация) и результаты применения метода потенциальных функций и лингвистического (структурного) метода. Излагаются проблемы, которые возникают при распознавании пыльцевых зерен и возможные пути их решения

Ключевые слова: палинология; пыльцевые зерна; предварительная обработка изображений; масштабирование; сегментация; бинаризация; метод потенциальных функций, структурный (лингвистический) метод, нейронные сети.

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

Введение

В работе рассматривается ряд классических методов распознавания изображений (лингвистический метод и метод потенциалов). Исследуется возможность их применения к распознаванию фотографических изображений пыльцевых зерен. Задача распознавания пыльцевых зерен относится к области палинологии. Споро-пыльцевой анализ широко применяется для решения палеоботанических, геоморфологических и геологических (стратиграфических) задач, при изучении состава перги и пыльцы в мёде (мелиттопалинология), в криминалистике, при выяснении причин возникновения некоторых видов аллергий.

Споро-пыльцевым анализом занимаются и исследователи биологического факультета Пермского государственного национального исследовательского университета [1]. Споро-пыльцевой анализ включает сбор пыльцевых зерен, их обработку (методы обработки выбирают в соответствии с задачей исследований) и, наконец, распознавание, т.е. отнесение распознанных зерен к тому или иному классу (рис.1.).

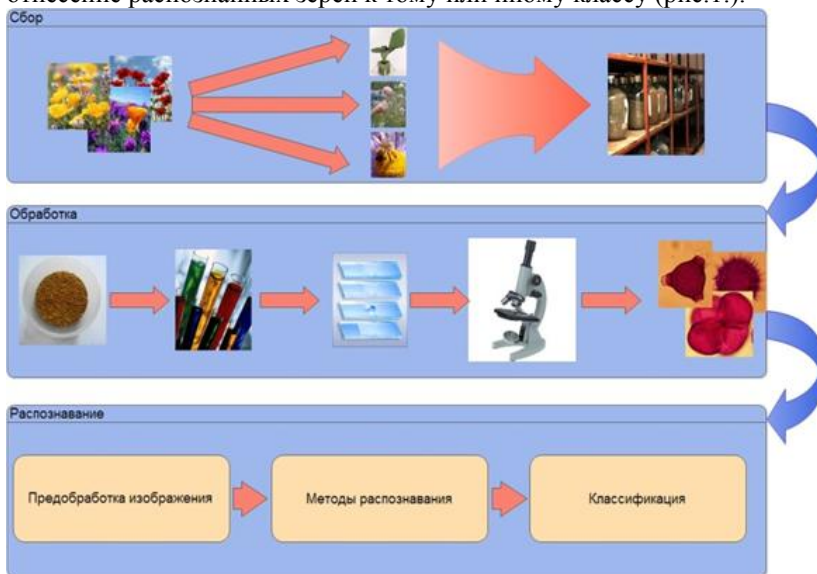


Рис. 1. Этапы распознавания фотографических изображений пыльцевых зерен

Решение задачи распознавания – трудоёмкий этап исследований. В настоящее время этот этап пыльцевого анализа выполняется вручную. Таким образом, актуальной становится задача автоматизации процесса распознавания пыльцы. Предполагается, что исходными данными для

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

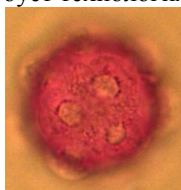
программной системы распознавания пыльцевых зерен являются фотографические изображения пыльцевых зерен, а результатом работы – заключение о том, к какому классу они принадлежат.

Результаты работы системы распознавания могут быть использованы в медицине следующим образом: по количеству и составу пыльцевых зерен в ловушке (один из способов сбора пыльцы) определяют начало периода пыления тех или иных растений, которые являются аллергенами. Обычно возникает необходимость предупредить о начале пыления растений, способных вызвать аллергическую реакцию, людей, страдающих астмой или другими видами аллергических заболеваний.

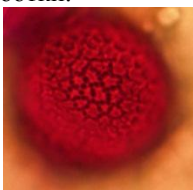
В свою очередь, анализ перги и пыльцы в меде позволяет сделать заключение о качестве исследуемого меда.

Исходные данные для программной системы распознавания пыльцевых зерен

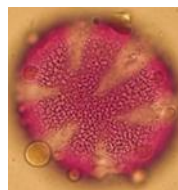
Итак, исходными данными для программной системы распознавания являются растровые фотографические изображения, на которых располагаются одно или несколько пыльцевых зерен. Следует отметить, что пыльцевые зерна очень часто обрабатываются красителем, как того требует технология их обработки.



Гвоздика-травянка



Крестоцветные



Мята

Рис. 2. Этапы распознавания фотографических изображений пыльцевых зерен

Одним из важнейших признаков при установлении морфологических типов пыльцы является строение апертуры (эластичные, гибкие, чаще тонкие или даже перфорированные места, служащие для выхода пыльцевой трубки), их число и расположение на поверхности пыльцевого зерна. Апертуры бывают простые (борозды, щели, поры и др.) и сложные, у которых борозды, поры и прочие образования обладают дополнительной апертурой. Примеры пыльцевых зерен и характерных для этих зерен апертур приведены на рис.2.

Рассмотрим более подробно этап распознавания пыльцевых зерен и проблемы, возникающие при использовании различных методов, как

Проблемы применения классических методов распознавания для фотографических зерен пыльцы предварительной обработки фотографических изображений, так и при применении методов распознавания и проведении классификации.

Этап распознавания пыльцевых зерен

Итак, этап распознавания пыльцевых зерен состоит из следующих последовательных шагов (рис. 1.): (а) предварительная обработка; (б) применение методов распознавания; (в) классификация.

Предварительная обработка изображений предполагает: (а) масштабирование изображения; (б) устранение шума на изображении; (в) сегментация.

Приведем решения, которые были приняты на каждом этапе предварительной обработки изображений. Прежде всего, рассмотрим проблемы, возникающие при масштабировании. В микроскопах, с помощью которых получают фотографические изображения пыльцевых зерен (рис. 3), используют линейный масштаб – масштабную линейку, разделенную на равные части. Размер линейки микроскопа – 20 *um* (*um* – это микрометры (мкм) или микроны (устаревшее)).

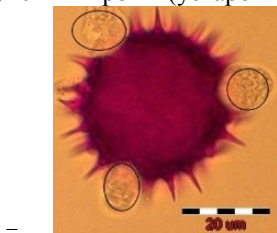


Рис. 3. Пример фотографического изображения пыльцевого зерна, обработанного красителем

Основное назначение линейного масштаба – определять размер пыльцы и элементов ее текстуры. Под текстурой понимают рисунок поверхности пыльцевого зерна, обусловленный его структурой.

Размеры пыльцевых зёрен и спор экземпляров одного вида растений варьируют незначительно, но разные растения характеризуются спорами или пыльцевыми зёрнами, которые имеют размеры, весьма отличные друг от друга. Размер пыльцевого зерна не является ключевым признаком при распознавании, но очень часто может быть использован при отнесении пыльцевого зерна к тому или иному классу.

Далее рассмотрим процедуру устранения шума.

Итак, края изображения пыльцевых зерен могут быть нечёткими, размытыми. Этот эффект объясняется наличием шума, который возникает при обработке пыльцевых зерен микроскопами (чаще всего, тоннельными – современные микроскопы позволяют получать качествен-

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

ные изображения, на которых шум отсутствует). Для устранения шума автоматизированная система распознавания использует ранжирующий фильтр [2].

Следующий шаг в предварительной обработке изображений – это сегментация. Известно, что сегментация – это процесс разделения цифрового изображения на несколько сегментов (сегментом называют множество пикселей, иначе сегмент называют еще суперпикселем). Целью сегментации является упрощение и/или изменение представления изображения и приведении его к виду, удобному для распознавания.

Существует большое количество методов сегментации [2, 3], которые основаны на кластеризации, на подборе модели и т.д. Одним из часто используемых методов сегментации является сегментация на основе пороговой обработки. Среди них различают методы сегментации, основанные на локальном и глобальном порогах.



Рис.4. Результаты обработки фотографического изображения пыльцевого зерна методом бинаризации Отсу

На рис.3. представлено пыльцевое зерно, фиолетовый цвет которого обусловлен красителем, с помощью которого зерна обрабатываются. Пыльцевое зерно контрастирует с фоном. На фоне видны разводы. Для того чтобы в результате сегментации разводы на фоне не могли восприниматься в качестве пыльцевых зерен или их компонентов, целесообразно выбрать метод сегментации, основанный на обработке глобального порога. В качестве такого метода был выбран метод глобальной бинаризации Отсу. Этот метод отличается хорошим быстродействием. Результаты применения метода глобальной бинаризации Отсу приведены на рис.4.

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

Методы распознавания

Для распознавания пыльцевых зерен на настоящем этапе исследований были рассмотрены два классических метода: лингвистический метод и метод потенциалов.

Лингвистический (синтаксический) метод анализа цепочки букв алфавита для проверки ее правильности можно применить и к анализу изображений. Для реализации метода выполняют поточечный обход *контура* исходного изображения и получают цепочку символов. Обход начинают с самой левой нижней точки изображения. Для выбранной точки определяют «соседей», т.е. точки, принадлежащие изображению, а не фону. Соседние точки просматриваются по ходу часовой стрелки. Затем осуществляется переход к новой точке – найденному соседу. Если сосед находится вверху, то переход к нему кодируется 1, по диагонали вверху слева – 2, справа – 3, по диагонали снизу справа – 4 и т.д. до 8. Если у точки несколько соседей, то переход осуществляется на первую из них, а остальные отбрасывают. Таким образом, любой обход описываю как последовательность цифр, например 112233311. После упрощения цепочки по правилам подстановки преобразованную цепочку сравнивают с кодами известных образов и делают вывод о принадлежности изображения одному из классов изображений. Правила подстановки используют для устранения шума[4], последовательность цифр 111122111111 заменяют последовательностью 111111111, считая, что смещение контура на 2 пикселя влево в линии, состоящей из 10 пикселей (соотношение 1:5) является случайным и произошло из-за нечеткости в обработке изображения. Апертура в этом случае не используется.

Метод потенциальных функций предполагает, что с каждым образом (образ в нашем случае – изображение пыльцевого зерна в виде набора пикселей, которые либо закрашены, либо нет), появившимся в процессе обучения, связана некоторая функция, аналогичная по форме электрическому потенциалу. Функция имеет максимальное значение для этого образа и убывает по всем направлениям от него (образ, таким образом, окажется как бы источником потенциала). Такой функцией может быть евклидово расстояние между образами (образом, представленным для обучения, и образом, представленным для распознавания). Чем меньше значение расстояния, тем образы более схожи между собой. При определении евклидова расстояния между образами изображения предварительно были закодированы [5](1 – закрашенный пиксель, 0 – незакрашенный, к 1 закрашенного пикселя при кодировании прибавляется по 0.5 от соседних закрашенных пикселей). Апертура в этом случае используется для распознавания в полной мере.

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

Результаты тестирования программы

Программный комплекс (ПК), предназначенный для распознавания пылевых зерен, был протестирован. Результаты тестирования для лингвистического и метода потенциальных функций приведены в табл.1 и табл.2.

Табл. 1. Результаты тестирования ПК для распознавания фотографических изображений пылевых зерен лингвистическим методом

Наименование пылевого зерна	Обучение	Тестирование	Правильно распознанные	В %
Борщевик	5	26	22	84,62%
Вереск	6	30	24	80,00%
Земляника	5	22	7	31,82%
Иван чай	5	23	20	86,96%
Подсолнух	7	34	33	97,06%
Гвоздика	7	32	11	34,38%
Вьюнок	8	27	11	40,74%
Лесной горох	6	13	3	23,08%
Чистотел	5	22	10	45,45%

Табл. 2. Результаты тестирования ПК для распознавания фотографических изображений пылевых зерен методом потенциальных функций

Наименование пылевого зерна	Обучение	Тестирование	Правильно распознанные	В %
Борщевик	5	26	12	46,15%
Вереск	6	30	11	36,67%
Земляника	5	22	5	22,73%
Иван чай	5	23	7	30,43%
Подсолнух	7	34	18	52,94%
Гвоздика	7	32	24	75,00%
Вьюнок	8	27	12	44,44%
Лесной горох	6	13	3	23,08%
Чистотел	5	22	6	27,27%

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

Хорошие результаты программный комплекс показал при распознавании пыльцевых зёрен иван-чая, борщевика, подсолнечника. Пыльцевые зерна земляники дали самые плохие результаты.

Недостаточно хорошие результаты при использовании классических методов распознавания образов связано с тем, что материалов для обучения было недостаточно, в то время, как апертуры некоторых пыльцевых зерен весьма схожи. Кроме того, следует отметить, что положение пыльцевых зерен меняется, и на фотографических изображениях вид апертуры одного и того же зерна в значительной степени отличается. Следует учитывать также и наложение одной пыльцы на другую, поэтому, необходимо решать проблему восстановления изображения пыльцевого зерна по фрагменту.

Архитектурные принципы построения программного комплекса для распознавания пыльцевых зерен

Фотографические изображения пыльцевых зерен последовательно обрабатываются модулями предварительной обработки и модулем распознавания. Модуль распознавания на настоящий момент реализует алгоритмы распознавания с применением методов потенциальных функций и лингвистического метода. Далее, в зависимости от настроенных параметров, осуществляется статистическая обработка результатов распознавания. ПК может эксплуатироваться под управлением Windows, MacOS и систем Linux. Механизм плагинов позволяет сделать ПК открытым и легко настраиваемым для применения других методов предобработки изображений. Кроме того, ПК способен выполнять распознавание и других фотографических изображений, например, изображений для поиска патологий при медицинских исследованиях.

Выводы

Проведенные исследования показали, что классические методы распознавания фотографических изображений зерен пыльцы дают очень средние результаты. В статье указываются возможные причины неудач. По этой причине в настоящее время ведется разработка программных средств, реализующих нейронные сети. Цельсообразным представляется также необходимость использования других метрик, например, метрики Махаланобиса.

Проблемы применения классических методов распознавания для фотографических зерен пыльцы

Список источников

1. Новоселова, Л.В. Изменчивость люцерны хмелевидной / Л.В. Новоселова, М.А. Данилова // Вестн. Перм. ун-та. 2001. - Вып. 4: Биология. - С. 22-26.
- 2.Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход.: Вильямс, 2004.
- 3.Павлидис Т. Алгоритмы машинной графики и обработки изображений. М.: Радио и связь, 1986. – 394 с.
- 4.Фу К. Структурные методы в распознавании образов. -М.: Мир, 1977. -319 с.
- 5.Аркадьев А.Г., Браверман Э.М. Обучение машины классификации объектов 1971. 192 с.

Метод классификации объектов различных классов на изображениях

Роман Захаров

СГАУ имени академика С.П. Королёва, Самара, Россия.
roman.zakharovp@yandex.ru

Аннотация. Статья посвящена вопросу классификации и распознавания объектов различных классов, как на изображениях. Исследуется предлагаемый алгоритм, основанный на Histogram of Oriented Gradients (HOG) и Local Binary Patterns (LBP). Описана схема решения задачи для объектов различных классов. Представлены результаты вычислительных экспериментов.

Ключевые слова: распознавание, классификация, машина опорных векторов, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), PCA.

Введение

В настоящее время для решения многих практических задач используются системы компьютерного зрения (системы видеонаблюдения, системы помощи водителю и другие). В работе рассматривается задача распознавания и классификации объектов разных классов, таких как (мотоциклы, автомобили, люди и др.) на изображениях с использованием классификатора SVM. Представлены результаты вычислительных экспериментов на базе данных изображений PASCAL Visual Object Challenge-2007 (VOC2007, <http://pascallin.ecs.soton.ac.uk/challenges/VOC>).

Важной частью распознавания и локализации объекта является выбор признакового описания объекта. В области компьютерного зрения разработано множество дескрипторов для описания изображений HOG[1], LBP[3]. В работе представлена схема формированию дескриптора основанного на комбинации дескриптора HOG и LBP

Постановка задачи

Общая постановка задачи распознавания образов следующая. Предполагается, что имеется M изображений каждого из K объектов. Каждое изображение представляется вектором $x = [x_1, x_2, \dots, x_N]^T$ размерности N , где x_1, x_2, \dots, x_N – признаки. Векторы, соответствующие изображениям одного объекта, составляют класс. Совокупность векторов признаков всех классов образует обучающую выборку. Решение задачи распознавания состоит в конструировании решающей функции $f: R^N \rightarrow \{0, 1, \dots, K\}$, которая каждому вектору x ставит в соответствие некоторый класс. Для уменьшения числа неправильных классификаций вводится также класс с номером 0, соответствующий отказу в распознавании.

Качество распознавания, зависит от выбора системы признаков. Наряду с выбором системы признаков большую роль играют также используемая при распознавании мера близости и построенное на ее основе решающее правило.

Для решения задачи классификации объектов на изображении широко используется метод машины опорных векторов, а в качестве признаков наиболее популярными являются гистограммы ориентированных градиентов (HOG)[1], также часто используют метод Viola-Jones, который показывает хорошие результаты для детектирования человеческих лиц в кадре, и различные методы, основанные на комбинировании методов.

В настоящей работе ставится задача провести исследования метода формирования дескриптора основанного на информации о форме объекта и дескриптора основанного на текстурных характеристиках объекта. Дескрипторы HOG и LBP являются довольно распространенными дескрипторами для локализации и классификации объектов на изображениях.

Новизна работы заключается в формировании метода для получения дескриптора, основанного на дескрипторах HOG и LBP. Для построения дескрипторов используется несколько подходов – это как объединение дескрипторов и извлечение из них более важной информации, так и получение правила, на основе которого выбирается тот или иной дескриптор в различных ситуациях. Классификация происходит с по-

мощью метода основанного на машине опорных векторов на сформированных дескрипторах.

Описание алгоритма

Предлагаемый метод основан на комбинации двух дескрипторов - это дескриптора основанного на информации о форме объекта, и дескриптора составленного с использованием локальных бинарных признаков на изображении. На (рис. 1) изображена общая схема классификации для предлагаемого метода

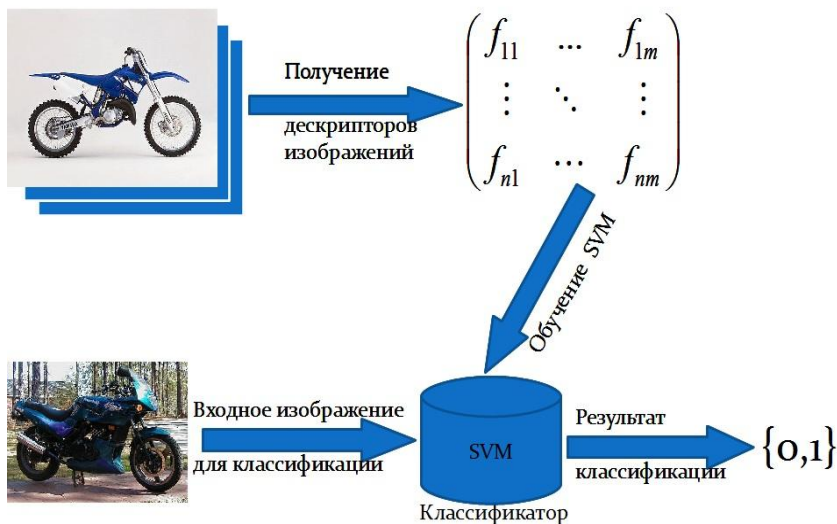


Рис. 1. Общая схема классификации на основе SVM для распознавания изображений и классификации изображений

Далее опишем общие схемы для классификации объектов. В данной работе исследовалось два подхода. Первый это извлечение более существенной информации из объединенных дескрипторов и второй это построение адаптивного алгоритма. Далее рассмотрим последовательно каждый из методов, опишем их схему работы и кратко рассмотрим формирование самих дескрипторов HOG и LBP.

Метод сокращения размерности для дескрипторов HOG и LBP

Для сокращения размерности был выбран метод (PCA[2]). Метод PCA (Principal Component Analysis) анализирует дескрипторы и выделяет более информативные их части. Ниже на (рис. 2) приведём схему алгоритма.

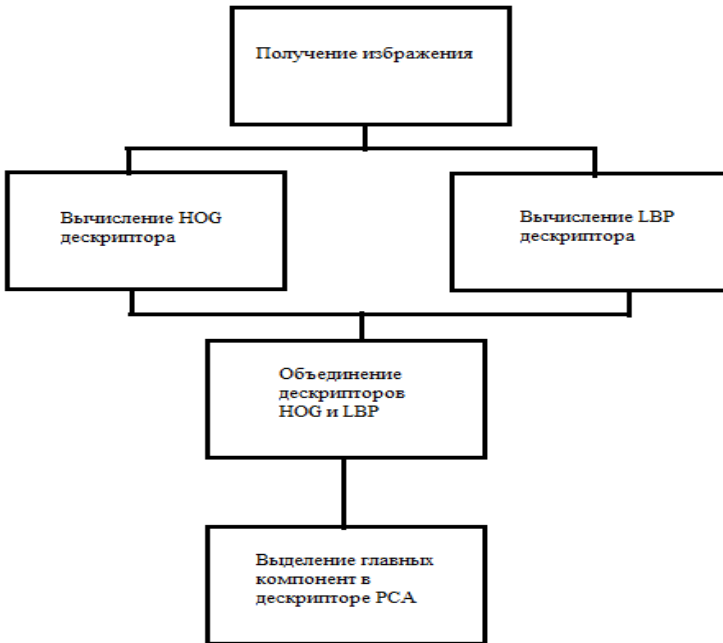


Рис. 2.Схема получения нового дескриптора на основе дескриптора HOG и LBP с применением метода сокращения размерности дескриптора PCA

Ниже представлены основные шаги для выделения главных компонент в дескрипторах.

1. Подготовка данных

На этом шаге обучающая выборка дескрипторов должна быть представлена в виде вектора G_i где i это номер дескриптора.

2. Вычитание среднего из выборки

Вычисляется средний дескриптор из выборки, затем он вычитается из выборки Γ . Для дескрипторов HOG и LBP применяется одна, Евклидова метрика, с помощью которой находится средний вектор

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (1)$$

$$\Phi_i = \Gamma_i - \Psi \quad (2)$$

3. Вычисление ковариационной матрицы

На этом шаге ковариационная и информационная матрица C и L соответственно вычисляется следующим образом

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T \quad (3)$$

$$L = \frac{1}{M} \sum_{i=1}^M \Phi_i^T \Phi_i \quad (4)$$

4. Выбор главных компонент

Вычисляются собственные числа и вектора (v) для матриц C и L . Было доказано, что не нулевые собственные числа этих матриц совпадают. Размерность матрицы L много меньше размерности матрицы C , это позволяет вести обработку на матрице меньшей размерности, что повышает эффективность метода и программы в целом.

5. Собственные дескрипторы

Собственные дескрипторы вычисляются следующим преобразованием собственных векторов информационной матрицы

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k \quad (5)$$

Для распознавания можно взять первые M' собственных дескрипторов, расположенных по убыванию соответствующих им собственных значений. Фактически это размерность базиса, на который мы будем проецировать другие вектора. Значение M' выбирается экспериментальным путём.

Применение собственных дескрипторов для дальнейшего распознавания

Нормируем каждый собственный дескриптор и тогда получим, что собственные дескрипторы образуют ортонормированный базис в пространстве дескрипторов. Будем проецировать на этот базис все дескрипторы, и получать координаты в новом базисе следующим образом

$$\omega_k = u_k^T (\Gamma_{new} - \Psi), \quad \text{где } k = 1 \dots M' \quad (6)$$

$$\Omega_{new}^T = [\omega_1, \omega_2, \dots, \omega_{M'}] \quad (7)$$

Далее полученный дескриптор переходит на вход к классификатору.

Адаптивный алгоритм для классификации

Адаптивный алгоритм формируется с использованием весовых коэффициентов, которые присваиваются отдельным методам, то есть для гистограммы ориентированных градиентов и метода LBP.



Рис. 3. Схема работы адаптивного алгоритма на основе дескриптора HOG и LBP

Получаем правило следующего вида

$$precision = w * precision_{HOG} + (1 - w) * precision_{LBP} \quad (8)$$

где: *precision* – оценка распознавания адаптивного алгоритма;

Метод классификации объектов различных классов...

$precision_{HOG}$ – оценка распознавания машины опорных векторов с использованием дескриптора HOG;

$precision_{LBP}$ – оценка распознавания машины опорных векторов с использованием дескриптора LBP;

w – весовой коэффициент, который подбирается экспериментальным путём и изменяется в интервале от 0 до 1.

На (рис. 3) показана схема для классификации адаптивным методом.

Дескрипторы HOG и LBP

Для формирования адаптивного алгоритма и алгоритма основанного на сокращении размерности были использованы стандартные алгоритмы компьютерного зрения для извлечения дескрипторов из изображения. Это дескрипторы HOG[1] и LBP[3].

Описание проведения экспериментов

Для проведения экспериментального исследования эффективности, предложенного метода, была выбрана база данных PASCAL Visual Object Challenge-2007 (VOC2007, <http://pascallin.ecs.soton.ac.uk/challenges/VOC>) содержащая более 5000 тестовых изображений, разделённых на 20 различных классов. Данная выборка содержит изображения различных объектов с разными разрешениями при различной освещённости, разного размера и снятых с различных ракурсов. На (рис. 4) представлена выборка тестовых изображений.



Рис. 4. Примеры тестовых изображений из базы данных PASCAL VOC-2007

Для тестирования будем оценивать точность(аккуратность) алгоритма. Точность оценивается как отношение правильно распознанных изображений к общему числу изображений в тестовой выборке.

Ниже представлена формула для вычисления *Accuracy*

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (9)$$

где: *tp* – основной класс, классифицированный как основной класс, то есть когда классификатор сработал правильно;
tn– неосновной класс, классифицированный как неосновной класс, то есть когда классификатор сработал правильно;
fp– неосновной класс, классифицированный как основной класс, то есть классификатор сработал неправильно;
fn – основной класс, классифицированный как неосновной класс, то есть когда классификатор сработал неправильно.

Также для тестирования будем оценивать две различные ошибки:

1. Ошибки первого рода
2. Ошибки второго рода

Ошибки первого рода – это вероятность принятия основного класса за вторичный. То есть вероятность «промаха», когда основной класс будет пропущен. Ошибки второго рода – это вероятность принять вторичный класс за основной. То есть это вероятность «ложного срабатывания», когда за искомый знак, будет принят другой знак, это ещё называется доля ложных положительных классификаций (False Positive Rate, FPR).

Результаты экспериментов

Приведём (табл. 1) и (табл. 2) с результатами тестирования исследуемых подходов. В таблицах сравнивается точность(*Accuracy*) алгоритмов, для исследуемых методов.

Табл. 1. В таблице показана точность распознавания методов основанных на дескрипторе HOG, LBP и на объединённом дескрипторе на группах классов с различными объектами

Метод распознавания/ Класс объекта	LBP	HOG	Объединённый дескриптор LBP и HOG
aeroplane	0.958	0.871	0.938
bicycle	0.951	0.698	0.851
Bird	0.943	0.657	0.716
boat	0.965	0.694	0.912
bottle	0.957	0.672	0.816
bus	0.964	0.589	0.804

car	0.797	0.534	0.605
cat	0.934	0.550	0.802
chair	0.431	0.527	0.721
cow	0.974	0.799	0.928
diningtable	0.961	0.652	0.887
dog	0.915	0.622	0.701
horse	0.944	0.715	0.865
motorbike	0.955	0.707	0.842
person	0.594	0.518	0.511
Potted plant	0.954	0.566	0.737
sheep	0.980	0.895	0.980
sofa	0.954	0.697	0.827
train	0.947	0.508	0.899
tv monitor	0.953	0.774	0.691

Табл. 2. В таблице показана точность распознавания адаптивного алгоритма и метода построенного с применением метода главных компонент на группах классов с различными объектами

Метод распознавания/ Класс объекта	Адаптивный алгоритм	ЛВРНОГ с применением PCA
aerosplane	0.958	0.938
bicycle	0.951	0.851
Bird	0.943	0.710
boat	0.965	0.912
bottle	0.957	0.802
bus	0.964	0.804
car	0.797	0.703
cat	0.934	0.802
chair	0.527	0.721
cow	0.974	0.908
diningtable	0.961	0.887
dog	0.915	0.701
horse	0.944	0.865
motorbike	0.955	0.842
person	0.594	0.511
Potted plant	0.954	0.799
sheep	0.980	0.980
sofa	0.954	0.827

train	0.947	0.899
tv monitor	0.953	0.791

Выводы

1. Предложены две схемы формирования дескрипторов адаптивная и с использованием метода PCA.
2. Основное достоинство предлагаемого метода это улучшение качества классификации объектов в адаптивном алгоритме, по сравнению с методами построенными только на дескрипторах HOG и LBP.
3. Из результатов видно, что в качестве слагаемых адаптивного метода следует взять и объединённый дескриптор и дескриптор на основе LBP, как дескрипторы, показывающие наиболее лучшие результаты, а дескриптор на основе HOG исключить.

Список источников

1. Dalal, N. Histograms of Oriented Gradients for Human Detection. / N. Dalal, W. Triggs // IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05. – 2005. – Vol. 1(3). – P. 886-893.
2. A.A. Miranda, Y.-A. Le Borgne, and G. Bontempi New Routes from Minimal Approximation Error to Principal Components, Volume 27, Number 3 / June, 2008, Neural Processing Letters, Springer
3. Pietikäinen, M., Hadid, A., Zhao, G. and Ahonen, T. (2011), Computer Vision Using Local Binary Patterns, Springer. <http://www.springer.com/mathematics/book/978-0-85729-747-1>. This book presents a detailed description of LBP methods and their variants, and provides an overview as how texture methods can be used for solving different kinds of computer vision problems.

«Бизнес Семантика»: практика интеграции информационных систем с использованием семантических технологий

Сергей Горшков

«Бизнес Семантика», Екатеринбург, Россия. serge@business-semantic.ru

Аннотация. В статье рассматриваются практические аспекты реализации обмена данными между информационными системами при помощи продукта «Бизнес Семантика». Приводится пример использования этого продукта в рабочей среде.

Ключевые слова: Semantic Web; семантические технологии; семантическая интеграция; обмен данными.

Введение

Проблема интеграции информационных систем, особенно в компаниях среднего масштаба, является одним из наиболее острых и тяжело решаемых вопросов построения ИТ-инфраструктуры. В этом специалисты нашей компании имели возможность убедиться, выполнив около ста внедрений систем, предназначенных для управления взаимоотношениями с клиентами (CRM), ресурсами предприятия (ERP), управления проектами, документооборота. Предлагаемый в настоящее время на рынке набор программных средств интеграции лишь в небольшой степени отвечает реальным потребностям компаний. Одни средства, такие как MDM-системы, слишком дороги и сложны во внедрении, а получаемые с их помощью результаты ограничены; другие средства, такие как построение сервисно-ориентированной архитектуры SOA, требуют

Практика интеграции информационных систем с применением семантических технологий

слишком большого объема работ проектировщиков и программистов, по созданию и поддержке сервисов интеграции.

Нашей задачей при создании продукта «Бизнес Семантика» было предоставление инструмента, способного обеспечить высокую эффективность интеграции (работу с различными типами, большими объемами информации), при минимальных затратах на программирование и настройку системы при каждом внедрении. Для достижения таких характеристик необходимо было заложить в продукт технологический принцип, являющийся синтезом лучших существующих практик интеграции, а также несущий существенный элемент новизны. Таким принципом стало использование семантических технологий, для кодирования информации, передаваемой между информационными системами.

В деталях принцип архитектуры системы «Бизнес Семантика» был представлен в нашей статье «Интеграция информационных систем с применением семантических технологий». В этой статье мы опишем некоторые аспекты практической реализации продукта, и приведем пример его использования в реальном внедрении.

Процесс запуска интеграции при помощи системы «Бизнес Семантика»

Система «Бизнес Семантика» состоит из двух типов программных компонентов: сервера, который управляет маршрутизацией данных, передаваемых между интегрируемыми информационными системами (далее ИС), и клиентских модулей, которые встраиваются в ИС, и/или взаимодействуют с их базами данных. Способ подключения клиентского модуля к ИС зависит от архитектуры самой клиентской системы: в большинстве случаев есть возможность использовать стандартный клиентский модуль, который поддерживает подключение к наиболее популярным базам данных – Oracle, PostgreSQL, MySQL. В этом случае вмешательства в программный код самой клиентской ИС не требуется. В других случаях, например, при интеграции с системами на платформе IC, компоненты программного кода клиентских модулей «Бизнес Семантика» необходимо встраивать в структуру самой системы. Так, в случае с IC, часть кода размещается в модулях конфигурации IC, а часть находится в СОМ-объектах.

Общая структура серверных и клиентских компонентов показана на рис. 1. Для наглядности, в качестве примера интегрируемых ИС выбраны веб-приложение с базой данных MySQL, и продукт на платформе IC.

Практика интеграции информационных систем с применением семантических технологий

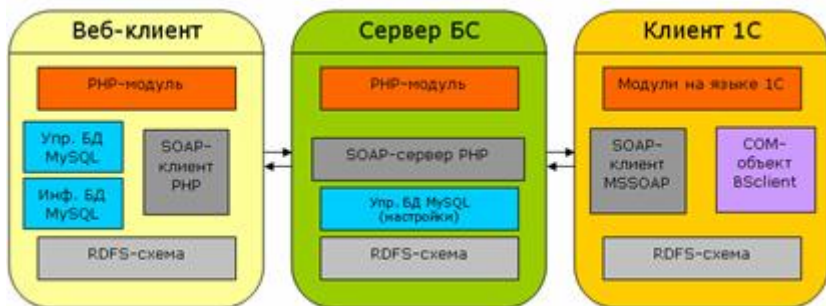


Рис. 1. Серверные и клиентские компоненты системы «Бизнес Семантика»

Для запуска процесса интеграции необходимо выполнить следующие шаги.

1. **Составить онтологию.** Результатом этого шага является файл в синтаксисе RDFS, который загружается в модуль настроек сервера. Варианты этого файла (возможно, усеченные) загружаются в настройки клиентских модулей.

2. **Настроить сервер.** Необходимо зарегистрировать в нем клиентские системы, и определить их права доступа (какие виды объектов и их свойств сможет читать, изменять и удалять каждая система).

3. **Настроить клиентские модули.** Важнейший шаг этой настройки - установка соответствия между элементами онтологии и полями базы данных этой информационной системы. Обычно эта операция выполняется при помощи визуального интерфейса. Для тех объектов и свойств схемы, которым нельзя однозначно сопоставить какие-либо элементы базы данных, придется определить специальные обработчики, и подключить их к клиентскому модулю при помощи простого программного интерфейса. Обработчики создаются на языке, естественном для данной программной среды.

После этого процесс обмена данными можно запустить.

Обмен данными между информационными системами в процессе взаимодействия

Общая схема процесса обмена выглядит так.

1. Клиентские модули отслеживают изменения в данных, которые подлежат передаче другим ИС. При возникновении таких событий они помещаются в очередь отправки. С достаточно большой частотой (один раз в 1-2 минуты) очередь отправки обрабатывается, и информация обо

Практика интеграции информационных систем с применением семантических технологий

всех изменившихся объектах отправляется серверу «Бизнес Семантика». Передаваемые данные записаны в синтаксисе RDF.

2. Сервер осуществляет маршрутизацию полученных сообщений, в соответствии с правами доступа, помещая их в очередь отправки для соответствующих ИС. При этом происходит ряд обработок – выявление и слияние дублей, первичное присвоение URI новым информационным объектам, контроль и восстановление целостности данных.

3. Клиентские ИС получают от сервера сообщения об изменениях в тех данных, в которых они заинтересованы. Они выполняют преобразование (интерпретацию) полученных данных, и сохраняют их в своих внутренних информационных структурах.

Пример использования продукта «Бизнес Семантика» в рабочей среде

Проект реализован в компании "Росэнерготранс", входящей в группу "Свердловэлектро". Профиль деятельности предприятия - разработка трансформаторов и другого оборудования для электроподстанций. Менеджеры по продажам используют в качестве основной рабочей среды систему index.CRM, основанную на сервере баз данных MySQL. В нее заносятся заявки на обсчет и проектирование различных вариантов исполнения трансформаторов, которые менеджеры предлагают заказчикам. Проектировщики, выполняющие расчеты, работают в другой информационной системе, "Заказчик". Эта система основана на БД Oracle. Задачей интеграции является обеспечение документооборота заявок. На приведенном ниже рис. 2 показана схема взаимодействия систем.



Рис. 2. Среда обмена информацией

Процесс работы с заявками на расчет стоимости оборудования выглядит так. Менеджер создает документ "Заявка" в CRM-системе, прикрепляет к нему необходимые файлы. Документ поступает в систему "Заказчик", и передается исполнителю. Исполнитель выполняет расчет,

Практика интеграции информационных систем с применением семантических технологий

прикрепляет к заявке результаты расчета, и отправляет ее на утверждение руководителю и другим лицам. Менеджер, через CRM-систему, отслеживает состояние обработки заявки. Для этого ему доступен просмотр всех событий, которые происходят с заявкой в системе "Заказчик". После утверждения результатов расчета, заявка возвращается менеджеру, который продолжает работу с ней в CRM-системе.

Диаграмма обмена информационными объектами показана на рисунке ниже. В данном случае их три: документ "Заявка", прикрепленные к нему файлы, и события, происходящие с заявками.

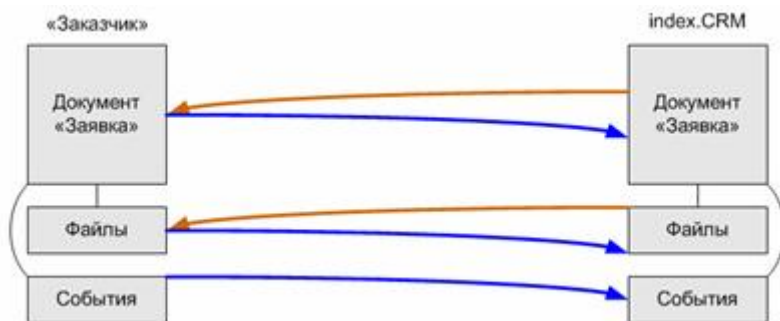


Рис. 3. Диаграмма процесса обмена информацией

Обмен информацией проходит в прозрачном для пользователя режиме. Задержка между внесением данных в одной системе, и их отображением в другой, составляет не более двух минут (возможна и более высокая частота обновления, однако в целях данного проекта она является достаточной). Пользователь никак не взаимодействует с программными компонентами, осуществляющими обмен данными, работая только в интерфейсе того или иного прикладного программного продукта.

Кроме того, преимущества использования «Бизнес Семантики» состоят в легкости перенастройки обмена в случае необходимости, простоте включения в обмен других ИС, отсутствии вмешательства в программный код клиентских систем, сохранении их структуры данных.

Выводы

- 1. Продемонстрированы технические особенности, положенные в основу нового продукта для интеграции информационных систем – «Бизнес Семантика».*
- 2. Представлен способ использования этого продукта в конкретном внедрении.*

Анализ статистических алгоритмов снятия морфологической омонимии в русском языке

Е. Д. Лакомкин¹, И. В. Пузыревский², Д. А. Рыжова³

¹egor.lakomkin@gmail.com, ²ivan.pouzyrevsky@gmail.com,
³daria.ryzhova@mail.ru

НИУ ВШЭ, Москва, Россия
компания Яндекс, Москва, Россия

Аннотация. Статья посвящена анализу работы широко известных в англоязычной литературе статистических алгоритмов POS-теггинга, основанных на скрытой марковской модели (НММ, [9]) и марковской модели максимальной энтропии (МЕММ, [8]), в применении к задаче морфологической дизамбигуации в русском языке. В работе описан эксперимент, результаты которого показывают, что в целом эти модели демонстрируют эффективность POS-теггинга, близкую к результатам применения данных моделей к материалу английского языка. Однако их точность в применении к задаче дизамбигуации по расширенному набору грамматических характеристик не очень высока.

Ключевые слова: морфологическая омонимия; алгоритм; скрытая марковская модель; марковская модель максимальной энтропии.

1 Введение

Морфологическая разметка текста — это процесс определения для каждой его словоформы начальной формы (леммы) и грамма-

тических характеристик, таких как частеречная принадлежность, род, падеж, число, лицо, наклонение, время, степень сравнения и др. Значения грамматических признаков, приписываемые словоформе, обычно называют грамматическими тегами.

Многие задачи в сфере автоматической обработки естественного языка опираются на обширные корпуса морфологически размеченных текстов. Так, например, именно такие тексты необходимы для работы систем машинного перевода; морфологическая разметка используется при семантическом аннотировании и является важной составляющей синтаксического анализа. Получение большого объёма морфологически размеченных текстов вручную – задача крайне трудоёмкая, поэтому обычно для разметки текстов используют заранее сконструированные морфологические анализаторы (такие, как *Mystem*¹, *Pymorphy*² и др.). Однако автоматические разметчики, как правило, приписывают слову не единственный разбор, а все теоретически возможные. Ср., например, варианты морфологического разбора, которые выдаёт *Mystem* для словоформы *мыла*:

lex="мыль"gr="V,act,f,indic,ipf,norm,praet,sg,tran"

lex="мыло"gr="S,inan,n,nom,norm,pl"

lex="мыло"gr="S,gen,inan,n,norm,sg"

lex="мыло"gr="S,acc,inan,n,norm,pl"

В результате работы анализатора практически каждая словоформа получает больше одного разбора. Возникает необходимость в устранении неоднозначности: в выборе единственного варианта анализа, правильного для данного контекста. Так, например, в предложении *Мама мыла раму* для словоформы *мыла* подходит только один разбор: первый.

Процесс выбора одной группы тегов из нескольких возможных называют дизамбигуацией или снятием омонимии, и часто он тоже осуществляется автоматически. Для английского языка, как для языка с бедной морфологией, задача снятия морфологической омонимии сводится, как правило, к проблеме разрешения многозначности на уровне частей речи (так называемого POS-теггинга). При этом используются алгоритмы, основанные на статистических моделях, учитывающие вероятность появления тега той или иной части речи в данном контексте. Для английского языка эти алгоритмы работают достаточно хорошо и обычно демонстрируют не менее 96%

¹<http://company.yandex.ru/technologies/mystem/>

²<http://pymorphy.readthedocs.org/en/v0.5.6/>

точности, ошибаясь лишь в 4% случаев (см., например, [1], стр. 4; а также [8]).

Для русского языка картина будет иная сразу по нескольким причинам. Во-первых, морфологическая омонимия в русском языке, как это видно по примеру, приведённому выше, не сводится к омонимии частеречной, а охватывает большое количество различных грамматических признаков. Во-вторых, хорошая работа статистических моделей на материале английских текстов объясняется тем, что в английском языке фиксированный порядок слов. Это обстоятельство упрощает создание модели, так как позволяет, к примеру, опираться только на локальный контекст слова (соседние слова) без учета дальних зависимостей. Именно поэтому для морфологической дизамбигуации в английском языке часто успешно используются алгоритмы, основанные на марковских моделях и учитывающие зависимость каждого набора тегов только от одного элемента контекста – непосредственно предшествующего ему набора тегов.

В русском языке, напротив, порядок слов свободный, так что предполагается, что количество возможных контекстов из-за этого увеличивается и эффективность обучения простой модели, основанной на локальных зависимостях, снижается. Поэтому, наряду с марковскими моделями, для снятия морфологической омонимии в русском языке используются более сложные статистические модели (ср., например, [2]) или гибридные системы, в которых статистика дополняется набором правил (см., например, Transformation-Based Learning [4], а также [1]).

Цель данного исследования — определить экспериментально, действительно ли такие относительно простые вероятностные модели, как скрытая марковская модель (НММ) и марковская модель максимальной энтропии (МЕММ), при решении проблемы морфологической дизамбигуации в русском языке дают результат, существенно отличающийся от результата работы этих же алгоритмов с текстами на английском языке. Мы ставим перед собой следующие задачи:

- определить, какой процент ошибок допускают данные алгоритмы при снятии частеречной омонимии в русскоязычных текстах;
- оценить, применимы ли рассматриваемые алгоритмы для решения более широкой задачи снятия морфологической омонимии на уровне сразу нескольких грамматических категорий;
- проверить, возможно ли существенно изменять эффективность работы алгоритмов за счёт корректировки обучающей выборки и параметров обучения (например, повлияет ли на

качество работы системы изменение набора тегов частей речи: объединение одних классов и, наоборот, разделение других классов слов).

Таким образом, цели и задачи нашего исследования сближают нашу работу с исследованием С. Ю. Толдовой и А. В. Сокирко (см. [3]), где оценивается возможность применения скрытой марковской модели к задаче морфологической дизамбигуации. Однако новизна нашей работы заключается в том, что мы изучаем возможности двух различных статистических моделей, сравнивая их между собой. При этом мы используем свою реализацию НММ и не проводим сравнения наших результатов с результатами, полученными Сокирко и Толдовой, так как их сложно сравнить напрямую: в работе ([3]), в отличие от нашей, допускалось неполное разрешение неоднозначности.

2 Условия эксперимента

Корпус. Наше исследование выполнено на материале Национального корпуса русского языка (далее НКРЯ), а точнее, его подкорпуса — текстов со снятой морфологической омонимией (около 6 млн словоупотреблений). На этом корпусе мы проводили и обучение, и тестирование исследуемых моделей. Чтобы не тестировать алгоритм на тех же текстах, на основе которых проводилось обучение, но в то же время уменьшить ошибку, связанную с делением корпуса на обучающую и оценочную выборки, мы использовали метод кросс-валидации.

Морфологический анализатор. В качестве морфологического анализатора мы использовали систему *Mystem*: этот инструмент максимально близок к формату НКРЯ и находится в открытом доступе. Под близостью форматов мы подразумеваем степень близости инвентарей грамматических тегов и способов их обозначения. Так, например, анализатор *Rumorphu* использует систему тегов, основанных на русскоязычных названиях граммем и обозначаемых кириллическими символами (ср: П - прилагательное, мр – мужской род, им – именительный падеж), в то время как в НКРЯ используются сокращения от латинских названий грамматических признаков, записываемые латинскими буквами (ср.: А – adjective, m – masculine, nom – nominative). Помимо этого, сам набор грамматических признаков, входящих в разметку формата *Rumorphu*, отличается от стандарта, принятого в НКРЯ.

Морфологическая разметка. Формат разметки, принятый в системе *Mystem*, в целом соответствует формату НКРЯ. В нём теги также основываются на латинских названиях граммем, а набор грамматических признаков, включаемых в морфологическую характеристику лексемы, почти полностью повторяет набор, используемый в НКРЯ. Однако и здесь есть некоторые расхождения: в общем случае они связаны с тем, что разметка НКРЯ немного более детальная, чем разметка *Mystem*'а.

На основе анализа выдачи морфологического анализатора (*Mystem*'а) и «золотого стандарта» (НКРЯ) мы унифицировали теги и провели две серии экспериментов. В первой серии мы оценивали качество работы алгоритмов для задачи снятия только частеречной омонимии. При этом использовался набор частей речи, приведенный в приложении, и соответствующих им тегов.

В разметке НКРЯ используется также тег PRAEDIC-PRO (для обозначения местоимений-предикативов *некого, нечего*), однако в инвентаре тегов *Mystem*'а он не представлен, а случаи, которые помечаются этим тегом в НКРЯ, в разметке *Mystem*'а обычно получают тег PRAEDIC, поэтому теги PRAEDIC и PRAEDIC-PRO были склеены нами в один – PRAEDIC.

Вторая серия экспериментов была проведена с целью измерить качество работы алгоритмов морфологической дизамбигуации для расширенного набора грамматических признаков, включающих, помимо частей речи, граммемы падежа (nom, gen, gen2, dat, acc, ins, loc, loc2), числа (sg, pl), рода (m, f, n), лица (1p, 2p, 3p), времени (praes, praet, fut) и наклонения (indic, imper). Используемые в разметке НКРЯ и не представленные в разметке *Mystem*'а дополнительные падежные теги (acc2, dat2, adnum и voc) были удалены (dat2 и voc) или склеены с другими тегами по следующим правилам: acc2 => nom, adnum => gen. Тег граммемы общего рода (m-f), включённый в формат НКРЯ, но отсутствующий в инвентаре *Mystem*'а, был также удалён.

Таким образом, в первой серии экспериментов набор тегов для каждой словоформы сводился к одному тегу (части речи), а во второй – к целой группе грамматических признаков, т.е. словоформы типа *стол* или *красивой* попадали в группу однозначных в первой серии экспериментов и неоднозначных – во второй. Предполагалось, что в первом случае алгоритмы будут показывать более высокие результаты, так как они будут работать и с меньшим количеством неоднозначных слов, и с меньшим количеством факторов, влияющих на выбор тега. Однако было важно определить, насколько суще-

ственно результаты работы алгоритмов в применении к проблеме снятия частеречной омонимии будут отличаться от результатов работы тех же алгоритмов, но в применении к более сложной задаче полной морфологической дизамбигуации.

Помимо этого, для каждой серии экспериментов мы провели дополнительный подсчёт точности работы алгоритмов в условиях модифицированного набора тегов частей речи. Мы выделили в качестве отдельных частей речи полные причастия и деепричастия, так как дистрибутивно они отличаются от глаголов, а также отделили краткие прилагательные и краткие причастия от полных. И, напротив, предикативы мы объединили с классом наречий.

Оценка качества работы алгоритма. Качество работы алгоритма оценивалось по его точности, вычисляемой следующим образом: разметка, полученная в результате работы системы Mystem и последующего подключения одного из алгоритмов дизамбигуации, сравнивалась с «золотым стандартом» – исходным набором размеченных текстов со снятой омонимией. Затем считался процент совпадений полученных разборов с разборами НКРЯ, который и является мерой точности работы алгоритма. Для борьбы со смещенностью оценки из-за обучающей выборки была применена кросс-валидация с 5 корзинами; для каждого алгоритма считалось среднее значение точности и стандартное отклонение. При оценке качества работы алгоритма не учитывались слова, помеченные в исходном «золотом» корпусе как инициалы (init), аббревиатуры (abbr) или цифры (ciph). Не принимались во внимание также случаи, когда одному слову в НКРЯ соответствовало два слова в разборе Mystem'a (например, *бело-кремовый* vs. *белый* и *кремовый*).

При оценке точности работы алгоритмов мы старались оценить степень влияния на результат качества работы морфологического анализатора. Для этого мы, во-первых, определили верхнюю оценку на возможное качество дизамбигуатора: посчитали процент случаев, когда ни один из разборов, предложенных Mystem'ом, не является правильным (не совпадает с разбором из «золотого стандарта»), а значит, и наша статистическая модель заведомо не может выбрать верный набор тегов. Во-вторых, мы посчитали отдельно качество работы анализатора со знакомыми (представленными в словаре) и незнакомыми словами (такими, которых в словаре Mystem'a нет): ясно, что процент ошибок в разборе незнакомых слов оказывается существенно ниже общих показателей, а значит, и дизамбигуатор будет демонстрировать в таких случаях более низкие результаты,

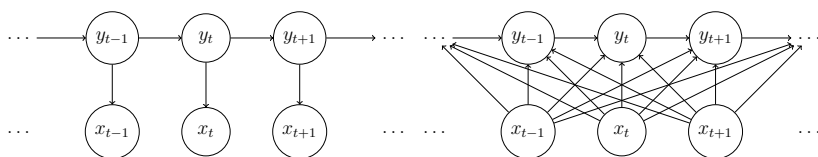


Рис. 1. НММ

Рис. 2. МЕММ

негативно влияющие на общую статистику и нарушающие чистоту эксперимента.

В качестве нижней границы, с которой можно было бы сравнить полученные нами результаты и оценить работу алгоритмов НММ и МЕММ, мы использовали точность работы «наивного» алгоритма морфологической дизамбигуации, основанного на сравнении частотности разборов. Принцип работы этого алгоритма заключался в том, что словоформе всегда присваивался тот набор тегов, который по материалам обучающей выборки оказывался для неё наиболее частотным.

3 Исследуемые алгоритмы: НММ и МЕММ

В ходе исследования мы проанализировали возможности применения двух статистических моделей, используемых для задачи POS-теггинга в английском языке ([9], [8]), – скрытой марковской модели и марковской модели максимальной энтропии – к задаче морфологической дизамбигуации текстов на русском языке.

Оба алгоритма представляют собой графическую модель с условно-зависимыми случайными величинами, упорядоченными в виде двух слоев: т. н. слоя скрытых величин (или *состояний* модели) и слоя наблюдаемых величин (или *наблюдений*). Обе модели предполагают, что случайные величины скрытого слоя образуют марковскую цепь первого порядка, то есть каждое последующее состояние зависит только от предыдущего.

Наблюдаемые величины в моделях устроены по-разному. В скрытой марковской модели наблюдаемые величины зависят от скрытых, причем наблюдаемая в определенный момент времени величина может зависеть *только* от скрытого состояния, соответствующего тому же моменту времени. В марковской модели максимальной энтропии, напротив, скрытые величины произвольно зависят от наблюдаемых.

Графически модели можно изобразить в виде диаграмм на рисунках 1 и 2.

В случае задачи морфологической дизамбигуации наблюдаемыми величинами являются слова (или производные от них), а скрытыми – истинные морфологические теги слова. Для скрытой марковской модели в качестве наблюдаемых значений мы использовали трёхбуквенные окончания слов. Замена полных слов на окончания была проведена из нескольких соображений: во-первых, это позволяет существенно сократить количество наблюдаемых состояний и заметно ускорить работу алгоритма; во-вторых, такая замена позволяет собрать более полную статистику (из-за большей ее плотности) без существенного влияния на результат: в русском языке грамматическая информация, в отличие от лексической, обычно сосредоточена именно в окончании словоформы ([1]).

Сама задача разрешения омонимии сводится к вычислению последовательности наиболее вероятных значений скрытых величин:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y} | \mathbf{x})$$

В силу того, что скрытые величины образуют марковскую цепь, наиболее вероятный набор состояний модели можно эффективно вычислить с помощью алгоритма Витерби, основанного на динамическом программировании ([7], [5]).

Наконец, отметим, что в случае скрытой марковской модели этап обучения состоит в сборе по корпусу корреляционных статистик пар тегов $P(y_i|y_j)$ (т. н. *матрица перехода* между состояниями), а также статистики $P(x_k|y_i)$ (т. н. *вероятности наблюдений*). В случае марковской модели максимальной энтропии этап обучения состоит в восстановлении условного распределения $P(y_{t+1}|y_t, \mathbf{x})$, используя принцип максимальной энтропии применительно к некоторым подсчитанным по корпусу статистикам, именуемым в литературе *признаками*.

Мы использовали частотные (индикаторные) корпусные признаки, выраженные в форме:

$$f_{y,\phi}(y_t, y_{t-1}, \mathbf{x}) = \begin{cases} 1 & \text{если } y_t = y \wedge \phi(y_{t-1}, \mathbf{x}), \\ 0 & \text{иначе} \end{cases}$$

где y перебирается по множеству возможных тегов, а ϕ – *контекстный предикат*.

Такая форма позволяет более естественно формулировать признак (к примеру, «предыдущее слово – прилагательное женского рода, а текущее слово имеет разбор с грамемой женского рода») и

упростить структуру программы без различимой потери в выразимости модели ([8], [5]).

В экспериментах, представленных в настоящей статье, мы использовали следующие признаки:

- наличие у текущего слова фиксированного трехбуквенного окончания (к примеру, "-ный "-ная"; перебирались все возможные окончания),
- тег, приписанный предыдущему слову,
- наличие у текущего слова фиксированного разбора, выданного морфологическим анализатором,
- наличие предлога в окрестности текущего слова,
- согласованность по роду/числу/падежу с двумя предыдущими словами.

4 Полученные результаты

Полученные в ходе исследования результаты приведены в таблице 1.

Табл. 1. Результаты

	POS			Теги		
	Общ.	Зн.	Незн.	Общ.	Зн.	Незн.
Нижн. гр.	.8590	.8586	.8885	.6817	.6836	.5525
НММ	.9482	.9489	.8996	.8873	.8909	.6550
МЕММ	.9516	.9524	.8967	.8670	.8706	.6332
Верхн. гр.		.9895	.9081		.9741	.7017
С вынесением кратких прилагательных и глагольных форм						
Нижн. гр.	.8565	.8560	.8898	.6818	.6838	.5563
НММ	.9490	.9498	.8984	.8872	.8908	.6550
МЕММ	.9519	.9528	.8955	.8686	.8708	.6333
Верхн. гр.		.9895	.9063		.9739	.7053

Анализ результатов и ошибок. Оба алгоритма неплохо справляются с задачей частеречной дизамбигуации, но значительно хуже снимают омонимию по расширенному набору грамматических тегов.

Как правило, алгоритмы ошибаются при разметке имен собственных, местоимений, римских цифр, инициалов и сокращений. Помимо этого, модели не справляются со случаями субстантивации прилагательных и с выбором некоторых падежных форм: в первую очередь,

с разграничением между номинативом и аккузативом, что связано с особенностями порядка слов в русском языке.

Алгоритм МЕММ в целом работает лучше в применении к задаче POS-теггинга, чем НММ, что неудивительно: он позволяет учитывать большее количество факторов при обучении модели.

5 Выводы и перспективы

Таким образом, оба алгоритма решают задачу снятия морфологической омонимии в русском языке примерно на том же уровне, что и при работе с английским материалом. К тому же, точность немного меняется в зависимости от того, какой набор тегов частей речи подаётся алгоритму на вход. С задачей дизамбигуации по расширенному набору тегов, напротив, оба алгоритма справляются очень хорошо, не превышая порога точности в 90%.

В дальнейшем мы планируем провести аналогичные серии экспериментов с использованием алгоритма CRF [6], предоставляющего больше возможностей и позволяющего учесть большее количество факторов взаимовлияния словоформ внутри предложения. Далее мы планируем выбрать алгоритм, демонстрирующий наиболее высокую степень точности, отладить его работу, дополнив его при необходимости локальными правилами, сделать алгоритм независимым от конкретного морфологического анализатора и выложить полученный инструмент в открытый доступ.

Список источников

1. Ю. Зеленков, И. Сегалович, В. Титов. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог'2005». – М.: Наука, 2005. 616 с.
2. В. В. Петроченков. Морфологическая разметка русскоязычных текстов с помощью теггера на основе SVM // Информационные технологии и системы (ИТиС'12), Петрозаводск, 2012. С. 147-151.
3. А. В. Сокирко, С. Ю. Толдова. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка // Международная конференция «Корпусная лингвистика 2004». С.-Пб., 2004.
4. Brill. A simple rule-based part of speech tagger // Proceedings of the third conference on Applied natural language processing (ANLC '92).

- Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 152-155.]
5. Jurafsky, Martin. *Speech and Language Processing* // – Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.
 6. Lafferty, John D. and McCallum, Andrew and Pereira, Fernando C. N. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data* // *Proceedings of the Eighteenth International Conference on Machine Learning*. – ICML '01, 2001, pp. 282–289
 7. Manning, Schütze. *Foundations of statistical natural language processing* // – Cambridge, MA, USA: MIT Press, 1999.
 8. Ratnaparkhi. *A Maximum Entropy Model for Part-Of-Speech Tagging* // *Proceedings of the Empirical Methods in Natural Language Processing* – Philadelphia, PA, USA: 1996, pp. 133–142.
 9. Weischedel et. al. *Coping with ambiguity and unknown words through probabilistic models* // *Computational Linguistics* – Cambridge, MA, USA: MIT Press, Volume 19 Issue 2, June 1993, pp. 361–382.

Приложение

А Таблица тегов частей речи

Табл. 2. Теги частей речи

S	существительное
SPRO	местоимение-существительное
A	прилагательное
APRO	местоимение-прилагательное
V	глагол
ADV	наречие
ADVPRO	местоименное наречие
NUM	числительное
ANUM	порядковое числительное
PR	предлог
CONJ	союз
PART	частица
INTJ	междометие
PRAEDIC	предикатив
PARENTH	вводное слово

Синтаксический анализ музыкальных текстов

Ирина Голубева¹, Андрей Юрьевич Филиппович²

¹МГТУ имени Н.Э. Баумана, Москва, Россия. irinadanshina@mail.ru

²МГТУ имени Н.Э. Баумана, Москва, Россия. philippovich@list.ru

Аннотация. Статья посвящена изучению синтаксиса музыкальных текстов. Показана возможность применения лингвистических методов для проведения анализа. В качестве конкретного материала для исследований выбраны музыкальные рукописи XI–XVII веков. Для их анализа предложены типы отношений между знаками, составлена синтаксическая модель, проведен статистический анализ.

Ключевые слова: модель языка, синтаксический анализ, статистический анализ, анализ музыки, знаменные песнопения, семиография, древние рукописи

Введение

В настоящее время актуально изучение вопросов невербальной коммуникации человека. В сообщениях, которые передаются, могут быть не только текстовые составляющие, но и музыкальные, эмоциональные, жестовые. Вопросы познания, восприятия, понимания музыки получили названия музыкальные инфо-когнитивные технологии. Инфо-когнитивные технологии (ИКТ) – технологии, которые связаны с тем, как построены процессы обработки и восприятия информации человеком. Это позволяет рассматривать не только то, как расположены знаки в тексте, но и то, как они зависят друг от друга (синтаксис) и как музыка воспринимается человеком (семантика).

В качестве конкретного материала для исследований выбраны музыкальные рукописи XI-XVII веков, записанные в знаменной нотации (рисунок 1). Знаменные песнопения появились на Руси в XI веке вместе с принятием христианства. Они представляют собой музыкальные тексты, записанные с помощью специальных знаков, отличающихся по своей природе от современных нот. Основные их отличия в том, что одному знаку («знамени») может соответствовать несколько звуков и нет точного соответствия между знаком и высотой звука.

В результате реформ музыкальных нотаций был утрачен «ключ» к расшифровке знаменных мелодий, который позволяет точно перевести старинные песнопения в ното-линейную систему [7].

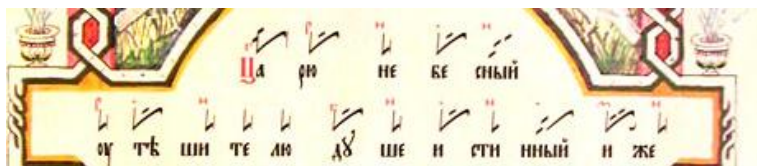


Рис. 1. Страница музыкальной рукописи в знаменной нотации

Для достоверного перевода необходимо выявлять в знаменной нотации внутренние законы, в силу которых мелодии записывались с помощью одних знамен, а не других.

Данная работа выполняется в рамках проекта «Автоматизированная система научных исследований в области компьютерной семиографии (АНСИ КС)», а также поддержана грантом РГНФ №110412025в. Было проведено исследование и выявлено, что знаменные песнопения имеют аналогичную структуру, что и естественный язык. В качестве подтверждения этого можно привести примеры аналогий между ними: Знаменный алфавит → Лексикон, Последовательность знамен → Словосочетания, предложения, Знамя (крюк) → Слово, лексема, Пометы, Признаки → Диакритические знаки. Это позволяет применять лингвистические методы для обработки и анализа песнопений.

Описанные ниже результаты получены с помощью программ SemioStatistik [3] и Semantic_Statistik [4], специально разработанных для анализа знаменных песнопений.

Дополнительная информация о проекте и проведенных исследованиях представлена в Интернете по адресу <http://it-claim.ru/semio>.

Статистический анализ

Для того, чтобы проверить гипотезу о наличии в знаменных песнопениях семиотической и синтаксической структуры, знаменная нотация была представлена как знаковая система [4].

Знамя — это графическое изображение, используемое для обозначения определенной высоты, длительности и характера исполнения. Алфавит знаменной нотации — список знамен.

Были проведены статистические исследования. На рисунке 2 приведен график распределения частот знамен для книги «Октоих», содержащей знаменные песнопения.

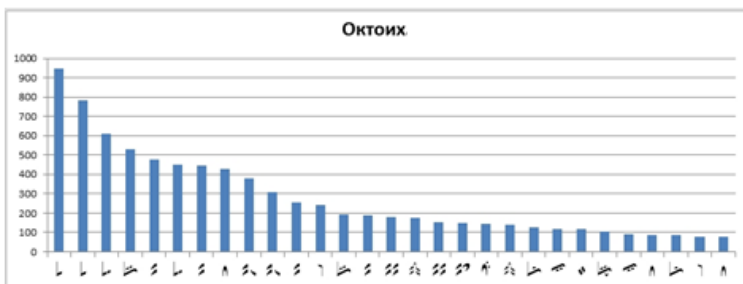


Рис.2. Частота встречаемости знамен

В результате данного этапа, было подтверждено, что знамена распределены в песнопении неравномерно, существуют «специальные» знаки для начала или окончания фразы, а также определенные сочетания знамен по 2, 3 и 4 знамени встречаются чаще, чем другие.

ⲁⲓⲛⲁ	257
ⲁⲓⲛⲁⲓ	194
ⲁⲓⲛⲁⲓⲛⲁ	174
ⲁⲓⲛⲁⲓⲛⲁⲓ	157
ⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁ	137
ⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓ	136
ⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁ	136
ⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁ	133
ⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁⲓⲛⲁ	131


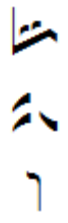
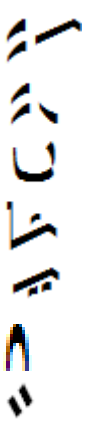
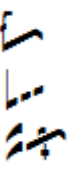
Рис.3. Наиболее частотные сочетания по три знамени

Синтаксический анализ музыкальных текстов

В дополнение было проверено, что данное распределение соответствует закону Ципфа-Мальденброта и закону Г.С. Хипса, которые справедливы для текста на естественном языке.

Последующий анализ позволил выделить группы знамен по частоте их встречаемости:

Табл. 1. Распределение знамен на 4 частотные группы

№ группы	I	II	III	IV
Число повторений каждого знамени	1000-3000	500-1000	150-500	1-150
Количество знамен в группе	6	7	30	323
Базовые знамена группы				 ...

Изучение синтаксиса

Для того, чтобы определить как связаны знамена между собой, было предложено 3 типа отношений, представленных в таблице 2.

Табл. 2. Три типа отношений в синтаксисе знаменных песнопений

Вид отношения	Описание	Пример
α – отношение: $Z_1 \xrightarrow{\alpha} Z_2$	Знамя Z_1 состоит в α – отношении со знаменем Z_2 , если Z_2 является производным от Z_1 .	
β – отношение: $Z_1 \xrightarrow{\beta} Z_2$	Знамя Z_1 состоит в β – отношении со знаменем Z_2 , если в структуре песнопения знамя Z_2 непосредственно следует за знаменем Z_1 .	
Вероятностное β – отношение: $Z_1 \xrightarrow{\beta} Z_2 (P_i)$	Если за знаменем Z_1 может следовать несколько знамен, то конкретное знамя (Z_2) следует с вероятностью P_i .	
γ – отношение: $Z_1 \xrightarrow{\gamma} Z_2$ $Z_2 \xrightarrow{\gamma} Z_1$	Знамя Z_1 состоит в γ – отношении со знаменем Z_2 , если эти знамена находятся в общем контексте (фразе, предложении, песнопении).	

Совокупность правил, соответствующих каждому из отношений, образует модель данного отношения. Таким образом, синтаксическая модель знаменных песнопений представляется в виде тройки:

$$M = \{M_\alpha, M_\beta, M_\gamma\}, \text{ где}$$

$$M_\alpha = \{(Z_1 \xrightarrow{\alpha} Z_2), p_\alpha\}_{n_\alpha} : \alpha\text{-модель знаменных песнопений,}$$

p_α – вероятность употребления α -правила,

n_α – количество α -правил,

$$M_\beta = \{(Z_1 \xrightarrow{\beta} Z_2), p_\beta\}_{n_\beta} : \beta\text{-модель знаменных песнопений,}$$

p_β – коэффициент непосредственной силы связи между знаменами,

Синтаксический анализ музыкальных текстов

n_β - количество β -правил,

$M_\gamma = \{(Z_1 \xrightarrow{\gamma} Z_2), p_\gamma\}_{n_\gamma}$: γ -модель знаменных песнопений,

p_γ -коэффициент контекстной силы связи между знаменами,

n_γ - количество γ -правил.

Были построены модели для каждого типа отношений по каждой из частей одного из крупнейших источников древнерусских музыкальных текстов - рукописи «Круг знаменных песнопений» под ред. Разумовского.

Фрагмент $M_\beta^{\text{Октоих}}$ (β – модели для части «Октоих») приведен в таблице 3 и на рисунке 4.

Для удобства анализа в строках и столбцах записаны цифровые коды знамен, а в соответствующих ячейках – коэффициенты непосредственной силы связи знамен.

Табл. 3. Фрагмент β – модели для «Октоиха»

	76	77	121	138	89	122
76	66	4	12	0	0	0
77	5	33	0	0	0	1
121	0	1	0	1	4	0
138	1	1	0	0	0	1
89	0	0	0	8	0	0
122	1	0	0	2	0	0

На рисунке 4 таблица 3 представлена в виде гистограммы. На горизонтальной оси расположены знамена, а по вертикальной оси откладываются соответствующие значения коэффициентов силы в процентном соотношении.

Синтаксический анализ музыкальных текстов

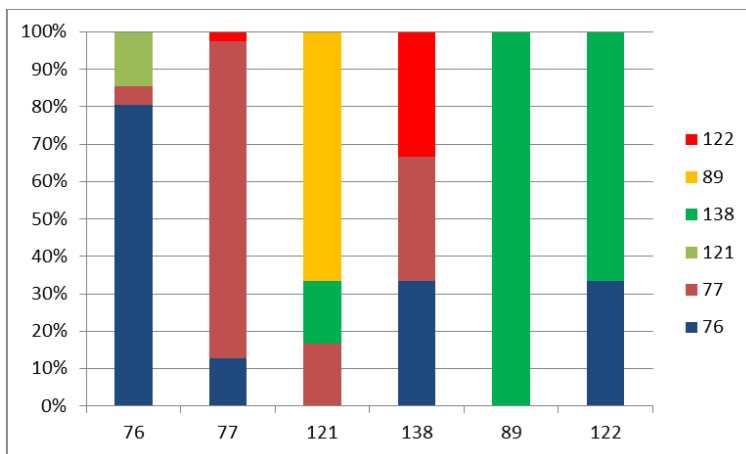


Рис.4. Фрагмент β – модели для «Oktoixh»

Фрагмент $M_{\gamma}^{\text{Oktoixh}}$ (γ – модели для части «Oktoixh») приведен в таблице 4 и на рисунке 5. Значения в них заданы аналогично β -модели.

Табл. 4. Фрагмент γ – модели для «Oktoixh»

	76	77	121	138	89	122
76	0,142	0,024	0,04	0	0	0
77	0,024	0,05	0	0	0	0,002
121	0	0,009	0	0,013	0,011	0
138	0,022	0,007	0	0	0	0,009
89	0	0,004	0	0,018	0	0
122	0,002	0	0	0,009	0	0

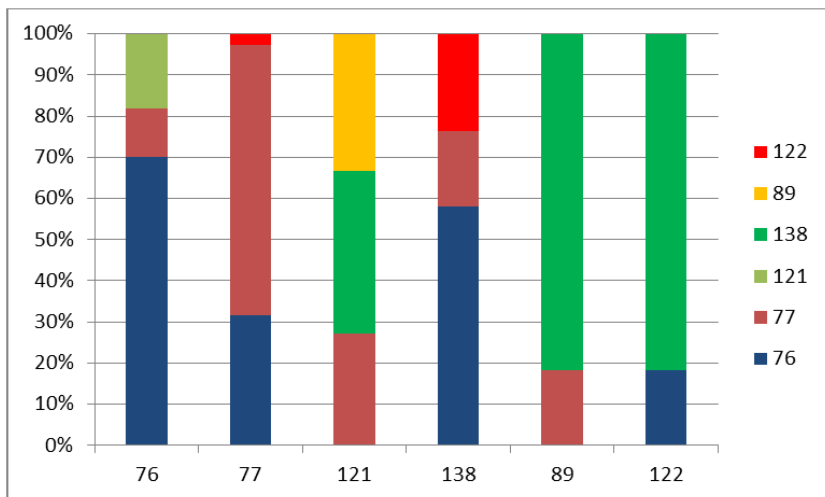


Рис.5. Фрагмент γ – модели для «Октоиха»

Данный этап позволил выявить особый характер связей между знаменами, который не виден на письме. Построение γ – модели позволяет выявить контекстно-связанные знамена, что дополняет общую картину связности знамен. Например, для приведенного отрывка выявлено, что знамя с кодом «89» связано не только со знаменем «138», но и с «77».

Таким образом, знаменная нотация была представлена как знаковая система, что позволило применить к ее изучению лингвистические методы. В дальнейшем предполагается использовать построенные модели для решения задач ИКТ – восприятия и понимания информации человеком.

Выводы

1. Эмпирически было доказано наличие в музыкальных текстах семиотической и синтаксической структур
2. Было предложено три типа отношений между знаками в музыке
3. Была предложена синтаксическая модель знаменных песнопений, которая может быть применена для решения задач восприятия, понимания и обработки музыкальной информации.

Список источников

1. Данышина И.В., Данышина М.В. Статистическое исследование знаменной системы на примере двух гласов Октоиха // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 9. М.: НОК «CLAIM», 2007. Данышина И.В., Данышина М.В. Структура и обработка древнерусских певческих рукописей. // Сборник тезисов докладов «Печатные средства информации в современном обществе (к 80-летию МГУП)». Секция «Электронные средства информации в современном обществе», М. 2010
2. Данышина И.В. Программа для проведения статистических исследований «SemioStatistik» // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 10. М.: НОК «CLAIM», 2009.
3. Данышина И.В. Анализ зависимости слов и знамен с помощью программы Semantic_Statistik. Сборник тезисов докладов общеуниверситетской научно-технической конференции «Студенческая научная весна - 2011», посвященной 50-летию полета Ю.А. Гагарина в космос. Том XI, часть 2. М.: МГТУ им. Н.Э. Баумана, 2011. - 310 с.
4. Голубева И.В. Исследование знаменных песнопений как знаковой системы. Информационные технологии и письменное наследие: материалы IV междунар. науч. конф. (Петрозаводск, 3-8 сентября 2012 г.), 2012
5. Филиппович А.Ю., Голубева И.В. Исследование синтаксиса семиографических песнопений. Проблемы полиграфии и издательского дела, 2012
6. М.Бражников. «Древнерусская теория музыки». - «Музыка», 1972г.

Распознавание и классификация актантов в русском языке

Илья Кузнецов

НИУ ВШЭ, Москва, Россия. iokuznetsov@hse.ru

Аннотация. В статье рассказывается о проекте системы автоматического распознавания и классификации актантов (Semantic Role Labeling) для русского языка. Мы вводим ряд важных концептов, связанных с этой задачей, и на основе этих концептов описываем архитектуру системы, а также возможные методы реализации её компонентов.

Ключевые слова: автоматическая обработка языка; извлечение фактов; data mining; semantic role labeling; семантические роли; semantic web.

Введение

Автоматическое распознавание и классификация актантов (Semantic Role Labeling, SRL) - одна из основных задач автоматической обработки языка на сегодняшний день. Общая формулировка задачи SRL состоит в следующем: для предложения требуется определить участников ситуации, описываемой в этом предложении, и распределить их роли.

В результате создается компактное структурированное представление информации, которого зачастую оказывается достаточно для извлечения фактов из неструктурированных текстов. Также модули SRL используются при формировании Semantic Web, в системах машинного перевода ([1]), при снятии лексической и грамматической неоднозначности ([2]).

Роман Абрамович купил за 112 млн. долларов долю в компании "Труфон"

Предикат: *купить*

Кто: *Роман Абрамович*

Что: *доля в компании "Труфон"*

Цена: *112 млн. долларов*

Результат работы модуля SRL

Основная масса работ по SRL сегодня ведется для английского языка с использованием методов машинного обучения и вспомогательных ресурсов: парсеров, синтаксических корпусов, лексикографических баз, тезаурусов. Создание таких ресурсов требует больших временных затрат и привлечения квалифицированных экспертов. Для русского языка большинство подобных ресурсов находится на стадии разработки¹, в результате чего направление SRL развивается достаточно медленно. Разумным решением представляется замена ресурсов, созданных вручную, на ресурсы, полученные полуавтоматическим способом. Это позволило бы сократить затраты на создание системы SRL для русского языка, хотя и привело бы к определённым потерям в точности.

Лингвистическое описание задачи

Перед тем, как дать формальное определение рассматриваемой задачи, необходимо ввести несколько определений. В первую очередь обратимся к понятию лексемы. **Лексемой** называется единица словарного состава языка. Классический способ описания лексического значения слова – с помощью толкования. В толковании могут содержаться переменные, н., "*подарить кому (X) что (Y)*". Лексемы, в толковании которых содержатся переменные, называются **предикатами**, сами переменные – **валентностями**. К типичным предикатам относятся глаголы (*купить*), номинализации (*покупка*), а также прилагательные. Языковая единица, заполняющая валентность предиката в конкретном тексте, называется его **семантическим актантом**. На уровне синтаксиса семантическим актантам соответствуют актанты **синтаксические**.

Семантические валентности различных предикатов формируют классы – **семантические роли**. Они, в свою очередь, группируются в **инвентари**. Размер и состав инвентаря семантических ролей могут варьироваться в зависимости от теоретических предпочтений исследователя и практических требований. Классический набор из 10 ролей,

¹ Так, FrameNet для английского языка ([3]) содержит порядка 10000 предикатов и 170000 предложений, тогда как размеченная часть русского FrameBank ([4]) – около 500 предикатов и 30000 предложений.

предложенный Ч. Филмором, включает в себя роли *Агенса* (субъект-инициатор), *Пациенса* (объект действия), *Время*, *Место*, а также другие (см. [5]). Другое распространенное решение – группировка предикатов по более общим ситуациям-**фреймам**, при этом внутри каждого фрейма используется единый набор семантических ролей ([3]).

Соответствие между семантическими и синтаксическими актантами, т.е. между ролями и их синтаксическим оформлением, называется **диатезой**. Диатеза может быть описана в словаре (например, с помощью моделей управления ([6])) или задана набором правил.

Прикладной аспект SRL

Рассмотрев теоретическую базу, мы можем сформулировать задачу Semantic Role Labeling более строго. Для произвольного предиката и произвольного предложения, содержащего этот предикат, требуется определить узлы синтаксической структуры, которые являются актантами рассматриваемого предиката, и присвоить каждому из этих актантов семантическую роль. Это определение исключает из задачи смежные области: определение валентностной рамки глагола, разрешение кореференции и др. Тем не менее, без соответствующих модулей создание универсальной системы SRL едва ли является возможным.

В рамках узкой предметной области модуль SRL может быть успешно разработан на основе **лингвистических правил и словарей** ([7]). При попытке применить эту методологию для решения общей задачи возникает ряд проблем: высокая стоимость разработки, высокая сложность результирующей системы, низкая адаптируемость к новым типам текстов. Среди положительных сторон таких методов можно отметить хорошую интерпретируемость и возможность тонкой настройки.

В качестве альтернативы могут быть использованы методы **машинного обучения**. Такой выбор позволяет сократить затраты на разработку и поддержку системы. Система SRL, обученная на одном типе текстов, будет работать менее качественно на текстах другого типа, однако существует возможность адаптировать систему, произведя обучение на текстах нового типа. Современные системы SRL, построенные на машинном обучении с учителем, демонстрируют качество работы порядка $P \approx 0.9$, $R \approx 0.8$, $F_1 \approx 0.85$ для исходного домена ([8]) и порядка $P \approx 0.7$, $R \approx 0.55$, $F_1 \approx 0.6$ для неизвестной предметной области без адаптации ([9]).

Проект системы

Существует несколько вариантов архитектуры SRL-процессора. Наиболее популярна двухступенчатая модель, впервые предложенная в

способы их маркирования могут быть получены из внешнего экспертного ресурса или полуавтоматическим способом. Представляется возможным обучить классификатор на общих семантических ролях, определенных экспертно для небольшого набора глаголов, и распространить результаты обучения на неизвестные предикаты. Мы планируем использовать данные русского FrameBank и расширенную модель, полученную на его основе.

В большинстве известных нам работ по SRL классификация на обоих этапах осуществляется с помощью набора MLE-оценок или метрик близости с последующим комбинированием полученных значений. На начальных этапах разработки мы планируем придерживаться этой методологии, исходя из того, что в случае SRL приоритетное значение имеет выбор признаков для обучения, а не тип классификатора.

При классификации актантов большую роль играет **лексическая информация**. Легко представить пример, в котором актанты могут быть распределены по ролям с использованием исключительно лексической информации: *“купить стол Иван 100 рублей”*. Для того чтобы система стабильно работала с неизвестными словами, необходим обширный внешний источник данных о лексической близости слов. В качестве такого источника может выступать **тезаурус**, составленный экспертно или на основе кластеризации лексики. Мы планируем опробовать различные методы кластеризации (по глагольным связям, по предикатным связям, по контексту) и выбрать решение, которое наилучшим образом моделирует аргументно-предикатную сочетаемость.

В результате работы модуля SRL релевантные узлы синтаксического дерева оказываются связаны с соответствующими предикатами именованными ролевыми отношениями. Эта информация передается на **выход** системы SRL.

Заключение

Задача SRL состоит в отборе и классификации актантов. Однако, как мы можем видеть, даже в минималистичной комплектации система требует использования внешних ресурсов. В рамках проекта по созданию системы SRL для русского языка планируется также совершенствовать ресурсы, предоставляющие вспомогательную информацию.

Для того чтобы сделать проект реализуемым в обозримой перспективе, планируется принять несколько упрощений. На первом этапе область внимания будет ограничена глагольными предикатами, именными актантами и простыми предложениями. После того как будут разработаны методы и инструменты для работы в упрощенных условиях, технология может быть экстраполирована на более сложные случаи.

Список источников

1. Boas, H. Bilingual FrameNet dictionaries for machine translation. // Proc. of LREC 2002. – 2002.
2. Кустова, Г.И., С.Ю. Толдова. НКРЯ: семантические фильтры для разрешения многозначности глаголов // Национальный корпус русского языка: 2006—2008. Новые результаты и перспективы. СПб: Нестор-история. – 2009.
3. Baker, C. F., C. J. Fillmore, J.B. Lowe. The Berkeley FrameNet Project. // Proc. COLING-ACL 98. – 1998.
4. Ляшевская, О.Н., Ю.Л. Кузнецова. Русский фреймнет: к задаче создания корпусного словаря конструкций // Диалог 2009. 8 (15). – 2009.
5. Fillmore, C.J. The Case for Case. // Universals in Linguistic Theory. – 1968.
6. Мельчук, И.А. Опыт теории лингвистических моделей "Смысл-Текст". – 1974.
7. Stallard, David. Talk'n'travel: A conversational system for air travel planning // Proc. ANLP'00. – 2000.
8. Che, W., Liu, T., Li, Y. Improving semantic role labeling with word sense // Proc. HLT'10. – 2010.
9. Croce, D., Giannone, C., Annesi, P., Basili, R. Towards open-domain Semantic Role Labeling // Proc. ACL'10. - 2010.
10. Gildea, D., D. Jurafsky. Automatic Labeling of Semantic Roles // Computational Linguistics, 28(3). – 2002.
11. Merlo P., Musillo G. A., Semantic Parsing for High-Precision Semantic Role Labelling. // Proc. CoNLL-2008. – 2008.
12. Màrquez, L. Comas, P., Giménez, J., Català, N.. Semantic Role Labeling as Sequential Tagging // Proc. CoNLL'05. – 2005.
13. Sharoff, S., Nivre, J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge // Proc. Dialogue 2011 – 2011.
14. Антонова А.А., Мисюрев А.В. “Об использовании синтаксического анализатора Cognitive Dwarf 2.0” // Труды ИСА РАН. Т. 38. – 2008.
15. Abend, O., Rappoport, A. Fully Unsupervised Core-Adjunct Argument Classification // Proc. ACL 2010. – 2010.

Применение автоассоциаторов к распознаванию последовательностей аккордов в цифровых звукозаписях

Н. Ю. Глазырин¹

¹ nglazyrin@gmail.com

УрФУ имени Б. Н. Ельцина, Екатеринбург, Россия

Аннотация. В статье предлагается способ представления звуковых данных в виде вектора признаков, полученного из спектра звука с использованием нейронной сети – автоассоциатора. Данное представление используется для распознавания звучащего аккорда.

Ключевые слова: Музыкальный информационный поиск, распознавание аккордов, автоассоциатор.

1 Постановка задачи

Для распознавания последовательности аккордов в звукозаписи необходимо построить последовательность аккордов с указанием времени начала и конца звучания каждого из них. Аккордом в теории музыки называется одновременное звучание трёх и более звуков разной высоты. Часто множество распознаваемых алгоритмом аккордов ограничивают только мажорными и минорными аккордами, состоящими из трёх одновременно звучащих нот (например, в [3]).

Обычный подход к решению данной задачи состоит в разбиении звукозаписи на короткие фрагменты и преобразовании каждого фрагмента в вектор признаков. Далее эта последовательность

преобразуется в последовательность названий аккордов. При этом, фактически, решается задача классификации: по вектору признаков необходимо определить звучащий на данном фрагменте аккорд либо его отсутствие. В случае только мажорных и минорных аккордов можно выделить 25 классов (24 мажорных и минорных аккорда, 1 класс для отсутствующего аккорда).

Алгоритмы вычисления признаков, используемых для представления звука в задачах музыкального информационного поиска, совершенствуются более 10 лет, охватывая всё большее количество характерных свойств музыкальных звукозаписей. Автоассоциаторы (автоэнкодеры), являющиеся одним из вариантов автоассоциативных нейронных сетей, позволяют автоматически получить признаки путем обучения без учителя на заданном наборе данных. Полученные таким образом признаки показывают хорошую производительность в задачах классификации (см. [5]). Но до сих пор признаки, полученные при помощи автоассоциаторов, не применялись в задаче распознавания аккордов.

2 Признаки, используемые при распознавании аккордов

Для получения вектора признаков на каждом фрагменте сначала вычисляется спектр звука. Вектор значений спектра уже можно рассматривать в качестве вектора признаков, однако многие его значения заметно коррелируют друг с другом. Кроме того, спектр часто бывает зашумлён. Было предложено множество различных преобразований над спектром для преодоления этих недостатков. Их результатом, как правило, является 12-мерный вектор, часто называемый вектором хроматических признаков или хроматическим вектором [4]. Каждая из компонент этого вектора соответствует одному тональному классу. Каждый тональный класс объединяет в себе ноты с одинаковым названием, находящиеся в разных октавах.

Иногда (например, в [3]) также используется 6-мерное пространство признаков, моделирующее взаимоотношения музыкальных звуков в равномерно темперированном строе [2]. Векторы из этого пространства получили название тональных центроид. Любой аккорд можно представить в виде тональной центроиды. При этом расстояния между ними имеют музыкальный смысл: часто используемые вместе аккорды расположены ближе друг к другу.

Для каждого аккорда можно построить эталонный вектор признаков. Например, хроматический вектор, у которого на позициях, соответствующих тональным классам входящих в аккорд нот, нахо-

дятся 1, а на остальных позициях — 0. Тогда для решения задачи классификации достаточно определить расстояния от данного вектора до всех шаблонов и выбрать шаблон с наименьшим расстоянием. Более сложным подходом является моделирование аккорда смесью многомерных нормальных распределений, как, например, в [3].

С появлением в 2005 году первой размеченной коллекции из 180 песен группы The Beatles качество работы алгоритмов распознавания аккордов, проверяется на стандартных коллекциях. При этом как используемые алгоритмы машинного обучения, так и используемые признаки наилучшим образом приспособляются к обработке композиций из этих коллекций. Так, по результатам задачи Audio Chord Estimation в рамках соревнования MIREX 2012 качество работы всех алгоритмов на звукозаписях из впервые использовавшейся для оценки коллекции музыки разных стилей оказалось заметно ниже, чем на давно используемой коллекции из песен групп The Beatles, Queen, Zweieck. Наилучший алгоритм правильно определил звучащий аккорд в среднем для 82,73% от длительности композиции для первой коллекции и 72,49% для второй.

3 Автоассоциаторы

Автоассоциатор (autoencoder) [5] представляет собой нейросеть, состоящую из входного, выходного и одного скрытого слоев. Количество нейронов на входе и на выходе одинаково, а количество нейронов в скрытом слое меньше, чем во входном (существуют варианты с большим количеством нейронов в скрытом слое). Он обучается таким образом, чтобы на выходе с наибольшей точностью восстанавливать входной вектор. При этом нейроны скрытого слоя обучаются эффективно представлять входной вектор в сжатом виде. Существует вариант автоассоциатора, в котором при обучении входные данные намеренно искажаются, а на выходе требуется получить исходный их вид. У него представление данных в скрытом слое оказывается более эффективным. Такие автоассоциаторы называют очищающими (denoising autoencoder). Применительно к распознаванию аккордов, обучающие данные для автоассоциатора могут быть сгенерированы из любых звукозаписей. На вход ему подается вектор значений спектра, а скрытый слой дает их представление в сжатом виде. Также автоассоциаторы можно располагать поверх друг друга, получая ещё более сложные и, вероятно, более устойчивые признаки. Для этого нейроны скрытого слоя одного автоассоциатора соединяются напрямую с нейронами скрытого слоя другого. При этом каждый слой обучается отдельно.

4 Реализация

С точки зрения реализации проще всего добавить поверх нескольких предварительно обученных скрытых слоев автоассоциаторов выходной слой, после чего дообучить полученную нейронную сеть на размеченных данных методом обратного распространения ошибки. Нейронная сеть и алгоритм обучения были реализованы при помощи пакета Theano¹. На вход сети подается предварительно вычисленный спектр фрагмента звукозаписи, состоящий из 60 компонент, соответствующих частотам от 65,4 Гц до 1975,5 Гц. При этом частота каждой компоненты соответствует частоте одной из ступеней равномерно темперированного строя. Нейронная сеть состоит из 2 слоев очищающих автоассоциаторов (54 и 48 нейронов соответственно) и одного выходного слоя из 6 нейронов. В любых двух соседних слоях каждый нейрон одного слоя имеет соединения со всеми нейронами другого слоя. При последовательном обучении двух автоассоциаторов к каждому элементу входного вектора примешивался шум — нормально распределенная случайная величина с параметрами $\mu = 0$, $\sigma = 0.5$ и $\sigma = 0.3$ для 1-го и 2-го слоев соответственно.

Полученные на выходе значения интерпретировались как векторы тональных центроид. В качестве результата выбирался аккорд, расстояние до шаблона которого минимально. Дополнительной обработки последовательности векторов признаков не производилось.

5 Результаты

На наборе из 180 композиций The Beatles и 100 композиций японской популярной музыки RWC Pop Songs [1] удалось добиться правильного распознавания звучащего аккорда в среднем для 65,38% от длительности композиции. Коллекция была разбита на 10 равных частей по 28 композиций в каждой. Было проведено 10 экспериментов, каждый раз 9 частей составляли обучающую выборку, одна часть — тестовую. Указанный результат был получен усреднением по 10 экспериментам. При использовании одного из современных вариантов хроматических признаков (CRP features, см. [4]) автору удалось добиться результата в 69,01%. При использовании таких же хроматических векторов с дополнительной обработкой их последовательности удалось добиться результата 72,87%. Более поздние эксперименты с применением аналогичной дополнительной обработки к последовательности полученных на выходе нейронной сети векторов привели к близким результатам.

¹<http://deeplearning.net/software/theano/>

Таким образом, качество распознавания аккордов несколько ухудшилось при замене современных хроматических признаков на признаки, полученные с использованием автоассоциаторов. Это может быть связано как с неудачным выбором параметров нейронной сети, так и с использованием 6-мерных векторов признаков вместо более часто используемых 12-мерных.

Дополнительная обработка последовательностей векторов признаков позволяет учесть музыкальные закономерности для последовательностей аккордов. Можно высказать предположение о том, что при использовании современных признаков она сильнее влияет на качество распознавания аккордов, чем выбор конкретного алгоритма вычисления признаков.

Достоинством описанных признаков является их устойчивость к шумам. Такие признаки могут лучше работать, например, при анализе записей с микрофона мобильного телефона. Возможность использовать для обучения автоассоциаторов любые звукозаписи может быть полезна для точной настройки полученных признаков на конкретный стиль музыки или набор инструментов. Повышения качества распознавания можно ожидать от лучшего подбора параметров и топологии нейронной сети, от использования других алгоритмов классификации.

Список источников

1. M. Goto, H. Hashiguchi, T. Nishimura, R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. Proceedings of ISMIR 2002.
2. C. Harte, M. Sandler, M. Gasser. Detecting harmonic change in musical audio. Proceedings of the 1st ACM workshop on Audio and music computing multimedia, 2006.
3. E. J. Humphrey, T. Cho, J. P. Bello. Learning a robust Tonnetz-space transform for automatic chord recognition. Proc. of ICASSP 2012.
4. N. Jiang, P. Grosche, V. Konz, M. Müller. Analyzing Chroma Feature Types for Automated Chord Recognition. Proceedings of the 42nd AES Conference on Semantic Audio, 2011.
5. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. J. Mach. Learn. Res., вып. 11, 2010.

Использование связанных пространственных данных в геоинформационных системах

С. Б. Кузьмин¹

¹ to.stepan.kuzmin@gmail.com

УГТУ, Екатеринбург, Россия

Аннотация. В данной работе описывается расширение Quantum GIS для импортирования в проект географической и атрибутивной информации с помощью выполнения SPARQL запросов к хранилищам триплетов RDF.

Ключевые слова: связанные пространственные данные; геоинформационные системы; rdf; sparql; openstreetmap.

1 Введение

Семантическая паутина упрощает задачи интеграции данных, предоставляя инфраструктуру, основанную на RDF и онтологиях. Для того, чтобы использовать паутину как средство интеграции данных, требуются всеобъемлющие массивы данных и словари, позволяющие устранять неоднозначность и упорядочивать остальную информацию. Например, благодаря проекту DBpedia доступен большой массив данных, предоставляющий энциклопедические знания во множестве различных областей.

OpenStreetMap – это некоммерческий проект по созданию подробной, свободной и бесплатной географической карты мира. Этот

проект обладает потенциальной возможностью стать отправной точкой для интеграции пространственной информации в семантическую паутину.

Целью проекта LinkedGeoData[1, 2, 3] является внедрение данных OpenStreetMap в семантическую паутину. Это упростит задачи интеграции и сбора информации из реальной жизни, требующие обширных предварительных познаний о пространственных особенностях объектов реального мира.

Большинство информации получено конвертацией данных из проекта OpenStreetMap в RDF и построения на их основе упрощённой онтологии. Данные LinkedGeoData связаны с массивами данных DBpedia и GeoNames. Разработчики LGD стремятся создать словарь OWL нацеленный на упрощение обмена и повторного использования пространственной информации.

В данной работе описывается способ импортирования географической и атрибутивной информации из хранилища RDF триплетов (например, LinkedGeoData.org) в проекты, создаваемые с помощью ГИС Quantum GIS.

2 Постановка задачи

Для более тесной интеграции геопространственной семантической паутины с существующими геоинформационными системами, необходимо реализовать возможность обмена пространственной и атрибутивной информацией между хранилищами связанных данных и ранее созданными геоинформационными пакетами (ГИП). Первым этапом создания такой возможности является реализация импортирования связанных пространственных данных в геоинформационные системы общего назначения.

Задачей данной работы является создание расширения для геоинформационной системы общего назначения Quantum GIS, которое позволит выполнять SPARQL запросы к хранилищам RDF триплетов и импортировать полученные данные в проект.

3 Описание метода

Суть метода заключается в написании пользователем SPARQL запроса, который будучи выполненным, должен вернуть некий набор географически привязанных данных. Каждой записи из набора соответствует поле с геометрической информацией, которое определяется в процессе синтаксического разбора запроса (используются поля с типом *Geo:Geometry*), и набор полей-атрибутов. Для того, чтобы

определить наличие поля с геометрией в наборе данных, используется регулярное выражение из библиотеки `OpenLayers` для поиска текста в формате `Well-known text (WKT)`. В итоге, из набора формируется векторный слой, геометрия которого определяется полем с географической привязкой, а атрибутивные данные выбираются из остальных полей набора пользователем.

Отличие этого метода от прямого подключения к реляционной базе пространственных данных `OpenStreetMap` в возможности использования всех преимуществ, которые предоставляют связанные данные. Например, агрегация данных из распределённых источников.

4 Обзор расширения

На рис. 1 изображено главное окно расширения. Пользователь указывает имя нового слоя с данными, адрес `SPARQL-endpoint` и сам запрос. На рис. 2 векторный точечный слой, полученный в результате выполнения запроса, и его атрибутивная таблица, состоящая из двух полей.

Для быстрого прототипирования и разработки расширения выбран язык `Python`. Расширение зависит от нескольких библиотек, реализующих необходимую функциональность:

- `RDFExtras` — для синтаксического разбора запроса
- `SPARQL client` — для выполнения запроса

5 Дальнейшая работа

Расширение реализует базовую функциональность импортирования связанных пространственных данных в геоинформационные пакеты. Необходимо повысить стабильность расширения, переписав его на `C++`. Это позволит использовать все функциональные возможности `Quantum GIS`. Необходимо реализовать поддержку геометрий описываемых не только в формате `WKT`, но и в виде набора полей долготы/широты. Реализовать обработку наборов данных с полями со смешанной геометрией.

6 Заключение

В данной работе описан метод использования связанных пространственных данных в геоинформационных системах общего назначения.

Применение предложенного метода позволяет сократить время и упростить разработку геоинформационного пакета, благодаря ис-

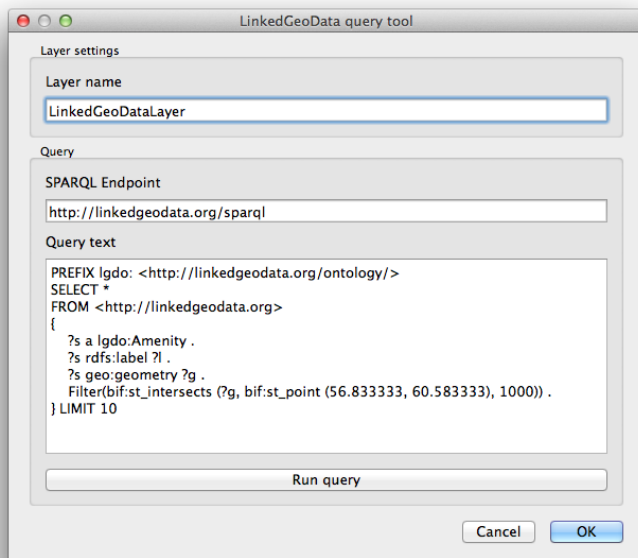


Рис. 1. Окно для выполнения SPARQL запроса

пользованию возможностей семантической паутины и связанных данных.

Расширение распространяется под лицензией GNU GPL v2. Исходный код доступен в репозитории <https://github.com/StepanKuzmin/lgd>.

СПИСОК ИСТОЧНИКОВ

1. LinkedGeoData – Collaboratively Created Geo-Information for the Semantic Web / Soren Auer, Jens Lehmann, Sebastian Hellmann, Universitat Leipzig
2. LinkedGeoData: Adding a Spatial Dimension to the Web of Data / Soren Auer, Jens Lehmann, Sebastian Hellmann, Universitat Leipzig
3. LinkedGeoData: A Core for a Web of Spatial Open Data / Krzysztof Janowicz, University of California, Santa Barbara, USA

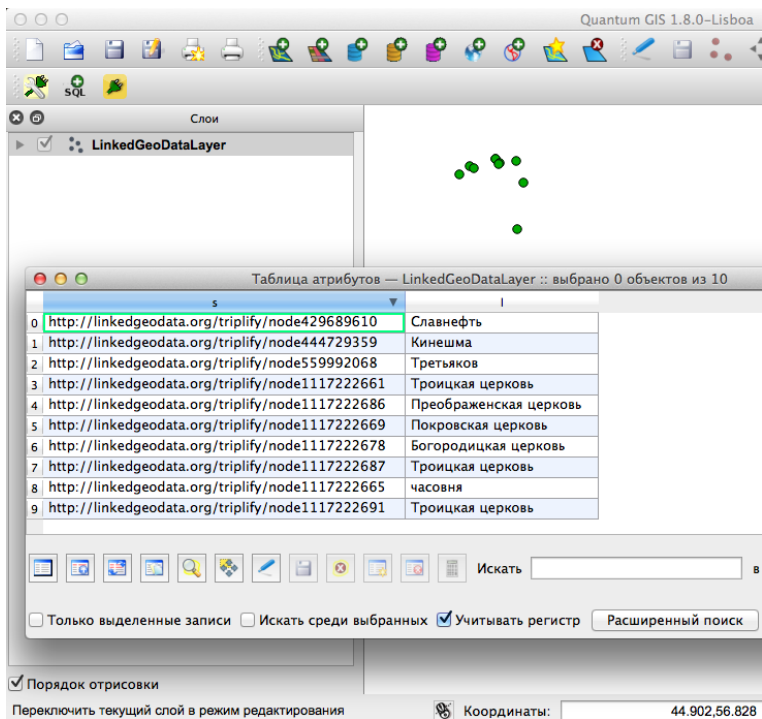


Рис. 2. Векторный слой и таблица атрибутов

Применение модели Бокса-Дженкинса для прогнозирования объемов инвестирования в факторы производства

Екатерина Касаткина¹, Дайана Насридинова²

¹Ижевский государственный технический университет им. М.Т. Калашникова, Ижевск, Россия, e.v.trushkova@gmail.com

²Ижевский государственный технический университет им. М.Т. Калашникова, Ижевск, Россия, daiana1604@yandex.ru

Аннотация. Статья посвящена вопросу применения модели Бокса-Дженкинса для прогнозирования объемов инвестирования в факторы производства, такие как производственный и человеческий капитал региональной экономики. Модель апробирована на статистических данных по Удмуртской Республике. В работе приводятся результаты прогнозирования рассматриваемых показателей на краткосрочную перспективу.

Ключевые слова: эконометрическое моделирование, инвестиции в производственный капитал, инвестиции в человеческий капитал, прогнозирование.

Введение

Ключевым элементом динамичного развития региональной экономики являются инвестиции в различные сферы жизни общества.

Экономисты выделяют инвестиции в производственный и человеческий капитал (факторы производства). Инвестиции в производственный капитал представляют собой совокупность затрат, направленных на

создание и воспроизводство основных фондов. Инвестиции в человеческий капитал включают вложения в его составляющие: капитал образования, капитал здравоохранения, капитал культуры [1].

Капиталовложения в факторы производства необходимо анализировать и прогнозировать, поскольку они являются важной составляющей, при исследовании динамики макроэкономических показателей, характеризующих эффективность функционирование региональной системы.

Инвестиции в производственный и человеческий капитал относятся к экономическим показателям, которые имеют достаточно сложную структуру. Моделирование таких временных рядов путем построения модели тренда, сезонности и циклической составляющей не приводит к удовлетворительным результатам, а ряд остатков часто имеет статистические закономерности. В таком случае используют авторегрессионные модели и модели скользящего среднего [2].

Моделирование динамики инвестиций и их прогноз

Для описания стационарных временных рядов предназначены авторегрессионные модели и модели скользящего среднего. Но, как правило, экономические показатели представляют собой нестационарные, одномерные временные ряды, поэтому более широкое применение получили интегрированные модели авторегрессии – скользящего среднего $ARIMA(p, q, k)$. Такую модель также называют моделью Бокса-Дженкинса [2].

Модель $ARIMA(p, q, k)$ подходит для построения краткосрочных прогнозов. Рассмотрим этапы ее идентификации.

Первоначально подбирается порядок модели k . Дж. Бокс и Г. Дженкинс предлагают взять за критерий стационарности быстрое убывание значений выборочной автокорреляционной функции AC [2].

После нахождения $y_k(t) = \Delta^k y(t)$ идентифицируют параметры $ARMA(p, q)$ модели. Все значения частной автокорреляционной функции PAC для лагов, больших порядка авторегрессии p , статистически незначимы. Для модели скользящего среднего все значения AC для лагов, больших q , близки к нулю [2].

Применим модель Бокса-Дженкинса для моделирования динамики инвестиций в факторы производства Удмуртской Республики. Статистические данные по объемам инвестирования в человеческий капитал (инвестиции в образование (J_1), здравоохранение (J_2) и культуру (J_3)) взяты на официальном сайте Казначейства РФ [3], инвестиций в производственный капитал (I) – на сайте Федеральной службы государственной статистики РФ [4].

Для *ARIMA* модели динамики инвестиций в образование порядок $k=1$ (см. *Рис. 1*). Параметры *ARMA(p,q)* модели определяются по выборочной и частной автокорреляционной функции первых разностей по инвестициям в образование: $p=4, q=4$ (см. *Рис. 2*).

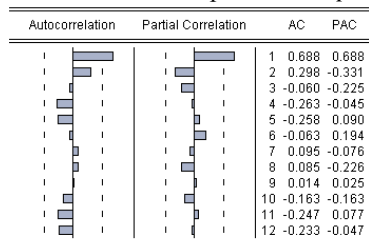


Рис. 1. Выборочная и частная автокорреляционная функция инвестиций в образование

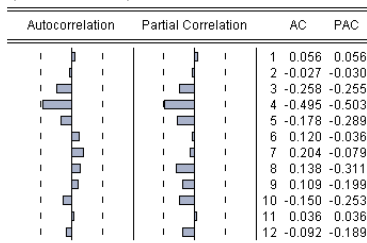


Рис. 2. Выборочная и частная автокорреляционная функция первых разностей инвестиций в образование

Модель Бокса-Дженкинса для моделирования динамики инвестиций в образование Удмуртской Республики имеет вид:

$$J_1(t) = 748,87 + J_1(t - 1) - 0,47J_1(t - 4) + 0,94\varepsilon(t - 4) \quad (1)$$

где: $\varepsilon(t - \tau)$ – ошибка с лагом τ .

Коэффициент детерминации (R^2) уравнения (1) составляет **0,78**, статистика Фишера $F=13,9$, что свидетельствует об адекватности построенной модели.

Прогноз инвестиций в образование на краткосрочную перспективу по модели (1) представлен на *Рис. 3*.

Аналогично осуществляется оценка параметров модели Бокса-Дженкинса для моделирования и прогнозирования динамики инвестиций в здравоохранение (*Рис. 4*), образование (*Рис. 5*) и производственный капитал Удмуртской Республики (*Рис. 6*).

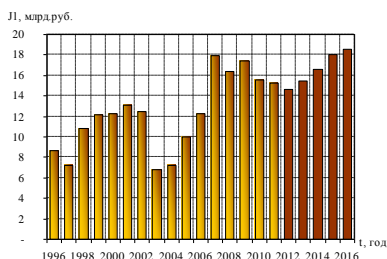


Рис. 3. Динамика инвестиций в образование

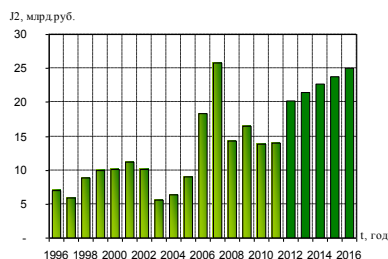


Рис. 4. Динамика инвестиций в здравоохранение

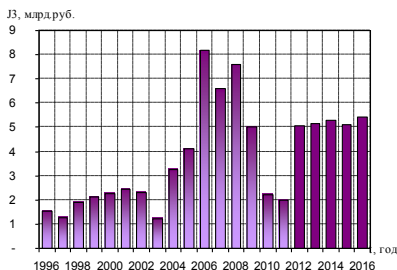


Рис. 5. Динамика инвестиций в культуру

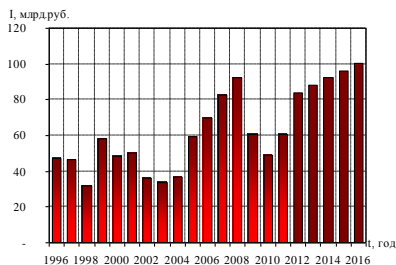


Рис. 6. Динамика инвестиций в производственный капитал

Построенные модели Бокса-Дженкинса для прогнозирования инвестиций в факторы производства обладают высокими коэффициентами детерминации, что говорит об их адекватности.

Выводы

1. Идентифицированы параметры ARIMA модели для динамики объемов инвестирования в факторы производства на основе корреляционно-регрессионного анализа.
2. Построены прогнозы инвестиций в образование, здравоохранение, культуру и производственные фонды с использованием модели Бокса-Дженкинса.
3. Прогнозная динамика инвестиций может быть использована для прогнозирования самих факторов производства с применением экономико-математических моделей.

Список источников

1. Кетова К.В., Русяк И.Г. Экономико-математическая модель анализа и прогноза фактора человеческого капитала // Экономика, статистика, информатика. Вестник УМО. 2007. № 2. С. 56–60.
2. Дуброва Т.А. Статистические методы прогнозирования: учеб. пособие для вузов. М.:ЮНИТИ-ДАНА. 2003. 206 с.
3. Отчетность об исполнении консолидированного бюджета РФ, Министерство Финансов Российской Федерации, Федеральное казначейство (Казначейство России). URL: <http://www.roskazna.ru> (дата обращения 30.05.2012).
4. Федеральная служба государственной статистики РФ. URL: <http://www.gks.ru> (дата обращения 30.05.2012).

Совершенствование одноязычных, двуязычных и мультязычных словарей: автоматизация процесса сбора материала

Мария Кюсева¹, Татьяна Резникова², Дарья Рыжова³

¹mkyuseva@gmail.com, ²tanja.reznikova@gmail.com, ³daria.ryzhova@mail.ru
НИУ ВШЭ, Москва, Россия

Аннотация. Статья посвящена проблемам перевода, а также методам создания и усовершенствования одноязычных, двуязычных и мультязычных словарей. Предлагается алгоритм составления вручную эффективных правил перевода (= словарных статей нового формата) и описывается начальный этап автоматизации работы этого алгоритма.

Ключевые слова: перевод; словарь; база данных; признаковая лексика; лексическая типология.

Введение, постановка задачи

Настоящая статья посвящена проблемам перевода. Общеизвестно, что информации, почерпнутой из словарей, недостаточно для того, чтобы выяснить все условия употребления той или иной леммы. Так, например, в Большом русско-французском словаре АBBYY LINGVO¹ в

¹ Большой русско-французский словарь АBBYY LINGVO представляет собой дополненное и значительно переработанное издание известного русско-французского словаря Л.В.Щербы и М.И.Матусевич (см.[8]) и на сегодняшний день является одним из самых полных русско-французских словарей.

качестве эквивалента русского прилагательного *мягкий* в прямом значении предлагается четыре лексемы: *mou*, *tendre*, *doux* и *moelleux*. Можно было бы предположить, что эти слова взаимозаменяемы и могут употребляться в любом контексте «физической» мягкости, однако первая же попытка реализовать такую замену приводит к неудаче. Так, при переводе сочетания *мягкие руки* на французский язык употребление прилагательного *moelleux* (**les mains moelleuses*) невозможно; слово *mou* (*les mains molles*) повлечет за собой значение «слабые, безвольные руки»; *tendre* (*les mains tendres*) – «ласковые, заботливые руки», и лишь сочетание *les mains douces* окажется адекватным переводом.

Казалось бы, устранить эту проблему можно только путём приведения вместе со словарной дефиницией полных списков коллокаций определяемой леммы. Однако теоретические исследования в области лексики, прежде всего представителей Московской семантической школы (см. [1], [3], [4], [5]), наглядно показывают, что этот неэффективный и трудоемкий способ решения задачи не является единственно возможным. Лексика системна: сочетаемость каждого слова формируется не стихийно, а согласно некоторым вполне определённым правилам. И перечисления этих правил в словарной статье оказывается достаточно для исчерпывающего описания особенностей употребления слова.

Структура правил и процедура их выявления

В центре внимания нашего исследования находятся частотные признаковые слова (прилагательные и наречия). Этот выбор обусловлен, во-первых, сравнительно малым вниманием, уделяемым изучению семантики адъективных слов (исключениями в отечественной лингвистике являются работы [1], [3], [6]), а во-вторых, достаточно простой лексической структурой материала (в отличие, например, от глаголов, прилагательные и наречия представляют собой одновалентные конструкции, а значит, для описания особенностей их употребления оказывается достаточно сформулировать правила заполнения только одного слота).

База данных по многозначным качественным прилагательным и наречиям русского языка. Первым этапом в нашей работе стало детальное исследование сочетаемости русской признаковой лексики, продуктом которого является База данных по многозначным качественным прилагательным и наречиям русского языка (см. [2]). Этот инструмент можно назвать толковым словарем русского языка нового типа. Одной записи в Базе данных соответствует одно значение многозначного слова. Каждое значение характеризуется по ряду параметров: ему приписывается словарное определение, даются примеры употребления, ука-

зывается значение, от которого образовано данное, и тип семантического перехода, посредством которого оно возникает. Кроме того, для каждого значения определяются особенности семантического контекста (= таксономический класс существительных, в сочетании с которыми слово приобретает это значение). Так, например, для прилагательного *дорогой* в Базе хранится две записи: первая отражает значение «стоящий больших денег», которое лексема приобретает в сочетании с существительными, обозначающими артефакты, вторая – «такой, к которому испытывают привязанность, любовь». Это значение, образованное от первого метафорическим переносом, прилагательное принимает, в основном в сочетании с существительными, обозначающими людей (*дорогой друг*), речь (*дорогие слова*), объекты из ментальной сферы (*дорогие сердцу воспоминания*).

Оказалось, что для каждого значения заносимых в Базу слов можно указать таксономические классы существительных, в сочетании с которыми это значение реализуется. Более того, для разных значений одной лексемы эти классы не пересекаются (за редкими исключениями, которые эксплицитно отображаются в Базе). А значит, эти данные можно использовать для автоматической обработки языка. Так, на их основе создаются «семантические фильтры», которые позволят частично снять омонимию в Национальном корпусе русского языка. Благодаря таким фильтрам, например, на запрос «прилагательное цвета в сочетании с существительным, обозначающим эмоцию (*черная грусть, белая зависть, зеленая тоска*)» не будет выдаваться сочетание *пшеничная мука* (второе значение прилагательного *пшеничный* – «очень светлый, с золотистым оттенком»; существительное *мукá* в корпусе с непроставленной акцентуацией смешивается с лексемой *мука*), так как в графе «семантический контекст» для значения ‘золотистый’ прилагательного *пшеничный* в Базе данных нет таксономического класса «эмоции» (см. [7]).

Типологически ориентированная база данных адъективной лексики. Следующий шаг нашего исследования – создание мультязычного словаря, которым станет разработанная нами Типологически ориентированная база данных адъективной лексики. В эту Базу заносится семантическая информация не только о русских признаковых словах, но и об их переводных эквивалентах (для каждого признака на данном этапе работы приводятся материалы 5 – 15 языков, всего в Базу включено 50 качественных признаков).

Для того чтобы составить исчерпывающее описание адъективных слов в разных языках, мы также собираем их сочетаемостные характеристики, и в данном случае основой для сбора материала служат типологические анкеты. Эти анкеты представляют собой списки существи-

тельных, разные для разных полей. Для каждого языка при каждом слове поля указывается возможность/невозможность его употребления с данным существительным².

Первоначально анкеты составляются на основе анализа русского материала, который проводится с помощью толковых словарей, НКРЯ [10] и языковой интуиции исследователей, для которых русский язык является родным, потом они дополняются материалами корпусов, словарей и опросов носителей других языков.

Такое детальное типологическое исследование признаков слов подтвердило гипотезу о системном устройстве лексики. Одному прилагательному в русском языке часто соответствует несколько слов в некотором другом языке (и наоборот). Однако эти слова никогда не являются полными синонимами, взаимозаменяемыми в любом контексте: они всегда распределяются либо по диалектам, либо по регистрам, либо (чаще всего) по значению. Разница в значениях таких прилагательных предопределяет и разницу в их сочетаемости: они покрывают различные зоны семантики русского слова (в обратном же случае, русские квази-синонимы покрывают разные области значения более широкого прилагательного другого языка). В том же случае, если переводной эквивалент у русского слова один, его сочетаемость практически никогда полностью не совпадает с сочетаемостью русского прилагательного, и различия описываются простыми правилами.

Примером ситуации первого типа может послужить признаковое поле 'острый'. Одному прилагательному в русском языке соответствуют две лексемы в коми-зырянском – *лэчыд* и *ёсь*. При этом *лэчыд* описывает только инструменты с режущим краем (нож, коса, пила), а *ёсь* – с колющим концом (игла, стрела, копьё). Во французском языке переводом слова *острый* в физическом значении могут служить прилагательные *tranchant*, *aigu* и *pointu*. И если *tranchant* характеризует объекты с острым краем, а *aigu* – с острым концом, то *pointu* описывает предметы особой формы: сужающиеся к концу (ср. 'острый нос/колпак/подбородок').

Ситуацию, когда одному прилагательному некоторого языка соответствует несколько русских признаков лексем, можно проиллюстрировать примером лексикализации поля 'тонкий'. Так, например, в тегинском говоре хантыйского языка прилагательное *vas* описывает как длинные вытянутые предметы цилиндрической формы (нити, верёвки, стволы деревьев), так и длинные вытянутые плоские предметы (ленты, полосы). В первом случае его переводным эквивалентом в русском языке является прилагательное *тонкий*, а во втором – *узкий*.

² Ср. фрагмент типологической анкеты для признака 'тяжелый' в приложении.

Случай неполного совпадения сочетаемости на первый взгляд эквивалентных прилагательных в двух разных языках можно проиллюстрировать на примере прилагательных *острый* (русск.) и *оштар* (срб.). Эти слова восходят к одному праславянскому корню, и условия их употребления в русском и сербском языках во многом совпадают. Однако при этом только *оштар* сочетается с существительными, обозначающими ткань или одежду (одеяло, свитер), и выражает при этом значение, близкое к тому, что в русском языке передаётся прилагательным *колючий*³.

Таким образом, на основе материала нескольких языков можно выделить *фреймы* – минимальные ситуации, на которые «реагирует» язык (о термине «фрейм» см., например, [9]). Другими словами, это минимальное количество позиций нашей анкеты, для описания которых в каком-нибудь языке может существовать специальная, отдельная лексема. Так, например, в случае с полем ‘острый’ такими фреймами будут ‘инструмент с заточенным краем’ – объекты только такого типа описывают в прямом значении коми-зырянское прилагательное *лэчыд* и французское *tranchant*; ‘инструмент с колющим концом’, который является единственным в физической зоне для французской лексемы *aigu*, ‘объект острой формы’, покрываемый французским прилагательным *pointu*, и ‘предмет с колющейся поверхностью’, описываемый русской лексемой *колючий*.

Для поля ‘тонкий’ выделяются, среди прочих, фреймы ‘небольшие в диаметре вытянутые предметы цилиндрической формы (тонкий стержень, веревка)’, ‘плоские вытянутые предметы небольшой ширины’ (узкая лента, полоса).

На основе таких фреймов и строятся сочетаемостные правила для каждой лексемы. Например, правила перевода хантыйского прилагательного *vas*’ на русский язык будут выглядеть так:

- *vas*’ => *тонкий* (в сочетании с существительными, обозначающими длинные вытянутые предметы цилиндрической формы);
- *vas*’ => *узкий* (в сочетании с существительными, обозначающими вытянутые плоские предметы).

Автоматизация процесса сбора правил: этап подготовки

Выявление правил такого рода позволяет создать очень качественный мультиязычный словарь, пригодный как для ручного, так и для ма-

³ Заметим при этом, что в сербском языке есть и аналог русского признакового слова *колючий* – *бодљикав*, однако распределение значений в паре *бодљикав* – *оштар* отличается от распределения значений в паре *колючий* – *острый*.

шинного перевода. Однако у этого метода есть один существенный недостаток: он крайне трудоёмкий и требует очень большого количества человекочасов для своей более или менее полной реализации.

Но теперь, когда тактика работы по выявлению правил сочетаемости уже выработана, представляется возможной хотя бы частичная автоматизация процесса сбора материала. Мы выполнили начальный этап этой работы: перевели на машинную основу процесс составления типологических анкет.

Для этого мы, используя биграммы google [11], автоматически собрали списки коллокаций исследуемых нами русских прилагательных. Прежде всего, мы выделили биграммы, в которых первым элементом является интересующее нас русское прилагательное. Далее с помощью морфологического парсера Mystem [12] мы отбросили сочетания прилагательного с не-существительным (ср. *острый и, остро в*) и оставшиеся сочетания отсортировали по частотности, отрезав нижнюю часть списка (биграммы, частотность которых меньше 100). В результате в нашем распоряжении оказался список существительных, с которыми сочетается интересующее нас русское аъективное слово.

Затем мы автоматически, с помощью машиночитаемых русско-английских словарей, определили список английских прилагательных, претендующих на роль переводных эквивалентов русских слов. И затем с использованием английских биграмм составили список коллокаций найденных английских прилагательных. Полученные списки английских существительных, с которыми сочетаются интересующие нас прилагательные, мы автоматически перевели на русский язык и объединили с уже имевшимся списком существительных, полученным на основе русских биграмм.

Таким образом, получилась достаточно подробная анкета⁴, уже заполненная материалами русского и английского языков. Эта же процедура будет проделана и с прилагательными других языков, для которых доступны списки биграмм или достаточно большие электронные корпуса. Составленные и заполненные таким образом анкеты заметно облегчат и ускорят процесс выделения фреймов, а значит, и процесс составления правил перевода, который в дальнейшем, вероятно, можно будет полностью перевести на автоматическую основу.

⁴ Отличие анкет, созданных по такому алгоритму, от анкет, составляемых вручную, заключается в том, что существительные в них не сгруппированы по таксономическому классу. Эту проблему мы надеемся решить с помощью привлечения материалов тезаурусов.

Список источников

- 1 Апресян Ю.Д. 1995. Избранные труды. Том 1. Лексическая семантика. Синонимические средства языка. М.: Наука.
 - 2 Карпова О.С., Резникова Т.И., Архангельский Т.А., Кюсева М.В., Рахилина Е.В., Тагабилева М.Г., Рыжова Д.А. 2010. База данных по многозначным качественным прилагательным и наречиям русского языка // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог 2010» (Бекасово, 26-30 мая 2010 г.). Вып. 9 (16).– М.: РГГУ.
 - 3 Кустова Г. И. 2004. Типы производных значений и механизмы языкового расширения. М.: Языки славянской культуры.
 - 4 Рахилина Е.В. 2000. Когнитивный анализ предметных имен: семантика и сочетаемостью М: Русские Словари.
 - 5 Рахилина Е.В. 2010. Лингвистика конструкций. М: Азбуковник.
 - 6 Толстая С.М. 2008. Пространство слова. Лексическая семантика в общеславянской перспективе. М.: Индрик.
 - 7 Шеманаева О.Ю., Кустова Г.И., Ляшевская О.Н., Рахилина Е.В. 2007. Семантические фильтры для разрешения многозначности в национальном корпусе // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог–2007». М. С. 582–587.
 - 8 Щерба Л. В., Матусевич М. И. 1969. Русско-французский словарь. — М.: Советская энциклопедия.
 - 9 Fillmore, Charles J. 1985. Frames and the semantics of understanding // *Quaderni de Semantica*, 6.2, 222-254. [в рус. пер.: Фреймы и семантика понимания // Новое в зарубежной лингвистике вып. XXIII. М.: Прогресс. 1988]
- Электронные ресурсы:**
- 10 Национальный корпус русского языка (НКРЯ):
www.ruscorpora.ru
 - 11 Google Books Ngram Viewer
<http://storage.googleapis.com/books/ngrams/books/datasetsv2.html>
 - 12 Морфологический анализатор Mystem
<http://company.yandex.ru/technologies/mystem/>

Приложение

Фрагмент анкеты для русского прилагательного *тяжёлый*

физические значения		<i>тяжёлый</i>
то, что тяжело нести	сумка	+
	чемодан	+
	рюкзак	+
	ведро с водой	+
	волосы	+
то, что трудно сдвинуть/вытащить	ящик	
	дверь	
	педаль	
тяжелый на вид	люстра	+
	грузовик	+
	серьги	+
	арка, портьеры	+
	свод	+
	тучи, облака	+
	подбородок, черты лица	+
	конверт, бумажник	+
вес больше нормы	дубина	+
	кулак	+
	подзатыльник	+
	камень, булыжник	+
одежда: теплый	шуба	
	шапка	
	шаль/шарф	
	носки	
высокая степень: неподъемный	шкаф	+
	пианино	+

Дедупликация почтовых адресов с помощью методов обработки естественного языка и машинного обучения.

Артем Филиппов, Александр Семёнов

afilippov@kpmg.ru, alexandrsemenov@kpmg.ru

КPMG Москва

Аннотация. В докладе представлена разработка решения для дедупликации текстовых данных на примере почтовых адресов с использованием методов обработки естественного языка и алгоритмов машинного обучения. Описываются основные этапы работы, оценка качества результатов и дальнейшие шаги по усовершенствованию системы.

Ключевые слова: дедупликация текстовых данных, машинное обучение, обработка естественного языка, информационный поиск, деревья решений, алгоритм C4.5

Введение

Автоматическое определение, разбор и дедупликация адресов в больших массивах слабоструктурированных данных является одной из наиболее востребованных задач в области "информационного поиска" ("Information Retrieval") и обработки естественного языка ("Natural Language Processing"). В рамках оптимизации сервиса, предназначенного для автоматического анализа конфликтов интересов, аффилированности контрагентов и выявления прочих рисков, мы столкнулись с проблемой дедупликации больших объёмов текстовых данных (имена людей, адреса, номера телефонов, названия

организаций), полученных из различных внешних источников, характеризующихся разными форматами данных.

Зачастую, при выгрузке данных даже из одного источника, один и тот же адрес может быть написан тремя различными способами. С учётом того, что количество таких строк исчисляется тысячами, поиск взаимно однозначных соответствий между несколькими похожими адресами, полученными из различных внешних источников, может быть выполнен с оптимальными затратами лишь при помощи алгоритмов обработки естественного языка и машинного обучения.

Таким образом, нашей задачей было нахождение оптимального с точки зрения точности и простоты внедрения метода автоматической дедупликации текстовых данных подобного рода, и в первую очередь – почтовых адресов. В данной работе мы опишем наш подход к решению этой проблемы, предварительные результаты и предполагаемые способы их улучшения.

Существующие решения

Одним из решений для сравнения двух адресов, представленных в текстовом виде, на предмет их схожести, может являться выделение частей для дальнейшего их сравнения.

В результате анализа способов выделения частей в адресной строке [1] было принято решение, что разбиение почтового адреса на основе алгоритма нейронных сетей, описанного в [1], может оказаться не точным.

Например в строке «423806 г Набережные Челны Набережные Челны г, Железнодорожников ул, 20, 3» тип населенного пункта (в данном случае «город») встречается дважды, название населенных пунктов находятся между этими типами, что не позволит методу на основе нейронных сетей правильно выделить части.

Поэтому более подходящим решением в данной ситуации, на наш взгляд, является использование правил преобразования, а также словаря для определения названий, не принадлежащих ни одному типу.

Для дальнейшего сравнения частей следует использовать алгоритмы машинного обучения в силу большого объема имеющихся у нас данных, которые были размечены вручную. В качестве основного метода нами были выбраны алгоритмы построения обучаемых деревьев решений, т.к. они являются более

прозрачными с точки зрения модели в сравнении с остальными методами. Для построения деревьев решений использовался алгоритм C4.5 [3].

Формат и объем данных

Для апробации алгоритма было собрано 32500 пар адресов, которые были вручную размечены как совпадающие или несовпадающие (0/1). Каждый адрес представлен в виде текстовой строки. Обучение производилось на 22500 пар адресов, соответственно проверка результатов и оценка точности производилась на 10000 пар.

Далее мы опишем этапы работы нашего алгоритма.

Разбиение адреса на части.

Данный этап предполагает разделение исходного адреса в текстовом представлении на части (индекс, населенный пункт, улица, дом, корпус/строение, квартира) текстового формата.

На первом этапе осуществляется разбиение строки на лексемы и дальнейшее выделение известных частей (типов, например ул., дом, г, кв), а также разделителей (например, запятых). Далее лексемы группируются в блоки, а границей блоков служат типы либо разделители. Неизвестная часть в блоке считается названием типа. В случае нахождения блоков без типа используется словарь. В некоторых противоречивых ситуациях используется словарь либо набор эвристических правил.

Например: 423806 г Набережные Челны Набережные Челны г, Железнодорожников ул, 20, 3

Выделенные типы: 423806 – индекс, г – населенный пункт, ул –улица.

К первому типу «город» в блок попадут неизвестные части – «Набережные Челны Набережные Челны», но у второго типа «город» нет неизвестной части. Данная ситуация будет проверена при помощи словаря и выделены части: «г Набережные Челны» и «Набережные Челны г».

Ниже перечислены примеры эвристических правил:

- *Если указываем на блок без типа и он не числовой, и встречался город - то это улица;*

- Если указываем на блок без типа и он числовой, и встречалась улица, и не встречался дом - то это дом;
- Если указываем на блок без типа и он числовой, и встречалась дом, то это квартира;
- и т.д.

Сравнение одноименных частей адресов

Сравнение разделенных частей осуществляется на основе алгоритмов нечеткого сравнения строк: Дамерау-Левенштейна [4] и Fuzzy String Matching [5]. В результате сравнения частей на выходе данного этапа получается числовое значение отдельно для каждой части (количественная мера).

Построение дерева решений.

В результате получения количественных мер сравнения для частей адресов алгоритмом С 4.5 строится обучаемое дерево принятия решений. Фрагмент дерева изображен на рисунке 1.

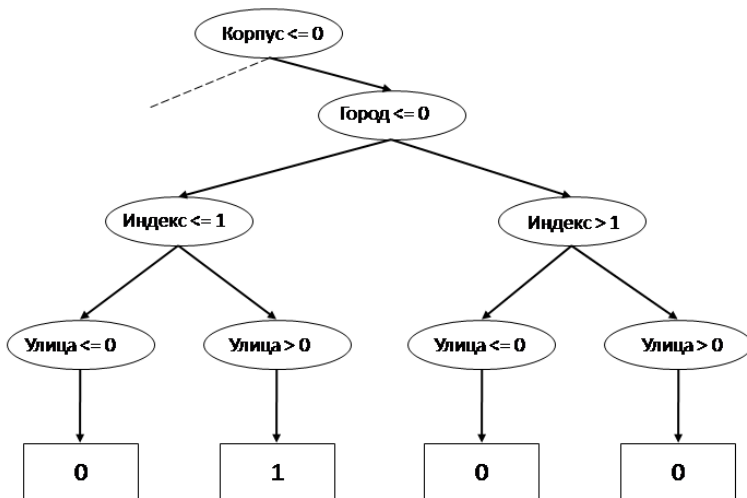


Рисунок 1. Фрагмент дерева решений.

В «узлах» дерева находятся условия проверки критериев сравнения по частям, на которые были разделены адреса (индекс, населенный пункт, улица, дом, и т.д.). Запись вида «Индекс ≤ 1 » означает, что в результате сравнения двух одноименных частей (индексов) критерий сравнения наберет значение не более 1 очка (в данном случае по критерию Левенштейна) - необходимо перемещаться в левое дерево.

В «листьях» дерева хранятся результаты сравнения двух адресов: 0 – адреса не совпадают, 1 – адреса совпадают.

Полученные результаты

На данный момент достигнуто 98,58% точности (отношение числа правильно классифицированных данных к общему числу данных. 9858 правильных классификаций). Среди неправильных классификаций 66 неодинаковых адресов определены как одинаковые (46% от общего количества false positive) и 76 одинаковых – как неодинаковые.

Дальнейшие шаги по улучшению работы алгоритма

В качестве возможных способов улучшения работы алгоритма выделены следующие способы:

1. Произвести тестирование на том же объеме данных с разбиением исходного массива в отношении 80/20 и выборкой данных из других мест. Данный способ позволит на размеченных на текущий момент данных увеличить точность работы алгоритма, а также произвести его тестирование.
2. Тестирование данных на объеме 500 000 адресов. Выявление ошибок, улучшение точности.
3. Использование алгоритма бустинга [6]. Усиление модели принятия решений.
4. Сравнение реализованного подхода с другими алгоритмами классификации для достижения большей точности и полноты.

Выводы

- 1. Разработанная система, основанная на использовании методов обработки естественного языка и алгоритмов машинного обучения продемонстрировала достаточно высокие показатели точности результатов;*
- 2. Мы планируем увеличить точность за счёт увеличения соотношения и объёмов обучающей/проверяющей выборок, сравнить результаты нашего подхода с другими алгоритмами классификации, а также применить бустинг для усиления модели.*

Список источников

1. Алексей Арустамов "Разбор адреса на составляющие", Base-Group Labs <http://www.basegroup.ru/library/cleaning/addresses/>
2. "Точка, точка, запятая: машинное обучение" Блог компании Mail.ru <http://habrahabr.ru/company/mailru/blog/112142/>
3. J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.
4. В.И. Левенштейн (1965). "Двоичные коды с исправлением выпадений, вставок и замещений символов". Доклады Академий Наук СССР 163 (4): 845–8
5. Boytsov, Leonid (2011). "Indexing methods for approximate dictionary searching: Comparative analysis". Jea Acm 16 (1): 1–91.
6. Рассел, Норвиг. Искусственный интеллект, Издательство "Вильямс", 2007, с. 884.

Построение системы распознавания и определения типа галактик

А. А. Михайлов¹, В. М. Волкова

¹ alexander.a.mihailov@gmail.com

Кафедра программных систем и баз данных
НГТУ, Новосибирск, Россия

Аннотация. Статья посвящена решению задачи распознавания пространственных структур, описанных большим набором данных, на примере создания системы распознавания галактик. Работа выполняется в рамках общего проекта по моделированию космологических процессов.

Ключевые слова: анализ изображений; анализ данных; распознавание галактик; космология; классификация; нейронные сети; big data; data mining; particle mesh.

1 Предпосылки к работе

Совершенно ясно, что имитационное моделирование играет очень важную роль в развитии науки. Также сложно поспорить и с значимостью наличия хорошей визуализации результатов моделирования. В настоящее время в научном сообществе есть большой интерес к задачам космологии. В этой области в последнее время именно в ходе визуального анализа [6] были открыты, например, явления:

- рост сгущений вокруг квазаров (Sijacki, 2007; Di Matteo, 2005);
- потоки холодного газа, формирующего галактики (Dekel, 2009; Keres, 2005);

— развитие ионизационных фронтов в ходе эпохи реионизации (Shin, 2008; Zahn, 2007).

В качестве примера системы космологического моделирования, можно привести проект GADGET [5], в рамках которого публично были решены две задачи моделирования: MassiveBlack ($2 \cdot 3200^3$ частиц, один снимок занимает 3 терабайта, данные в сыром виде — 120 терабайт) и E5 (анимация, $2 \cdot 336^3$ частиц, 1367 снимков).

Объём данных, которым оперируют исследователи, делает непосредственный визуальный анализ малоэффективным. Так, возникает задача автоматизированного анализа результатов моделирования.

В настоящей работе рассматривается подход к созданию системы, позволяющей выделять и классифицировать в результатах космологического моделирования области высокой плотности (галактики).

2 Выделение галактик

Для определения областей, в которых могут находиться звёздные скопления, предлагается группировать исходные данные (множество точек) в некоторой однородной сетке, вычисляя в ячейках сетки значение плотности точек в соответствующих областях пространства. После создания сетки, к ней можно применить ряд методов, которые позволят выделить скопления.

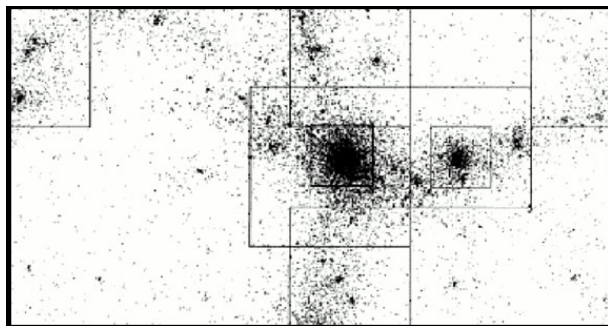


Рис. 1. Определение областей высокой плотности

Похожий принцип применяется во многих методах моделирования (в классе методов Particle Mesh, [3]), что позволяет, в случае высокой степени интеграции модуля распознавания с физическим модулем, не выполнять группирование, а использовать имеющиеся результаты.

На рисунке проиллюстрированы результаты выделения областей высокой плотности в двумерной области с помощью метода типа Particle Mesh, однако метод может быть обобщён и на трёхмерное пространство (РЗМ).

В большинстве типов галактик основное количество звёзд расположено вдоль некоторой плоскости (дисковые галактики, спирали с баром и без). Для других (эллиптические) имеет место осевая симметрия. Поэтому можно утверждать, что для каждой из таких галактик существует плоскость, при проецировании координат звёзд на которую почти не происходит потерь информации, полезной для процедуры классификации. Таким образом, можно перейти от трёхмерной задачи к двумерной.

3 Классификация

Существует общепринятая классификация галактик [7], согласно которой исследователь может отнести каждое из таких звёздных скоплений к одному из классов, либо идентифицировать как неправильный объект.

Первый вариант классификации предложен в 1926 году Э. Хабблом [4] и определял три больших группы: *эллиптические*, *спиральные галактики* и *спиральные галактики с баром*. Позже, данная классификация была уточнена де-Вокулёром в 1962 году [2].

Основные проблемы, которые возникают при классификации галактик, обнаруженных в результатах моделирования:

- большой объём данных (реальные галактики состоят из миллиардов звёзд);
- многомерный пространственный анализ (метод должен быть устойчив к повороту и масштабированию, либо требуется процедура нормализации объектов);
- отсутствие аналитической формализации галактик разных типов и малая мощность обучающей базы;
- необходимость выбора между близкими гипотезами.

В [1] предложены группы методов, которые могут быть применены к задаче распознавания типа галактик: классические стохастические и детерминированные методы классификации; спектральный анализ Фурье; экспертные системы; системы с адаптивным обучением.

4 Основные этапы решения

Подытоживая, приведём схему и конкретный набор методов для решения задачи поиска и классификации галактик.

- 1) Поиск областей высокой плотности (облаков) с помощью Particle Mesh. Результатом работы метода является набор областей-параллелепипедов, в которых точки расположены наиболее контрастно. Одна такая область может занимать несколько ячеек исходной сетки, а галактика лежать на нескольких областях.
- 2) Выделение признаков:
 - a) Соотношение осей эллипса-ядра галактики;
 - b) Плотность распределения звёзд на границе ядра;
 - c) Контрастность бара (соотношение плотностей внутри и снаружи ядра-бара);
 - d) Угол наклона основания рукавов галактики (если присутствуют) к бару;
 - e) Контрастность рукавов;
 - f) Ширина рукавов;
 - g) Длина рукавов.
- 3) Классификация обнаруженных облаков в пространстве признаков, описанных в предыдущем пункте, с помощью нейронных сетей (классификация по образцу). Типы выделяются согласно классификации по де-Вакулёру.

5 Параллелизм

Так как задача может оказаться слишком велика для решения на одном вычислительном узле, необходимо предусмотреть возможность параллельной реализации вычислений. Для этого предполагается произвести декомпозицию расчётной области на равномерной (либо, если есть возможность накопления информации о динамике развития системы, равномерной) сетке, после чего выполнить поиск и классификацию параллельно внутри каждого из элементов разбиения.

Для организации такого уровня параллелизма предполагается использование технологии MPI. Система будет запускаться на кластере Сибирского Суперкомпьютерного центра (<http://www2.sccc.ru>).

Кроме того, для эффективного выполнения подзадач, которые выполняются в рамках одного вычислительного узла (многоядерного процессора), параллелизм организуется как для системы с общей памятью посредством технологии OpenMP.

6 Заключение

В работе рассмотрена задача обработки результатов астрофизических данных, полученных в результате моделирования. Результаты представлялись в виде набора точек, описанных координатами трёхмерного пространства. Авторами предложен подход к систематизации таких данных путём выделения областей, в которых число точек контрастно выделяется в остальном пространстве, и дальнейшим анализом этих областей (галактик). Выделения областей пространства, соответствующих галактикам, упростит исследователю наблюдение и анализ процессов развития таких объектов и взаимодействия между ними.

В работе предложены признаки, над которыми работает классификатор. Предполагается, что зависимость решения от описанных признаков существенно нелинейна, поэтому для процесса классификации были выбраны нейронные сети.

Следующим этапом развития проекта является написание программного обеспечения, реализующего описанную математическую модель.

Список источников

1. Automated galaxy recognition / Barry Rappaport, Kurt Anderson // IN: Astronomy from large databases: Scientific objectives and methodological approaches; Proceedings of the Conference, Garching, Federal Republic of Germany. — 1988. — 10. — Pp. 233–238.
2. Classification of Galaxies by Form, Luminosity and Color / G. de Vaucouleurs. — 1962
3. Computational Astrophysics 1. Particle Mesh methods / Romain Teyssier, Oscar Agertz. — 2009.
4. Extragalactic nebulae / E.P. Hubble // Astrophysical Journal. — 1926. — 12. — Pp. 321–369.
5. Gadget. A cosmological Tree-PM-SPH code / Klaus Dolag. — 2010.
6. Terapixel imaging of cosmological simulations / Yu Feng, Rupert A.C. Croft, Tiziana Di Matteo et al. — 2011.
7. Wikipedia.org. Морфологическая классификация галактик. — web-page. — 2012.

Идентификация подписи с помощью радиальных функций

Эллина Анисимова

К(П)ФУ, Казань, Россия. ellin_a@mail.ru

Аннотация. Статья посвящена исследованию и идентификации on-line подписи с помощью радиальных функций и вейвлетов. Предлагается инвариантное относительно положения представление подписи в виде некоторой функции. Коэффициенты разложения этой функции по радиальному базису применяются для идентификации подписи. Даны оценки достоверности предложенной процедуры.

Ключевые слова: on-line подпись, идентификация, радиальная функция, вейвлет-преобразование.

Введение

Несмотря на наличие большого числа методов для биометрической идентификации личности, использование для этой цели подписи находит самое широкое применение. В этой связи понятен интерес исследователей к задаче автоматической идентификации личности по подписи. Первоначально подпись рассматривалась как графический объект, но с появлением новых устройств ввода возникла задача описания on-line подписи, то есть подписи вместе с динамикой ее создания [2], [3], [4].

Традиционный способ идентификации объекта сводится к вычислению некоторых векторов-признаков и дальнейшему сравнению полученных векторов с помощью какой-либо метрики. В данной работе

предложен новый метод получения таких векторов и приведены результаты экспериментов.

Новизна предлагаемого метода распознавания подписи

Основой любого алгоритма распознавания является выбор признаков, представленных в виде векторов. Для дальнейшего анализа полученных векторов применяют один из возможных способов классификации. Наибольшее распространение получили методы на основе нейронных сетей или SVM [1]. Задача распознавания никогда не имеет единственного решения. Предложенный способ получения векторов-признаков не отрицает имеющиеся методы распознавания, он является дополнительной альтернативой при определении результатов и заключении выводов.

Новизна предлагаемого метода заключается в том, что:

1. предлагается заменить подпись некоторой функцией от одного аргумента, причем вид этой функции не зависит от положения подписи на странице. После этого к полученной функции можно применить стандартные методы исследования. Исходную подпись с точностью до ее положения можно восстановить по построенной функции;
2. в качестве снимаемых параметров используется решение системы уравнений, дающее коэффициенты представления функции подписи через радиальный базис.

Представление подписи в виде функции

Переход от самой подписи к ее характеристике должен сохранять ее свойства, но не зависеть от положения подписи и ее ориентации. On-line подпись (рис. 1) представляется в виде текстового файла, каждая запись которого есть вектор $p_k=(x_k, y_k)$, заданный координатами очередной точки (сила нажима не учитывалась). Поскольку положение очередной точки определялось через равные интервалы времени, в указанном описании учитывалась динамика подписи. Файл преобразовывается в значения функции $f(t)$ по следующему правилу: строится последовательность чисел t_k , $k=0, 1, \dots$, где $t_0=0$, а $t_k=|p_{k+1}-p_k|$, $k>0$. Далее, $f_0=0$, $f_k=f(t_k)$ – есть угол между векторами p_k и p_{k+1} (рис. 2, рис. 3). Очевидно, что исходная подпись может быть восстановлена по указанным значениям функции с точностью до положения подписи на плоскости.

Идентификация подписи с помощью радиальных функций

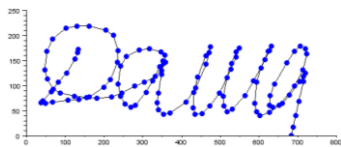


Рис. 1. Изображение подписи

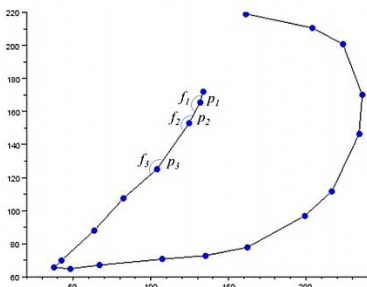


Рис. 2. Фрагмент подписи

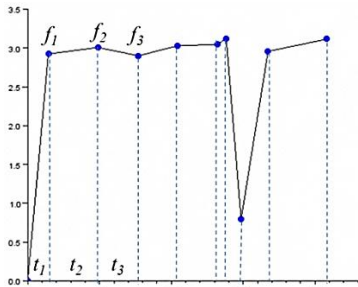


Рис. 3. Представление подписи в виде функции

Подписи одного и того же человека не являются совершенно идентичными. В частности, построенные функции будут иметь разные области определения (рис. 4). Чтобы получить возможность в дальнейшем сравнивать такие функции, функции подписи нормализуют (рис. 5). Для этого область определения разбивается на одно и то же число (N) точек: $x_1, x_2, x_3, \dots, x_N$, равноудаленных друг от друга, а значения функции в этих точках находятся с помощью линейной интерполяции.

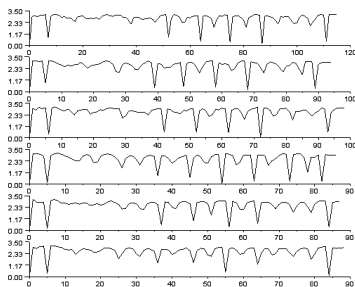


Рис. 4. Функции подписи

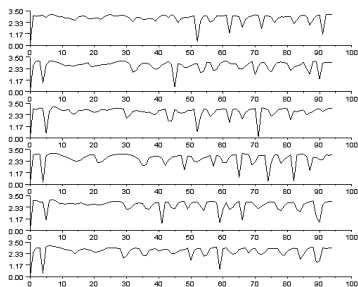


Рис. 5. Нормализованные функции подписи

Применение радиальных функций и вейвлетов для описания подписи

Формально, полученные значения функций можно использовать для описания подписи, однако, индивидуальность проявляется в зависимости значений этих функций в разных точках. С этой целью для отыскания такой зависимости был использован подход, представленный в [5]. Этот подход основывается на применении радиальных функций и вейвлетов к исследованию функции подписи.

Радиальная функция – это функция $\rho(x)$, зависящая только от расстояния между x и фиксированной точкой пространства X .

Аппроксимация (приближение) радиальной базисной функцией имеет вид:

$$\sum_{k=1}^N a_k \rho(\|x_j - x_k\|) = f_j \quad (1)$$

где: $\|x_j - x_k\|$ - евклидово расстояние между узлами x_j и x_k ;

$\rho(\|x_j - x_k\|)$ – функция, зависящая только от расстояния до соответствующего узла x_k и поэтому называемая радиалом;

a_k – весовые коэффициенты;

$f_j = f(x_j)$ – значение функции (величина угла) в точке x_j .

Видим, что (1) представляет собой свёртку вейвлет-функции $\rho(x)$ (в качестве вейвлет-функции взята радиальная функция) с сигналом a_k , а значит это дискретное вейвлет-преобразование. Оно переводит исходную функцию в форму, которая делает некоторые ее величины более поддающимися изучению, позволяет получить высокое соотношение сжатия в сочетании с хорошим качеством восстановления сигнала.

Возьмём каждый экземпляр подписи, построим и решим для него систему линейных уравнений (1):

$$\begin{cases} a_1 \rho(0) + a_2 \rho(\|x_1 - x_2\|) + \dots + a_N \rho(\|x_1 - x_N\|) = f(x_1), \\ a_1 \rho(\|x_2 - x_1\|) + a_2 \rho(0) + \dots + a_N \rho(\|x_2 - x_N\|) = f(x_2), \\ \dots \\ a_1 \rho(\|x_N - x_1\|) + a_2 \rho(\|x_N - x_2\|) + \dots + a_N \rho(0) = f(x_N). \end{cases} \quad (2)$$

В качестве $\rho(x)$ согласно [5] возьмём симметричную (радиальную) центральную базисную функцию

$$\rho(x) = |x|^\alpha \quad (3)$$

где: $0 < \alpha < 1$

Из каждой системы уравнений определим коэффициенты $\{a_k\}$ разложения через радиальный базис и построим их графики (рис. 6):

Идентификация подписи с помощью радиальных функций



Рис. 6. Графики коэффициентов $\{a_k\}$ при $0 < \alpha < 1$ ($\alpha = 0.3$)

На графиках $\{a_k\}$ экземпляров подписи одного автора наблюдается определённая закономерность и устойчивость значений в некоторых точках.

Кроме того, получаемая матрица коэффициентов позволяет восстановить исходные подписи. Наиболее значимая информация содержится при высоких амплитудах, а менее полезная – при низких, поэтому за счет отбрасывания низких амплитуд возможно сжатие данных. Возникает вопрос, в каких точках значения $\{a_k\}$ имеют наибольшую амплитуду? Для этого отметим на подписи точки с наибольшим значением a_k (рис. 7):

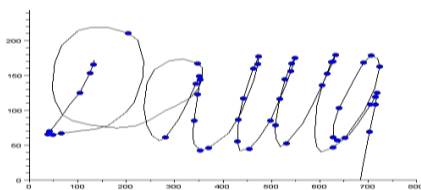


Рис.7. Точки с наибольшей амплитудой $\{a_k\}$

Здесь отмечено менее половины точек исходной подписи. Тем не менее, видим, что большинство отмеченных точек с наибольшим значением амплитуды – особые. Значит действительно, в высоких амплитудах содержится наиболее значимая информация о подписи. Тогда для сжатия можно отбросить значения коэффициентов $\{a_k\}$ с низкой амплитудой и оставить коэффициенты $\{a_k\}$ с высоким значением амплитуды.

Идентификация подписи с помощью радиальных функций

А теперь для сравнения возьмём $\rho(x)=|x|^\alpha$, где $\alpha=1$, т.е. $\rho(x)=|x|$ (рис. 8):

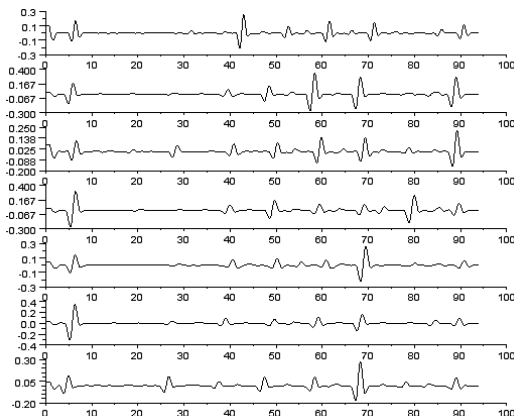


Рис. 8. Графики коэффициентов $\{a_k\}$ при $\alpha=1$

В этом случае закономерность также прослеживается, но не так чётко как при $0 < \alpha < 1$.

Введём метрику для коэффициентов $\{a_k\}$ с условием (3). Для этого первоначально определим средние значения $\{a_k\}$ для всех экземпляров подписи автора. Далее вычислим расстояние от «усреднённого» вектора $\{a_k\}$ до вектора коэффициентов $\{a_k\}$ каждого экземпляра (рис. 9):

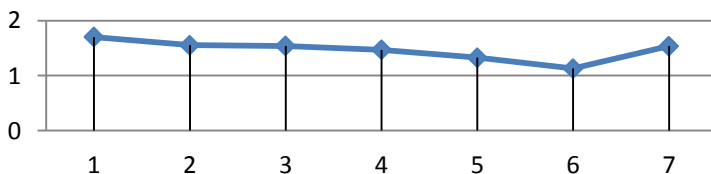


Рис.9. Расстояния от «усреднённого» вектора $\{a_k\}$ до вектора коэффициентов $\{a_k\}$ каждого экземпляра

Среднее значение расстояний равно 1.6003623.

Рассмотрим результаты исследования подписей следующего участника эксперимента (рис. 10):

Идентификация подписи с помощью радиальных функций

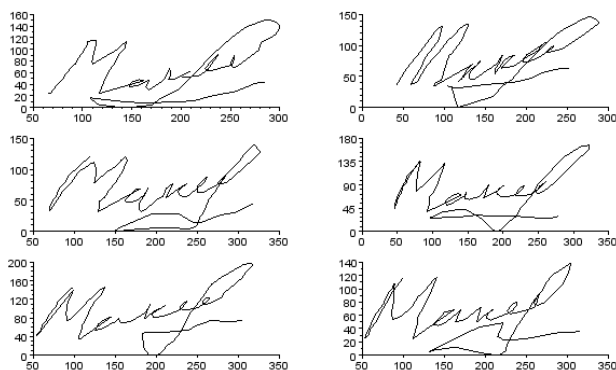


Рис. 10. Экземпляры подписи

Определяем коэффициенты $\{a_k\}$, строим их графики с радиальной функцией (3) (рис. 11).

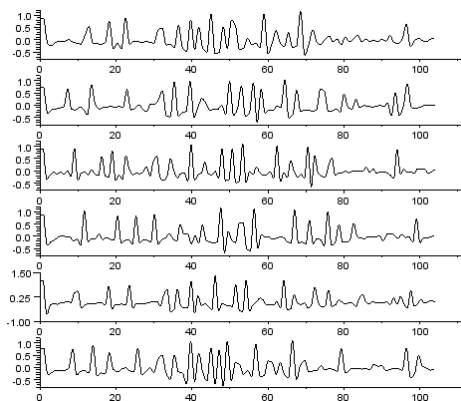


Рис. 11. Графики коэффициентов $\{a_k\}$ при $0 < \alpha < 1$ ($\alpha=0.3$)

Видим, что закономерность и устойчивость также имеют место.

Вычислим расстояние между «усреднёнными» векторами $\{a_k\}$ экземпляров подписей первого и второго авторов. Это расстояние равно 5.0454262. Таким образом, расстояние между функциями подписей, выполненных разными людьми, превышает расстояние между функциями подписей одного автора. Этот факт открывает возможность для идентификации подписей путём применения вейвлет-преобразований и радиальных функций и дальнейшим сравнением их с учётом вводимой метрики. Можно ввести определённый порог при сравнении расстояний между векторами подписей. Если расстояние между векторами подпи-

сей А и В будет ниже порога, то принимается решение об идентичности подписей А и В, а если расстояние между векторами подписей А и В будет выше порога, то считается, что они принадлежат разным авторам.

Результаты эксперимента

Эксперимент по идентификации подписи с помощью радиальных функций проводился на 5 участниках (А, В, С, D, E). У каждого участника эксперимента было использовано для исследования по 6-7 экземпляров подписей.

Подписи участников были представлены одномерными функциями с помощью углов. Решением систем уравнений получены векторы коэффициентов $\{a_k\}$ разложения функций через радиальный базис для каждой подписи. Для проведения идентификации подписей были вычислены расстояния между полученными векторами коэффициентов. Приведём результаты идентификации первого участника (А) (рис. 1). Для этого вычислим расстояния между вектором его усреднённой подписи и векторами остальных подписей (его и других участников). Для представления результатов идентификации построим ROC-кривую (Receiver Operator Characteristic) (рис. 12).

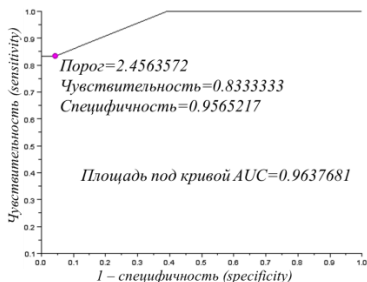


Рис. 12. ROC-кривая участника А

Видим, что площадь под построенной кривой AUC (Area Under Curve) равна 0.9637681, что говорит о неплохом качестве построенной модели. Значение порога для принятия решения найдём исходя из баланса между чувствительностью и специфичностью. Получим: порог = 2.4563572, чувствительность = 0.8333333, специфичность = 0.9565217. При этом ошибка первого рода равна 0.1666667, ошибка второго рода 0.0434783.

Аналогично результаты идентификации остальных участников эксперимента представлены с помощью ROC-кривых на рис. 13-16.

Идентификация подписи с помощью радиальных функций

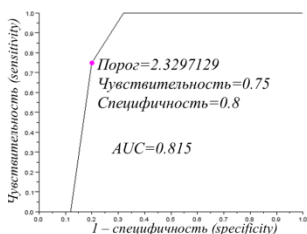


Рис. 13. ROC-кривая участника В

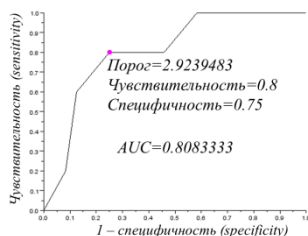


Рис. 14. ROC-кривая участника С

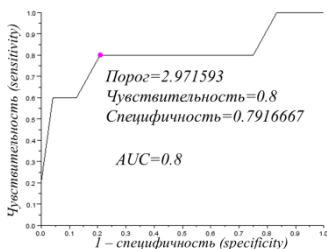


Рис. 15. ROC-кривая участника D

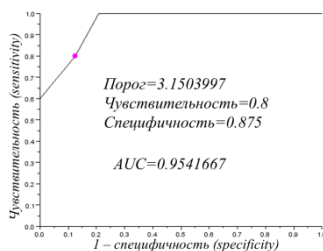


Рис. 16. ROC-кривая участника Е

Среднее значение ошибки первого рода при идентификации подписей, участвовавших в эксперименте, составило 0.2033333, среднее значение ошибки второго рода составило 0.1653623.

Выводы

1. Проведён анализ подписи как биометрической характеристики с учётом динамики. Применение в качестве вейвлет-функций радиалов позволило определить закономерности функций подписи среди экземпляров подписей каждого автора при разложении их через радиальный базис.

2. Новизна предлагаемого метода: 1) подпись представляется в виде функции от одного аргумента инвариантной относительно положения подписи на странице; 2) в качестве снимаемых параметров используется решение системы уравнений, дающее коэффициенты представления функции через радиальный базис. Радиальные функции (3) в качестве вейвлет-функций позволяют ослабить шумы и усилить существующие закономерности.

3. Получаемая матрица коэффициентов $\{a_k\}$ позволяет восстановить исходные подписи. Наиболее значимая информация (информация об особых точках) содержится при высоких амплитудах, а менее полезная – при низких. За счет отбрасывания низких амплитуд возможно сжатие данных.

4. Расстояние, вычисленное между функциями подписей разных авторов, превышает расстояние между функциями подписей одного автора, что открывает возможности для идентификации подписей. Для представления результатов идентификации для каждого участника эксперимента были построены ROC-кривые.

5. Предложенный способ не отрицает имеющиеся методы распознавания, он является дополнительной альтернативой при определении результатов распознавания подписи.

Список источников

1. Воронцов К.В. Машинное обучение (курс лекций, К.В. Воронцов) [Электронный ресурс] // Информационно-аналитический ресурс, посвящённый машинному обучению, распознаванию образов и интеллектуальному анализу данных. [2011–]. Дата обновления: 26.02.2013. URL: [http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_\(%D0%BA%D1%83%D1%80%D1%81_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9,_%D0%9A.%D0%92.%D0%92%D0%BE%D1%80%D0%BE%D0%BD%D1%86%D0%BE%D0%B2\)](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BA%D1%83%D1%80%D1%81_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9,_%D0%9A.%D0%92.%D0%92%D0%BE%D1%80%D0%BE%D0%BD%D1%86%D0%BE%D0%B2)) (дата обращения: 05.03.2013)
2. Иванов А.И. Нейросетевые алгоритмы биометрической идентификации личности. Кн. 15: [монография]. М.: Радиотехника, 2004 . 144 с.
3. Идентификация личности по рукописной подписи и динамике ее воспроизведения [Электронный ресурс] // Информационный сайт по проблемам защиты информации Разведка.ru. 2010. URL: <http://www.razvedka.ru/catalog/582/609/15634.htm> (дата обращения: 08.02.2013).
4. Леус А.В. Биометрическая аутентификация по динамическим характеристикам подписи [Электронный ресурс] // "Secuteck.ru" - о системах безопасности: Каталог "СКУД. Антитерроризм". 2009. URL: http://www.secuteck.ru/articles2/sys_ogr_dost/biometrigh-autentifikac-podinamich-harakter-podpisi (дата обращения: 07.02.2013).
5. Unser M., Blu T. Wavelets, Fractals, and Radial Basis Functions // IEEE Transactions on signal processing. 2002. Vol. 50, № 3. P. 543-553.

Сравнение онлайн-сообществ на основе лексического анализа ленты новостей

Д. А. Усталов¹, Ф. В. Краснов², Р. Э. Яворский³

¹ dau@imm.uran.ru, ² fk@sk.ru, ³ ryavorsky@sk.ru

¹ ИММ УрО РАН, Екатеринбург, Россия

^{2,3} Фонд Сколково, Москва, Россия

Аннотация. Предложен подход к формальной оценке характеристик онлайн-сообществ путём лексического анализа содержания ленты новостей. Описаны результаты экспериментов по обработке содержания новостных лент трёх разных онлайн-сообществ: IT-EBURG, GIS-Lab, Сколково. Такие параметры, как количество терминов и доля профессиональной лексики в новостях, рассматривается как важная характеристика культурной среды исследуемого онлайн-сообщества.

Ключевые слова: извлечение ключевых слов; профессиональное сообщество; онлайн-сообщество; характеристики онлайн-сообществ.

Введение

Онлайн-сообщество — это группа людей, объединённых на базе Web-сервиса, предоставляющего функциональность онлайн-дискуссии и совместной работы.

Несмотря на внешнюю схожесть, онлайн-сообщества заметно отличаются друг от друга по целям, внутренней структуре, принятому стилю общения, и другим характеристикам.

Разработка методов и инструментов для определения основных характеристик онлайн-сообщества является интересной и востребованной задачей [1, 2, 3].

1 Цели и задачи работы

Цель работы состоит в том, чтобы найти формальные (с прозрачным алгоритмом расчёта) параметры онлайн-сообществ, характеризующие стиль общения, принятый среди его участников.

В рамках данного исследования взяты тридцать последовательных текстов за осень 2012 г. из ленты новостей трёх разных онлайн-сообществ:

- IT-EBURG — локальное сообщество IT-специалистов Екатеринбурга [4];
- GIS-Lab — профессиональное сообщество исследователей и разработчиков геоинформационных систем [5];
- онлайн-сообщество инновационного центра Сколково [6].

Выбор данного временного периода обусловлен необходимостью минимизировать зашумлённость данных новогодними праздниками и летними отпусками.

2 Реализация

Из каждой новости автоматически извлечены ключевые слова и словосочетания при помощи инструментария, описанного в [7].

Полученные ключевые слова и словосочетания были размечены экспертом на два класса: 1) общая лексика; 2) профессиональная лексика.

Для каждого текста посчитано количество ключевых слов и вычислена доля профессиональной лексики.

Стоит отметить, что в каждом сообществе, независимо от профессиональной принадлежности, в новостях встречаются объявления общего типа: информация о мероприятиях, анонсы, отчёты, и т. д.

Для компенсации этой особенности из выборки были исключены тексты, в которых доля профессиональной лексики не превышает 10%. Таких текстов оказалось всего 13 для IT-EBURG, 19 для GIS-Lab, 14 для Сколково.

3 Результаты и их анализ

Итоговые результаты отражены на схеме (рис. 1), где каждому маркеру соответствует одна запись в новостной ленте. Также отобра-

жены маркеры для усредненного значения параметров по каждому сообществу.

На схеме видно, что сложившиеся специализированные сообщества (IT-EBURG и GIS-Lab) отличаются большой долей профессиональных терминов и небольшим количеством ключевых слов, входящих в текст новостей. В сообществе Сколково явным образом выделяется большое количество терминов общей лексики.

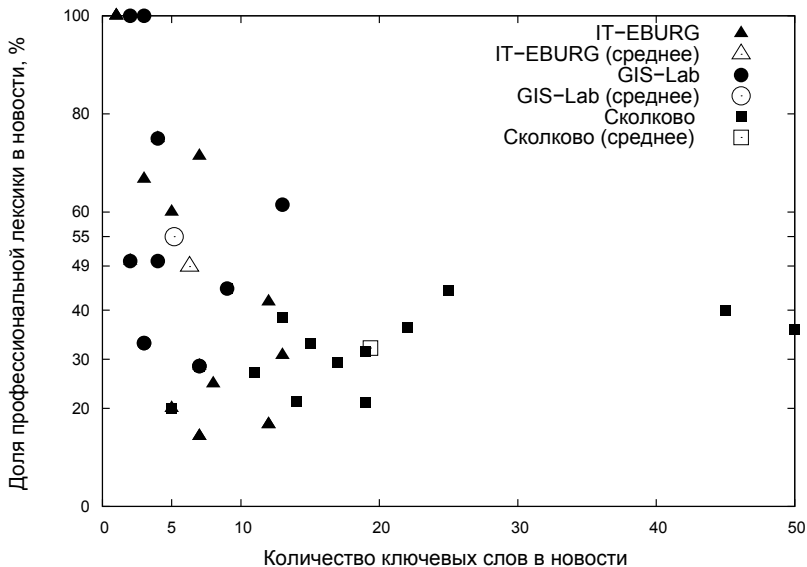


Рис. 1. Результаты экспертной оценки текстов новостей трёх онлайн-сообществ

Заключение

Предложен подход формальной оценки онлайн-сообщества путём лексического анализа новостных лент.

В этом направлении сделаны первые шаги. Отчасти подтвердилась гипотеза о том, что в качестве одного из таких параметров можно рассматривать долю профессиональной лексики в ключевых словах новостной ленты сообщества.

Для устоявшихся профессиональных сообществ этот параметр близок к 50% (49% для IT-EBURG и 55% для GIS-Lab), для проходящего стадию формирования онлайн-сообщества Сколково это параметр заметно ниже и составляет в среднем 30%.

Главным недостатком реализации предлагаемого подхода является наличие этапа ручной разметки, который ограничивает возможности масштабирования подхода.

В качестве следующего шага целесообразно использовать доступные средства автоматической тематической классификации документов [8].

Список источников

1. Yavorskiy R. Research Challenges of Dynamic Socio-Semantic Networks // The International Workshop on Concept Discovery in Unstructured Data, CDUD 2011. P. 119–121.
2. Краснов Ф.В. Развитие через общение // Intelligent Enterprise. 2012. Т. 9, С. 18–21.
3. Краснов Ф.В., Яворский Р.Э. Измерение уровня зрелости профессионального сообщества // Принята к публикации в журнале «Бизнес-информатика». 2013.
4. Информационные технологии в Екатеринбурге [Интернет-портал]. URL: <http://it-eburg.com/> (дата обращения: 05.02.2013).
5. GIS-Lab: Геоинформационные системы и Дистанционное зондирование Земли [Интернет-портал]. URL: <http://gis-lab.info/> (дата обращения: 05.02.2013).
6. Инновационный центр Сколково [Интернет-портал]. URL: <http://community.sk.ru/press/> (дата обращения: 05.02.2013).
7. Усталов Д.А. Извлечение терминов из русскоязычных текстов при помощи графовых моделей / в сб. «Теория графов и приложения = Graphs theory and applications: материалы конференции». Екатеринбург: Изд-во Урал. ун-та, 2012. С. 62–69.
8. Агеев М.С. Экспериментальные алгоритмы поиска/классификации и сравнение с “basic line” / М.С. Агеев, Б.В. Добров, Н.В. Лукашевич, А.В. Сидоров // Российский семинар по оценке методов информационного поиска (РОМИП 2004). Санкт-Петербург: НИИ Химии СПбГУ, 2004. С. 62–89.

Научное издание

Доклады по компьютерным наукам
и информационным технологиям
№ 2, 2013 г.

Доклады всероссийской научной конференции АИСТ'12
Екатеринбург, 4 – 6 апреля 2013 года
«Модели, алгоритмы и инструменты анализа данных;
результаты и возможности для анализа изображений,
сетей и текстов»

Редакторы Ольга Баринава, Дмитрий Игнатов, Михаил Хачай

Ответственный редактор Екатерина Черняк

Компьютерная верстка Б. Агафонцев

Дизайн обложки Ю. Васильев

Подписано в печать 06.03.2012. Формат 60x90/16.

Гарнитура «Таймс». Бумага офсетная. Печать офсетная.

Усл. печ. л.26,25. Тираж 500 экз. Заказ № 975

Национальный Открытый Университет «ИНТУИТ»

Москва, Электрический пер., 8, стр.3.

Телефон: +7 (499) 253-9312, 253-9313, факс: +7 (499) 253-9310

E-mail: info@intuit.ru, <http://www.intuit.ru>

Анализ изображений, сетей и текстов

**Модели, алгоритмы и инструменты анализа данных;
результаты и возможности для анализа изображений, сетей
и текстов**

Доклады Всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ, Екатеринбург, 2013). Рассматриваются проблемы в области компьютерного зрения, анализа изображений и видео, анализа форумов, блогов и социальных сетей, анализ сетевых (графовых) и потоковых данных, компьютерной обработки текстов, гео-информационных систем, математических моделей и методов анализа данных, машинного обучения и разработки данных (Data Mining), рекомендательных систем и алгоритмов, Semantic Web, онтологии и их приложений. Для студентов, аспирантов и специалистов в области машинного зрения, анализа изображений, текстов, социальных сетей и других неструктурированных данных.

**Доклады по компьютерным наукам
и информационным технологиям 02**
www.LectureNotes.ru

ISBN 978-5-9556-0148-9



9 785955 601489 >