Санкт-Петербургский государственный университет
Кафедра математической лингвистики

Направление: «Лингвистика»

Образовательная программа: «Прикладная и экспериментальная лингвистика»

Профиль: «Компьютерная лингвистика и интеллектуальные технологии»

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ ИЗВЛЕЧЕНИЯ ИМЕНОВАННЫХ СУЩНОСТЕЙ

Выпускная квалификационная работа
соискателя на степень магистра филологии

Крастынь Валерии Валерьевны

Научный руководитель
к.ф.н., доц. Хохлова М.В.

Санкт-Петербург
2017

Содержание:

Generating Table of Contents for Word Import ...

Введение

Целью данного исследования является разработка системы для извлечения именованных сущностей из текстов микроблогов (Твиттер) на русском языке на основе анализа существующих методов и инструментов извлечения именованных сущностей.

Для решения заявленной цели были поставлены следующие задачи:

- исследовать существующие методы извлечения именованных сущностей;

- изучить особенности предметной области – текстов микроблогов;

- выбрать и доработать наиболее подходящие инструменты для анализа выбранной предметной области;

- собрать и разметить корпус текстов микроблогов;

- провести эксперименты на корпусе микроблогов и на фоновом корпусе новостных текстов;

- сравнить результаты по двум корпусам и по двум инструментам;

- сделать выводы о результативности систем и дальнейших направлениях работы.

Актуальность работы обусловлена как широким применением методов извлечения именованных сущностей в различных задачах прикладной лингвистики, так и особенностями предметной области. Выделение именованных сущностей является одной из важных задач автоматической

обработки текста. Это обязательный этап во многих системах извлечения структурированной информации из неструктурированных данных: в задачах информационного поиска, при построении вопросно-ответных систем, автоматизированнм сборе и аннотировании новостей, анализе биологических и медицинских текстов. Извлечение именованных сущностей из текстов микроблогов находит применение в первую очередь в системах, используемых для анализа отзывов о товаре и упоминания бренда в сети.

Новизна исследования состоит в выборе и адаптации инструментов извлечения именованных сущностей к конкретному материалу исследования – текстам микроблога Твиттер на русском языке.

Практическая значимость исследования состоит, во-первых, в создании размеченного корпуса текстов микроблогов на русском языке; во-вторых, в экспериментальной оценке и сравнении результатов различных систем извлечения именованных сущностей. Полученные результаты могут быть использованы для дальнейшего совершенствования инструментов выделения именованных сущностей.

1. Особенности предметной области

По данным исследования Риттера [Ritter et al. 2011: 30] каждый день появляется более 100 миллионов новых сообщений в Твиттере. Социальные сети формата микроблога продолжают набирать популярность, в то время как более привычные текстовые блоги отошли на второй план. При подобном бурном росте объема свободно доступных современных текстов на множестве языков Твиттер, несомненно, является одним из важнейших источников данных для задач прикладной лингвистики. В то же время, ряд специфических характеристик данных заставляет исследователей искать нетрадиционные подходы к извлечению информации и в частности, к выделению именованных сущностей.

Работа Риттера и соавторов [30] показала характерные особенности текстов Твиттера, затрудняющие их обработку классическими средствами

АОТ. Две основные причины затруднений: во-первых, при обилии в сообщениях-твитах различных ИС (названий компаний, продуктов, музыкальных групп, кинофильмов, сериалов и т.д.), почти все типы, кроме имен людей (Персона) и географических локаций (Локация) встречаются относительно редко, и таким образом даже большой корпус вручную размеченных твитов будет содержать недостаточно сущностей каждого типа для обучения модели. Нехватка должным образом размеченных корпусов является важным сдерживающим фактором развития моделей, основанных на методах машинного обучения в этой сфере. В данной работе исследователи собрали корпус из 2400 случайных твитов и разметили 10 типов сущностей в нем.

Во-вторых, в связи с ограничением в 140 символов, твиты не всегда обладают достаточным контекстом, позволяющим с уверенностью определить тип ИС даже эксперту-аннотатору. Кроме того, тексты Твиттера зачастую не позволяют с уверенностью использовать один из классических признаков для выделения ИС – паттерн капитализации, так как многие сообщения носят отрывистый, телеграфный характер, и их авторы не считают необходимым соблюдение принятых шаблонов капитализации. В силу того же ограничения длины сообщений и их особого формата может быть также затруднен синтаксический анализ (многие сообщения представляют собой неполные предложения, не встречающиеся в более формальных текстах). В дополнение ко всему вышесказанному, тексты Твиттера содержат гораздо большую долю аббревиатур, сленговых сокращений и орфографических ошибок, чем любой другой жанр.

Ссылаясь на сложную, полную «шумов» природу микроблогов, Шерман Малмази и Марк Драс [Malmasi, Dras, 2016: 47] предлагают для выделения в тексте упоминаний локаций опираться на поиск именных групп и n-граммы взамен традиционного подхода к извлечению именованных сущностей.

Леон Держински в работе [Derczynski et al., 2013: 35] также отмечает, что тексты Твиттера сопровождаются обилием метаданных (время, место

написания), которые могут дать ключ к некоторым задачам семантической разметки.

В своем обзоре [Derczynski et al., 2014: 42] Держински отмечает помимо прочего, что сами сущности, упоминаемые в Твиттере отличаются от тех, что часто встречаются в новостных текстах. Если говорить о категории «Персона», то в то время как в новостях в нее попадают в основном политики, журналисты и представители бизнеса, микроблоги чаще говорят о спортсменах, актерах, персонажах кино и сериалов, а также о частных лицах – друзьях, родных. Для «Локаций» частотными в новостях будут названия стран, рек, городов, в Твиттере же часто говорят также ресторанах, барах, местных достопримечательностях – небольших объектах. То же характерно и для упоминаний организаций: вместо доминирования крупных в терминах капитализации и кадрового состава, международных или государственных организаций/корпораций, мы также можем часто встретиться с названиями музыкальных коллективов, небольших компаний, стартапов, спортклубов, как общеизвестных, так и местных.

Для твитов также характерно более частое по сравнению с новостями упоминание названий продуктов (примерно в 5% сообщений).

Таким образом, в условиях многообразия представленных типов сущностей усложняется задача обнаружения и классификации сущностей, не представленных на этапе обучения (или написания правил). Это негативно сказывается на результатах различных подходов, основанных как на газеттирах, так и на методах машинного обучения.

Кроме того, как отмечает Держински, для социальных сетей (и микроблогов в частности) характерно явление «смещения» (“drift”): набор сущностей, широко представленных в текстах микроблогов существенно меняется со временем. В результате система, подготовленная и обученная на корпусе текстов определенного временного периода может хорошо справляться с другими текстами того же периода, но с течением времени результаты станут менее впечатляющими.

О проблемах ненормативного написания в Твиттере и, как следствие, появления большого количества слов, не входящих в словарь (“out-of-vocabulary” - OOV), что делает менее эффективными все этапы обработки текста, также говорят Бо Хан и Тимоти Болдуин [Han, Baldwin, 2014: 24]. Авторы предлагают каскадный метод выявления и нормализации неверно написанных (“ill-formed”) слов, основанный на морфологической и фонетической близости. Однако авторы также признают, что лучшие результаты может дать подход, сочетающий данный спеллчекер с обширным словарем замен и списком «белых» OOV-слов – не нуждающихся в замене.

2. Основные методы извлечения ИС

2.1 Подходы к извлечению именованных сущностей

Именованные сущности — это объекты определенного типа, чаще всего составные, например, названия организаций, имена людей, даты, места, денежные единицы и.т.д. В зависимости от прикладных задач, может быть необходимо выделить в тексте, во-первых, имена собственные: имена лиц, топонимы, названия организаций, названия песен и исполнителей, названия товаров и брэндов; во-вторых, такие объекты как числа, даты, денежные единицы. Наибольшее распространение для широкого спектра задач получила выделение таких сущностей, как Персона (Per) – имена, фамилии, отчества людей; Локация (Loc) – топонимы; Организация (Org) – названия организаций, компаний, объединений; Разное (Misc) – в эту группу входят все прочие типы сущностей, если их более тщательное разделение не требуется для целей исследования.

Термин «именованная сущность» впервые был введен на шестой Конференции по Пониманию Сообщений (Message Understanding Conference, MUC-6) в 1996 году. MUC-6 и предшествующие ей Конференции по Пониманию Сообщений были посвящены задачам по извлечению информации: получение структурированной информации о компаниях и

военных операциях из неструктурированных текстов, как, например, газетных статей и военных сводок.

При постановке задачи по извлечению информации было замечено, что необходимо различать такие информационные единицы, как персона, организация, локация и числовые выражения, включающие в себя время, даты, деньги и проценты. Идентификация в тексте данных сущностей была признана одной из важнейших подзадач извлечения информации и была названа извлечение именованных сущностей.Одной из первых работ в данной области принято считать статью Лизы Pay [Rau, 1991: 12]. Она предложила использовать эвристические подходы и набор правил для выделения названий компаний в тексте. В случае невозможности создания обучающей выборки, данный метод является единственным возможным для решения задачи извлечения именованных сущностей. С тех пор за 26 лет исследований было предложено огромное количество решений и стратегий по извлечению имен. Задача была представлена не секциях различных конференций: Message Understanding Conference (MUC), Conference on Natural Language Learning (CoNLL), International Conference on Language Resources and Evaluation2 (LERC).

Большая обзорная работа была проделана Дэвидом Надю и Сатоши Секином [Nadeau, Sekine, 2006: 8]. Авторы подробно рассмотрели методы, используемые в области выделения и классификации сущностей с 1991 по 2006 года. В данной же работе мы ограничимся основными моментами, необходимыми для общего понимания подходов к решению задачи, и постараемся дополнить упомянутый обзор.

В работе [Brykina al., 2013: 5] рассматривается словарный подход для разрешения омонимии в задаче извлечения именованных сущностей. Разработанная система получает на вход список сущностей, интересных пользователю. При помощи существующих онтологий, в которых в структурированном виде отражена информация об объектах и их отношениях, из текстов извлекаются заданные сущности. Задачей системы является извлечение всевозможных верных синонимов интересующих

пользователя объектов. Как следствие, в работе рассматриваются различные случаи омонимии внутри типов именованных сущностей (Персон, Локаций и Организаций). Настраиваемый список позволяет пользователю регулировать предметную область извлекаемых сущностей под свои информационные запросы. Результатом работы системы является высокая оценка точности извлечения именованных сущностей.

Целью работы [Popov et al., 2004: 10] является адаптация для русского языка многоязыкового проекта MUSE [Maynard et al., 2003: 7], основанного на извлечении англоязычных именованных сущностей. Проект создан на основе правил с использованием справочников сущностей: крупнейшие компании, субъекты федерации, главные лица государства, известные персоны, распространенные имена мужчин и женщин, фамилии, названия месяцев.

В то время как первые исследования главным образом были основаны на созданных вручную правилах, последние работы используют методы машинного обучения с учителем. Они создают автоматически регулируемые системы, основанные на алгоритмах разметки данных, которые получают из обучающей коллекции документов.

Например, работа [Нехай, 2012: 3] использует метод опорных векторов в применении буквенных n-грамм и других статистик уровня символов и слов для задачи извлечения имён собственных.

В работе [Глазова, 2010: 2] для решения задачи извлечения имён собственных из текстов на английском языке используется метод максимальной энтропии, для которого характеристические функции представлены перечислением специальных предшествующих слов (mr., chairman и другие), наличием после словосочетания-кандидата глагола, частотой встречаемости слова в документах, присутствием аббревиатур и прочее.

В [Nigam et al., 1999: 9] для решения проблемы переобучения вводится использование априорного распределения модели (Гауссово распределение) для классификации текстов на естественном языке.

В работе [Антонова, Соловьев, 2013: 1] для анализа текстов на русском языке (задача распознавания именованных сущностей, определения частей речи и анализа отношения (положительного / отрицательного) к объекту) использован метод условных случайных полей. Как замечают авторы, данный метод позволяет решить проблему смещения метки (label bias problem), возникающую в методе максимальной энтропии.

Статья [Подобряев, 2013: 4] использует метод условных случайных полей для поиска упоминаний персон в новостных текстах. Помимо признаков уровня слова (прописные буквы и знаки препинания внутри слова-кандидата), используются также признаки контекста и онтологическая и фактографическая информации о слове-кандидате.

В работе [McCallum et al., 2003: 53] предложена новая категория систем для извлечения именованных сущностей, основанная на методе частичного обучения (semisupervised learning). Основной техникой данного метода является самообучение с использованием статистического бутстрэпа (bootstrapping), который включает в себя небольшую долю обучения с учителем, например, набор начальных данных для старта процесса обучения. Рассмотрим пример работы системы, направленной на извлечение названий болезней. В первую очередь, она получает небольшой список примеров таких названий. Затем система ищет предложения, которые содержат данные примеры, и пытается выявить некоторые общие признаки для известных примеров. После этого система пытается отыскать другие названия болезней, появляющиеся в аналогичных контекстах. Процесс обучения повторяется вновь для извлеченных сущностей, чтобы отыскать новые признаки искомых. По завершению нескольких итераций представляется список болезней и большое количество их контекстов.

В 1999 году был полностью разобран корпус текстов, содержащий около 90 000 именованных сущностей, в поисках шаблонов для такой модели [Collins, Singer, 1999: 6]. Примером такого шаблона может являться имя собственное со следующей за ней именной группой (например, «Mr. Cooper, a vice president of …»). Шаблоны хранятся в паре «написание слова -

контекст», где «написание» включает в себя именованную сущность, а «контекст» - именную группу в его контексте. Для кандидатов, удовлетворяющих правилу «написание», определяется их тип именованной сущности, и их «контексты» накапливаются в рамках каждого типа. Затем наиболее частые контексты превращаются в набор контекстных правил. После выполнения этих действий контекстные правила могут быть использованы для нахождения новых именных сущностей, не включенных в начальный список сущностей.

Работа демонстрирует, что при одновременном обучении нескольким типам именованных сущностей происходит выделение так называемого «негативного примера» - класса, выступающего в роли один против всех, который сокращает чрезмерную генерацию шаблонов. Хотя данный метод требует минимального набора обучающих данных, что, несомненно, является большим преимуществом, основным недостатком обучения с использованием метода частичного обучения является чрезмерная генерация шаблонов, которая для точных и полных результатов требует валидации экспертом.

Победители соревнования по NER CoNLL 2003 [Florian et al. 2003: 59], получившие 88.76% F1, представили систему использующую комбинацию различных алгоритмов машинного обучения. В качестве признаков был использован их собственный, вручную составленный газетир, POS-теги, CHUNK-теги, суффиксы, префиксы и выход других NER-классификаторов, тренированных на внешних данных.

Нейронные сети для выделения именованных сущностей.

Коллобер и соавторы [Collobert et al., 2011: 41] представили комбинацию сверточной нейронной сети с условными случайными полями, получившую 89.59% F1 на корпусе CoNLL 2003. Их нейросетевая архитектура не зависит от задачи и используется как для NER, так и для частеречной разметки (part-of-speech tagging), поиска синтаксически связанных групп соседних слов (chunking), установления семантических ролей (semantic role labelling). Для задачи NER они использовали три типа признаков - векторное представление

слова, капитализацию и небольшой газетир, включенный в соревнование CoNLL 2003.

[Chiu, Nichols, 2015: 60] представили комбинацию сверточных сетей, рекуррентных сетей и условных случайных полей. Они использовали такие же признаки как и в [41], дополнительный, вручную сформированный газетир на основе DBpedia и обучались на train+dev1 выборке CoNLL 2003. У них получилось 91.62% F1. Кроме корпуса CoNLL 2003 они тестировали архитектуру на более крупном англоязычном корпусе OntoNotes 5.0. На нем они получили state-of-the-art результат 86.28%.

[Yang et al. 2016: 61] представили глубокую иерархическую рекуррентную нейросетевую архитектуру с условными случайными полями для разметки последовательностей. Они использовали такие же признаки как в работе [41]. Кроме англоязычного корпуса CoNLL 2003, где они получили state-of-the-art 90.94% F1 при обучении только на обучающей выборке (train set), они тестировали работу нейросети на CoNLL 2002 Dutch NER и CoNLL 2003 Spanish NER. На этих корпусах они улучшили предыдущий state-of-the-art результат: 82.82% до 85.19% на CoNLL 2002 Dutch NER и 85.75% до 85.77% на CoNLL 2003 Spanish NER.

Современные работы используют векторное представление слов и условные случайные поля в своих моделях. Из сторонних признаков применяют только газетиры. В работе [Xu et al. 2014: 62] описано применение дополнительных признаков для слов (морфологических, синтаксических, семантических) для создания более совершенных векторных представлений. Такие векторные представления помогают повысить оценку качества в прикладных задачах [62].

Что касается методов, применяемых для извлечения ИС из текстов микроблогов, Леон Держински в работе [Derczynski et al. 2014: 42] дает достаточно развернутый обзор современных систем и их результатов на корпусе из 4264 твитов объемом в 29089 токенов на английском языке, созданном в рамках конкурса Making Sense of Microposts 2013 Concept Extraction Challenge.

В таблице 1 представлены основные характеристики некоторых систем, проанализированных в исследовании Держински, в таблице 2 – продемонстрированные ими результаты.

Таблица 1. Основные характеристики систем, проанализированных в работе [42]

Характеристика	ANNIE	Stanford NER	Ritter et al.	Alchemy API	Lupedia
Методы	Газеттиры и конечные автоматы	CRF	CRF	Машинное обучение	Газеттиры и правила
Языки	EN, FR, DE, RU, CN, RO, HI	EN	EN	EN, FR, DE, IT, PT, RU, ES, SV	EN, FR, IT
Предметная область/жанр	новости	новости	Твиттер	Универсально	Универсально
Число типов ИС	**7**	4,3 или 7	3 или 10	324	319
Схема разметки	MUC	CoNLL, ACE	CoNLL, ACE	Alchemy	DBpedia
Тип системы	Java (модуль Gate)	Java	Python	Веб-сервис	Веб-сервис
Лицензия	GPLv3	GPLv2	GPLv3	Некоммерческая	Неизвестно
Возможность адаптации	Да	Да	Частично	Нет	Нет
	Dbpedia Spotlight	TextRazor	Zemanta	YODIE	NERD-ML
Методы	Газеттиры и меры сходства	Машинное обучение	Машинное обучение	Меры сходства	Метод k ближайших соседей и Наивный Байес
Языки	EN	EN, NL, FR, DE, IT, PL, PT, RU, ES, SV	EN	EN	EN
Предметная область/жанр	Универсально	Универсально	Универсально	Твиттер	Твиттер
Число типов ИС	320	1779	81	1779	4
Схема разметки	Dbpedia, Freebase, Schema.org	Dbpedia, Freebase	Freebase	DBpedia	NERD

Тип системы	Веб-сервис	Веб-сервис	Веб-сервис	Java (модуль Gate)	Java, Python, Perl, bash
Лицензия	Apache Licence 2.0	Некоммерческая	Некоммерческая		GPLv3
Возможность адаптации	Да	Нет	Нет	Да	Частично

Таблица 2. Результаты сравниваемых систем

Система	F1 по типу ИС			В целом		
	Location	Org	Person	P	R	F1
ANNIE	24.03	10.08	12.00	22.55	13.44	16.84
DBpedia Spotlight	0.00	0.75	0.77	28.57	0.27	0.53
Lupedia	28.70	19.35	14.21	54.10	12.99	20.95
NERD-ML	43.57	21.45	**49.05**	51.27	**31.02**	38.65
Ritter T-NER	44.81	14.29	41.04	51.03	26.64	35.00
Stanford	**48.58**	27.40	43.07	**64.22**	29.33	**40.27**
Stanford-Twitter	38.93	**30.93**	29.55	38.46	26.57	31.43
TextRazor	9.49	13.37	20.58	38.64	9.10	14.73
Zemanta	35.87	14.56	11.05	63.73	12.80	21.31

Другие авторы, не вошедшие в данный обзор, предлагают смешанный подход к решению задачи выделения именованных сущностей в микроблогах. В частности, Сяохуа Лиу с соавторами [Xiaohua et al. 2011: 29] предлагают гибридный подход, сочетающий метод k ближайших соседей (для предварительной разметки) с моделью условных случайных полей. Такая комбинация методов в сочетании с использованием списков-газеттиров позволяет авторам получить F-меру 80,2% на англоязычных тестах.

2.2 Современные реализации инструментов извлечения именованных сущностей

На данный момент существует множество коммерческих и открытых систем извлечение именованных сущностей. Рассмотрим кратко некоторые из них.

RCO Fact Extractor SDK – это лингвистический анализатор текста, комплексный инструментарий для разработки информационно-поисковых и аналитических систем, использующих анализ текста на русском языке. Библиотека RCO FX Ru (ядро проекта) осуществляет полный синтактико-семантический разбор русского текста. Библиотека выделяет разные классы

сущностей, упомянутых в тексте (персоны, организации, географические названия, предметы, действия, атрибуты и др.), и строит сеть отношений, связывающих эти сущности, а также предоставляет дополнительную грамматическую информацию о составляющих текста. Средствами библиотеки также осуществляется семантическая интерпретация результатов разбора текста - производится описание ситуаций, удовлетворяющих заданным семантическим шаблонам. В состав лингвистического обеспечения пакета, помимо общих словарей и правил русского языка, входят правила выделения специальных объектов (дат, адресов, документов, телефонов, денежных сумм, марок автомобилей и пр.), шаблоны для распознавания различных классов событий и фактов (сделок, экономических показателей, конфликтов, биографических фактов и пр.), характеристик объекта (позитива, негатива и др.), высказываний прямой и косвенной речи.

ABBYY Intelligent Tagger SDK (Compreno). Это инструментарий разработчика, который анализирует неструктурированную текстовую информацию и автоматически извлекает из нее именованные сущности (персоны, организации, даты и другие) и метаданные документов. Полученные данные можно использовать для совершенствования и автоматизации различных бизнес-задач, таких как поиск и анализ знаний, классификация и маршрутизация входящей информации, управление документацией и выявление конфиденциальных данных в ней. Технология Compreno – это универсальная лингвистическая платформа для приложений, решающих множество прикладных задач по обработке текстов на естественном языке. В основе Compreno лежит многоуровневое лингвистическое описание. Помимо ручного описания Compreno использует для анализа большое количество информации, извлекаемой различными статистическими методами из текстовых корпусов. В Compreno реализована процедура семантико-синтаксического анализа текста, в результате которой любому предложению на естественном языке (английском или русском) ставится в соответствие семантико-синтаксическое дерево, моделирующее

смысл предложения и содержащее грамматическую и семантическую информацию о каждом слове предложения.

Томита-парсер (Яндекс) - инструмент для извлечения структурированных данных (фактов) из текста на естественном языке. Извлечение фактов происходит при помощи контекстно-свободных грамматик и словарей ключевых слов. Парсер позволяет написать свою грамматику, добавить свои словари и запустить на любых текстах.

PROMT Analyser. Анализирует любые тексты или документы, выделяет в нем сущности (персоналии, организации, географические названия, геополитические сущности и др.), а также определяет соотносящиеся с этим сущностями действия, дату и место совершения действия, формирует целостный образ документа. Система выполняет тонкий морфологический, синтаксический и семантический анализ, что позволяет максимально точно получать информацию из неструктурированных текстовых данных на разных языках, взаимодействуя даже с такими морфологически богатыми, как русский и немецкий. PROMT Analyser имеет обширную базу данных, но главным его достоинством является то, что он выделяет в текстах также сущности, не представленные в базах. Еще одним преимуществом программы является простая настройка – путем введения значения для новых типов сущностей.

NER от Айтеко. Система автоматического распознавания именованных сущностей служит для типизации имен собственных, терминов, различных названий и т.п. Представленный алгоритм использует статистические языковые модели и правила для «шаблонных» сущностей, таких, как url, e-mail, цифры и пр. Количество типов и их описание задается на этапе обучения системы и не зависит от словаря. Его возможности ограничиваются следующими типами: определение имен людей, названий компаний и организаций, географических объектов, продуктов и брендов, названия праздников, форумов и др. событий. Дополнительно к этому определяются url, e-mail, деньги и даты.

MF LIK R10 МетаФраз Лингвистический интеграционный комплект (Metafraz Lingware Integration Kit, MF LIK) R10 – SDK для разработчиков приложений в виде API к автономному ядру и серверу лингвистического ПО (интеграция технологий фразеологического машинного перевода и семантической обработки неструктурированной текстовой информации МетаФраз в сторонние приложения). Возможности: нормализация текста (для повышения качества поиска средствами СУБД); выделение из текста ключевых выражений, характерных для данной предметной области; классификация выделенных выражений; автоматическое составление аннотации (общего реферата) по документу; автоматическое составление контекстного реферата по документу с учетом пользовательской тематики или поискового запроса; выделение объектов (организации, персоны, должности, бренды и т.д.); определение и типизация связей между объектами; сравнение документов и установление степени их семантической близости для задач кластеризации (группировки документов по смыслу) и антиплагиата.

Eureka Engine. Высокоскоростная система лингвистического анализа текстов модульного типа, позволяющая извлекать новые знания и факты из неструктурированных данных огромных объемов. В систему входят такие модули как: Определение языка сообщения (24 языка, относящихся к разным языковым семьям); Автоматическое определение тональности документа (АОТ) для русского языка; Определение тематики (автоклассификация) для русского языка; Выделение именованной и имен собственных (NER) для русского языка (подключение английского в ближайшее время); Нормализация слов (русский язык); Разметка частей речи (морфоанализ) для русского языка. Возможна обработка не только материалов СМИ, но и сообщений социальных сетей, форумов и блогов. Есть online-демо.

Хурма (Hurma). Хурма – проект, основная цель которого формулировалась как создание простого и удобного в использовании веб-сервиса для массовой обработки текстов и извлечения из них различной информации, полезной как профессиональным прикладным лингвистам и

исследователям, так и различного рода аналитикам коммерческих компаний. Хурма - это не только простой способ быстро обработать большой объём текстов и получить на выходе информацию в структурированном и нормализованном виде, но и возможность строить разнообразную аналитику и проводить статистические исследования на пользовательской коллекции документов.

Zamgi - высокоскоростная система лингвистического анализа текстов модульного типа, позволяющая извлекать новые знания и факты из неструктурированных данных огромных объемов. В систему входят следующие подсистемы: определение языка сообщения; определение тональности документа для русского языка; классификация тематики документа для русского языка; выделение именованных сущностей и имен собственных (NER) для русского и английского языков; нормализация слов для русского языка; определение частей речи и морфоанализ для русского языка.

АРИОН-Лингво. На вход Лингвистический процессор получает текстовый документ. Результатом его работы является массив связной фактографической информации, который далее передается в модуль идентификации для выделения похожих и слияния совпадающих объектов. Выделение фактографической информации осуществляется с помощью специализированных правил, которые описывают процедуры выделения объектов и связей на внутрисистемном языке Лингвистического процессора, построенном на базе XML.

Textocat – облачный веб-сервис, предоставляющий RESTful API для решения базовых задач аналитики русскоязычных текстов. В текущей версии поддерживаются следующие функции: распознавание упоминаний сущностей, таких как люди, организации, геополитические сущности, сооружения и локации; выделение временных и денежных выражений; полнотекстовый поиск с учетом выделенных аннотаций.

DictaScope Tokenizer от компании Dictum занимается выявлением в текстах на русском языке текстовых объектов и фактов, таких как: персона,

должность, спортивные команды, организации (коммерческие и некоммерческие), географические объекты, даты, количественные показатели, высказывания персон, должность, место работы и др. Выявленные объекты и факты приводятся к канонической форме (нормализуются). В состав модуля включаются образцы правил для выявления и нормализации некоторых из перечисленных категорий текстовых объектов и фактов. Входной формат – plain-текст. Результат может быть выдан в формате XML. Для работы программы требуется морфологический словарь. Программа поставляется в виде динамической библиотеки для Windows/FreeBSD.

XANALYS. Этот инструмент извлечения сущностей из различных текстов (ранее известен как Quenza и PowerIndexer), извлекает из текста различные объекты:

- Сущности, такие как: Лица, Организации, События;

- Атрибуты сущностей, такие как: Пол лица, Профессия лица, Название компании;

- Отношения, такие как: находится, работает в, участвовал в событии.

Indexer имеет интерфейсы , достаточные для его интеграции во внешнюю систему.

iLab - лаборатория по извлечению информации. Извлечение структурированной информации из неструктурированных и слабоструктурированных текстов. В настоящий момент сделано извлечение адресов с их нормализацией. Извлечение организаций и персон на стадии разработки.

Businessobjects Text Analysis. Программа позволяет извлекать информацию по 35 типам объектов и событий, включая людей, географические места, компании, даты, денежные суммы, email-адреса, и выявлять взаимосвязи между ними. Обладает мощными лингвистическими возможностями по чтению и пониманию документов на 30 языках. На основе структуры естественных языков программа может распознавать информацию, связанную с заданными пользователем объектами, такими как

названия проектов, анализировать взаимосвязи между событиями и конкретные фразы на предмет сентимент-анализа (sentiment analysis).

AeroText. Версия AeroText 5.x существует в виде набора компонентов. Программа позволяет осуществлять извлечение информации, связанной с конкретными объектами (персоны, организации, географические объекты и т.п.), ключевые фразы (указание на конкретное время, объемы денег) и т.п. Решение также анализирует взаимосвязи между сущностями, позволяя решить проблему множественных референтов одной и той же сущности, осуществляет идентификацию взаимоотношений между сущностяи, извлечение событий (кто, где, когда), категоризацию тем (предмет, его определение), определение временного промежутка, когда имело место событие, определение места, которое может быть привязано к карте.

FreeLing. Пакет FreeLing предоставляет функционал для анализа текста с учетом специфики языка. В него входят следующие компоненты:

1 Разметка текста (токенизация);

2. Выделение предложений;

3. Морфологический анализ;

4. Определение составных слов;

5. Вероятностное определение части речи неизвестного слова (hmm tagger);

6. Обнаружение и определение именной группы;

7. Классификация именной группы;

8. Построение дерева зависимостей (слов в предложении);

9. Определение местоимений (местоименных словоформ);

10. Нормализация и определение дат, чисел, процентных соотношений, валюты и физических величин (скорость, вес, температура, плотность и т.д.);

11. Определение части речи (вероятностное);

В настоящее время проект поддерживает языки: испанский, каталонский, галисийский, итальянский, английский, валлийский, португальский, австрийский, русский.

3. Материал исследования - корпусы текстов

3.1 Корпус текстов микроблогов

Корпус текстов социальной сети Твиттер собран с помощью API Twitter в формате .json. Корпус насчитывает 8 600 записей на русском языке за период с начала 2014 года по январь 2017 года объемом 136 070 словоупотреблений. Для отбора записей и отсеивания записей, не содержащих именованных сущностей, критерии поиска включали распространенные имена, фамилии известных людей, а также наименования организаций из перечня, сформированного на основе выборки из новостных текстов, проанализированной и размеченной вручную.

Для разметки границ именованных сущностей широко распространена схема IOB: метка B означает начало сущности; I – расположение внутри неё; меткой O отмечаются токены, не входящие в именованную сущность.

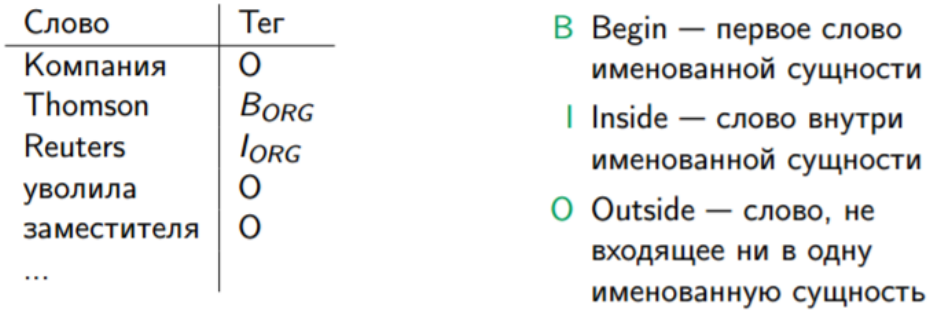

Рисунок 1. Схема аннотации IOB

Разметка корпуса проведена вручную автором и вторым аннотатором (взрослым носителем русского языка, имеющим филологическое образование).

Для оценки практических результатов работы из корпуса были удалены записи, при разметке которых наблюдались разногласия между аннотаторами (1141 запись из 8600).

Пример аннотации:

Газеты [Org B] "Вечерний [Org I] Минск" [Org I], "Минский [Org B] Курьер" [Org I] прислали [O] КП [O]. Предлагают [O] разместить [O] у [O] них [O] рекламу [O] инет-магазина [O])) Думаю [O], конверсия [O] зашкалит[O]

3.2. Корпус новостных текстов

В качестве фонового корпуса был использован корпус новостных текстов, подготовленный проектом OpenCorpora к соревнованию FactRuEval в рамках конференции Диалог 2016.

Предложенный в рамках конференции «Диалог» корпус состоит из 122 новостных текстов. Каждому тексту соответствует 4 файла:

1. Файл с токенами – деление текста на токены и предложения. Каждая строка содержит идентификационный номер - id токена, позицию его начала, длину и текст.

2. Файл со спанами – первый уровень разметки. Кроме всего прочего включает в себя id спана и id входящих токенов.

3. Файл с объектами – упоминание объектов. Включает id объекта и id входящих в него спанов.

4. Файл кореференций и идентификаций - отношения между несколькими идентифицированными объектами.

Рассмотрим подробнее первые 3 из них, которые были использованы в работе. Примеры файлов приведены ниже. На рисунке 2 показан файл токенов, рисунок 3 иллюстрирует пример разметки файла со спанами, рисунок 4 представляет файл с объектами данной демонстрационной коллекции.

```
89968 0 7 Встреча
89969 8 1 с
89970 10 6 послом
89971 17 6 Италии
89972 24 1 в
89973 26 4 миде
89974 31 6 Грузии

89975 39 2 По
89976 42 10 инициативе
89977 53 11 итальянской
89978 65 7 стороны
89979 73 12 чрезвычайный
89980 86 1 и
89981 88 11 полномочный
89982 100 5 посол
89983 106 6 Италии
89984 113 1 в
89985 115 6 Грузии
89986 122 7 Виторио
89987 130 7 Сандали
89988 138 10 встретился
89989 149 1 с
89990 151 12 заместителем
89991 164 8 министра
89992 173 11 иностранных
89993 185 3 дел
89994 189 6 Грузии
89995 196 11 Александром
89996 208 11 Налбандовым
89997 219 1 .
```

Рисунок 2. Фрагмент файла токенов.

```
32962 loc_name 17 6 89971 1  # 89971 Италии
32963 org_name 26 4 89973 1  # 89973 миде
32965 loc_name 31 6 89974 1  # 89974 Грузии
32966 job 10 6 89970 1  # 89970 послом
64002 job 10 13 89970 2  # 89970 89971 послом Италии
32951 loc_name 106 6 89983 1  # 89983 Италии
32952 loc_name 115 6 89985 1  # 89985 Грузии
32953 name 122 7 89986 1  # 89986 Виторио
32954 surname 130 7 89987 1  # 89987 Сандали
32955 loc_name 189 6 89994 1  # 89994 Грузии
32956 name 196 11 89995 1  # 89995 Александром
32957 surname 208 11 89996 1  # 89996 Налбандовым
32958 job 73 32 89979 4  # 89979 89980 89981 89982 чрезвычайный и полномочный посол
32959 job 151 37 89990 4  # 89990 89991 89992 89993 заместителем министра иностранных дел
32960 job 164 24 89991 3  # 89991 89992 89993 министра иностранных дел
32961 job 100 5 89982 1  # 89982 посол
64007 job 73 48 89979 7  # 89979 89980 89981 89982 89983 89984 89985 чрезвычайный и полномочный посол Италии в Грузии
64013 job 100 21 89982 4  # 89982 89983 89984 89985 посол Италии в Грузии
```

Рисунок 3. Пример разметки файла со спанами

```
16972 LocOrg 32962 # Италии
16975 Org 32963 32965 # миде Грузии
16974 LocOrg 32965 # Грузии
16967 LocOrg 32951 # Италии
16968 LocOrg 32952 # Грузии
16969 Person 32953 32954 # Виторио Сандали
16970 LocOrg 32955 # Грузии
16971 Person 32956 32957 # Александром Налбандовым
```

Рисунок 4. Фрагмент файла с объектами

Для составления выборки именованных сущностей совершается последовательный обход представленных файлов:

1) Из файла с токенами было получено разбиение текста на предложения (пустая строка в файле) и список всех токенов с их идентификационными номерами.

2) Из файла объектов извлекаются типы именованных сущностей и id входящих в их состав спанов.

3) В файле спанов находились спаны по идентификационным номерам, полученным на предыдущем шаге. Затем для каждой именованной сущности получался набор id токенов, входящих в её состав.

4) Происходит разметка полученного на первом шаге списка токенов, разбитого на предложения, по схеме IOB.

4. Практическое применение инструментов выделения именованных сущностей

Анализ методов, применяемых для выделения именованных сущностей показал, что с данной задачей хорошо справляются как методы, основанные на правилах и словарях, так и различные методы машинного обучения. В то же время было показано, что особенности предметной области затрудняют

применение обеих групп методов и снижают результативность традиционных систем, настроенных на обработку научных и публицистических текстов.

Состояние разработанности проблемы, обилие готовых систем с открытым кодом, адаптированных для тех или иных типов текстов и сущностей заставило искать решение задачи среди существующих инструментов, любой из которых, несомненно, требовал доработки и адаптации с учетом конкретного материала.

Исходя из вышесказанного, для дальнейшей доработки и тестирования были выбраны 2 инструмента – Gate и Томита-парсер. Обе системы работают с правилами-грамматиками и словарями. Особенностью Gate, послужившей основой для её выбора является то, что эта система хорошо зарекомендовала себя при обработке текстов микроблогов на русском языке. Томита-парсер же был выбран в силу относительной простоты работы с ним и адаптированности для текстов на русском языке.

4.1. Система GATE

GATE (General Architecture for Text Engineering) – модульная система обработки текста для извлечения информации, основанная на правилах, разработанная университетом Шеффилда.

Для проведения эксперимента была использована модифицированная и дополненная версия системы Gate, предложенная Калиной Бончевой и Леоном Держински в 2013 году – TwitIE [Bontcheva et al. 2013: 33].

На рисунке 5 представлена схема работы системы Gate с плагином Twitie.

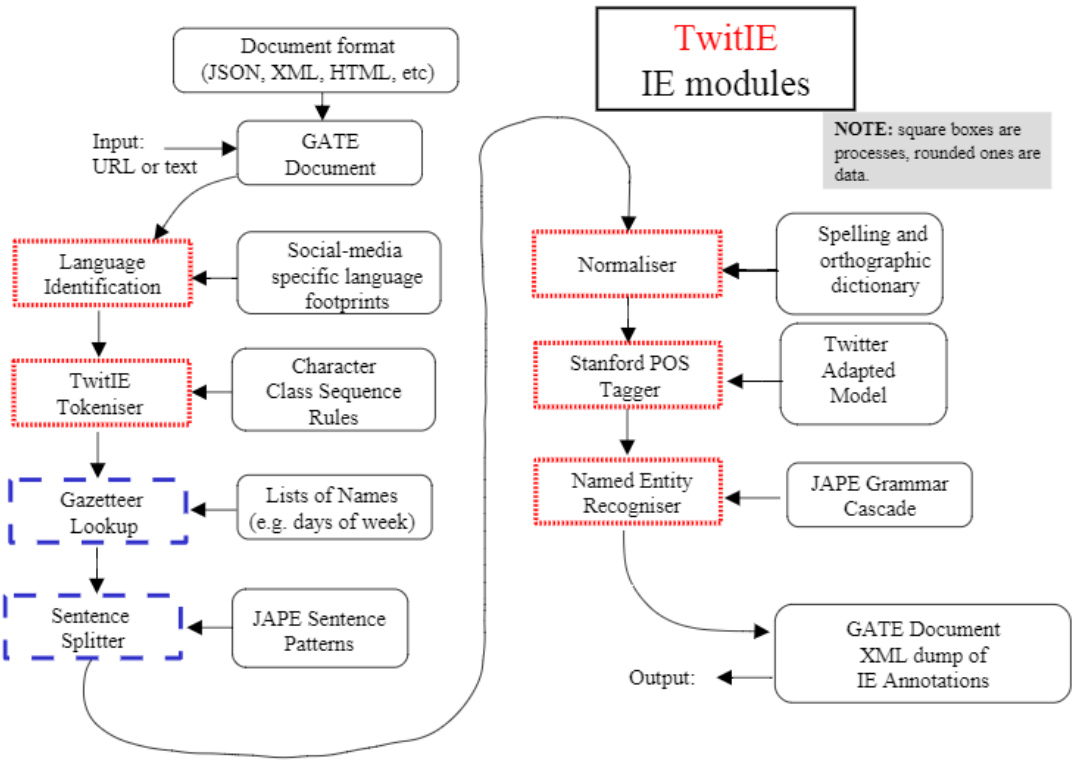

Рисунок 5. Схема работы Twitie

Этапы работы:

При работе с системой Gatекорпус текстов последовательно проходит несколько модулей.

1. Модуль определения языка работает на основе инструмента TextCat (версия, адаптированная для твиттера – [Carter et al., 2013: 63]), который в данный момент поддерживает 5 языков, в их числе нет русского. Для обеспечения работы данного модуля он был обучен на половине корпуса.

2. Токенизатор: вместо токенизатора по умолчанию (ANNIE English Tokenizer) использован GATE Unicode Tokeniser. При этом аббревиатуры и URL считаются одним токеном. Хештег и следующее за ним упоминание пользователя делятся на 2 токена. Сохраняется паттерн капитализации.

3. Газеттиры. Списки имен, названий стран, континентов, городов, организаций на русском языке были предоставлены плагином Russian plugin и дополнены вручную. Списки содержат все падежные формы каждого входящего в них слова. В списки имен помимо полных имен добавлены

распространенные сокращенные варианты (например, Александр – Саня, Саша, Сашка, Шурик). Список названий организаций насчитывает 21040 элементов, список имен – 1566, список геолокаций (страны, города, континенты) – 2065 элементов. Помимо данных списков, составлены газеттиры слов-указателей на именованную сущность (формы обращения в людям, некоторые профессии и должности, организационно-правовые формы предприятий, и т.д.) В эти списки для Персон вошли 343 элементов, для Организаций - 47, для Локаций – 99.

```
813  Петр
814  Петра
815  Петре
816  Петром
817  Петру
818  Питер
819  Питера
820  Питере
821  Питером
822  Питеру
823  Питт
824  Питта
825  Питте
826  Питтом
827  Питту
828  Руслан
829  Руслана
830  Руслане
831  Русланом
832  Руслану
833  Станислав
834  Станислава
835  Станиславе
836  Станиславом
837  Станиславу
```

Рисунок 6. Фрагмент газеттира мужских имен.

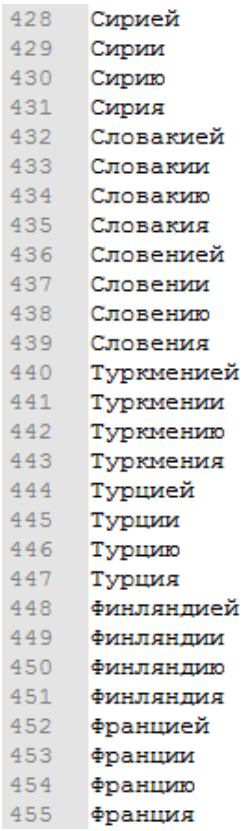

Рисунок 7. Фрагмент газеттира названий городов.

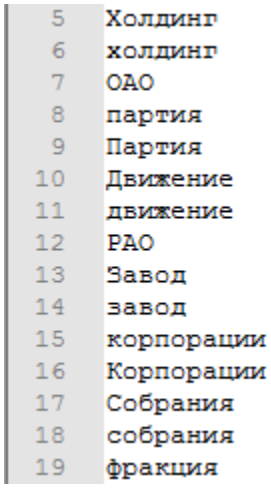

Рисунок 8. Фрагмент газеттира слов-указателей именованной сущности типа «ORG» (Организация)

4. Модуль выделения предложений (Sentence Splitter) системы Gate применяется без изменений.

5. Модуль нормализации включает спеллчекер на основе расстояния Левенштейна и словари замен на русском языке, составленные вручную (на основе анализа собранного корпуса), включающий нестандартные написания, характерные для соцсетей.

Подготовлены словари опечаток (587 замен), сокращений (158) и специфического сленга (198).

Примеры из словаря опечаток:

дигистировать	дегустировать
дегистировать	дегустировать
дигустировать	дегустировать
рождетство	рождество
рождетсво	рождество

Примеры из словаря сленга:

пачиму	почему
патаму	потому
шта	что

Примеры из словаря сокращений:

мб	может быть
хз	хрен знает
спб	Санкт-Петербург
смр	Самара
екб	Екатеринбург

6. Вместо Stanford POS tagger подключен модуль частеречной разметки из Russian Plugin.

7. Модуль выделения именованных сущностей (Named Entity Recogniser) является встроенным модулем системы. На основании грамматик, описанны

4.2. Томита-парсер

Томита-парсер – созданный компанией Яндекс вариант GLR-парсера (от англ. Generalized Left-to-right Rightmost derivation parser — Обобщенный восходящий магазинный анализатор), впервые описанного Масару Томита в 1984 году. В настоящее время открытый код парсера доступен для разработчиков в коммерческих и некоммерческих целях. В составе парсера три основных лингвистических процессора: токенизатор (осуществляет разбиение входного текста на слова и несловарные токены), сегментатор (разделяет текст на предложения) и морфологический анализатор mystem (производит частеречную разметку).

Основными компонентами парсера являются: газеттир, набор контекстно-свободных (КС) грамматик (пользовательских шаблонов) и набор описаний типов фактов, которые могут фиксироваться (порождаться) этими грамматиками в результате процедуры интерпретации.

Газеттир — словарь ключевых слов, которые используются в процессе анализа КС-грамматиками. Каждая статья этого словаря задает множество слов и словосочетаний, объединенных общим свойством (например, «мужские имена»).

Грамматика представляет собой множество правил на языке КС-грамматик, описывающих синтаксическую структуру выделяемых цепочек.

Грамматики для Томита-парсера состоят из правил. У каждого правила есть левая и правая части, разделенных символом —>. В левой части стоит один нетерминал (S в примере, приведенном ниже). В правой части стоит список терминалов или нетерминалов (S1 ... Sn), после которого указываются условия (Q), применяемые ко всему правилу в целом.

Грамматический парсер запускается всегда на одном предложении. Перед запуском терминалы грамматики отображаются на слова (или словосочетания) предложения. Одному слову может соответствовать много

терминальных символов. Таким образом, парсер получает на вход последовательность множеств терминальных символов. На выходе - цепочки слов, распознанные этой грамматикой.

Факты — таблицы с колонками, которые называются полями фактов. Факты заполняются во время анализа парсером предложения. Как и чем заполнять поля фактов указывается в каждой конкретной грамматике (интерпретация). Типы фактов описываются в отдельном файле.

Для запуска Томита-парсера созданы файлы: config.proto — конфигурационный файл парсера (сообщает парсеру, где искать все остальные файлы и как их интерпретировать); dic.gzt — корневой словарь, содержит перечень всех используемых в проекте словарей и грамматик; mygram.cxx — грамматика; kwtypes.proto — описания типов ключевых слов.

Фрагмент файла dic.gzt:

```
encoding "utf8";
import "base.proto";
import "articles_base.proto";
import "kwtypes_my.proto";
import "facttypes.proto";
TAuxDicArticle "LOC"
{
    key = { "tomita:loc.cxx" type=CUSTOM }
}
city "Нижний_Новгород"
{
    key = "Нижний Новгород";
    mainword = 2;
}
city "Санкт_Петербург"
{
    key = "Санкт-Петербург" | "Питер" | "Петербург";
    lemma = "Санкт-Петербург";
```

Фрагмент файла config.proto:

```
encoding "utf8";
TTextMinerConfig {
  Dictionary = "dic.gzt";       // корневой словарь газеттира
  PrettyOutput = "debug.html";  // файл с отладочным выводом
  Input = {
    File = "test.txt";          // файл с анализируемым текстом
    Type = dpl;                 // режим чтения "document per line" (каждая строка
- отдельный документ)
  }
  Articles = [
    { Name = "LOC" }       // Запустить статью корневого словаря "Location"
  ]
  Facts = [
    { Name = "LocFact" }       // Сохранить факт "LocFact"
  ]
  Output = {
    File = "facts.txt";         // Записать факты в файл "facts.txt"
    Format = text;              // используя при этом простой текстовый формат
  }
}
```

Алгоритм работы парсера:

Парсер ищет вхождения всех ключей из газеттира. Если ключ состоит из нескольких слов (например, «Нижний Новгород»), то создается новое искусственное слово, которое разработчики назвали «мультиворд». Из всех найденных ключей газеттира отбираются те, которые упоминаются в грамматике.

Среди отобранных ключей могут встречаться и мультиворды, пересекающиеся друг с другом или включающие в себя одиночные ключевые слова. Парсер должен покрыть предложение непересекающимися ключевыми словами так, чтобы как можно большие куски предложения были охвачены ими.

Линейная цепочка слов и мультивордов подается на вход GLR-парсеру. Терминалы грамматики отображаются на входные слова и мультиворды.

На последовательности множеств терминалов GLR-парсер строит все возможные варианты разметки. Из всех построенных вариантов также отбираются те, которые как можно шире покрывают предложение.

Затем парсер запускает процедуру интерпретации на построенном синтаксическом дереве. Он отбирает специально помеченные подузлы, а слова, которые им соответствуют, записываются в порождаемые грамматикой поля фактов.

При создании газеттиров и грамматик использовались те же списки имен, названий стран, континентов, городов, организаций, что и при работе с системой Gate.

5. Методика оценки результатов

Оценка систем выделения сущностей является стандартным индикатором прогресса данной области, и может служить проверкой работоспособности новых методов. По общему правилу оценка систем проводится на корпусах, размеченных вручную (создается так называемый «эталон» разметки - “gold standard”). Методики измерения основных показателей, однако, отличаются от работы к работе.

В ходе серии конференций CoNLL был предложен следующий интуитивно понятный способ оценки: именованная сущность считается выделенной системой правильно, если и ее тип, и границы, отмеченные системой, совпадают с типом и границами, размеченными аннотаторами в корпусе; в противном случае можно считать, что сущность выделена неправильно. Назовем такой способ оценки оценкой методом точного соответствия. Точность (P), полнота (R) и F-мера в данном случае определяются следующим образом:

P = количествово верно выделенных сущностей/кол-во всех выделенных сущностей,

R = количествово верно выделенных сущностей/ кол-во сущностей в корпусе,

$F = 2\ PR / (P + R)$.

Данный метод оценки широко распространен, однако подвергается критике. Оценка точным соответствием не позволяет снисходительно относится к ошибкам в границе сущности или в ее классе, которые вполне могут быть совершены и людьми при разметке текста. Кристофер Маннинг предложил способ подсчета сегментов, который бы учитывал 3 дополнительных типа ошибки: сущность выделена, но есть неточность в границе, есть ошибка в классе сущности, но граница верна, ошибка есть как в классе, так и в границе сущности. Однако, предложенный способ не нашел широкого распространения.

Наравне с вышеназванным существуют и другие способы оценки, применявшиеся в разное время и для подсчета результатов на различном материале.

Основные недостатки стандартных способов расчета точности и полноты:

- Если считать правильно выделенными только фрагменты, которые точно совпадают с границами фрагментов-эталонов, скорее всего, результаты будут слишком низкими и не будут отражать потенциал системы. Кроме того, эксперты-аннотаторы также расходятся в оценке границ многословных сущностей.

- В то же время, если рассчитывать точность и полноту на основании «наложения» (“overlap”) [Choi et al., 2006: 64; Breck et al., 2007: 65], то предпочтение неминуемо будет отдаваться более длинным фрагментам - вплоть до фрагментов, содержащих целые предложения, если эталон содержит любой фрагмент этого предложения.

Предлагаемая система позволяет избежать этих крайностей. Крайние значения метрик в данном случае будут ограничены снизу оценкой точного совпадения, а сверху – оценкой «наложения».

Для оценки результатов тестирования хочется использовать схему, основанную на пересечении (в отличии от «наложения»), предложенную Йохансоном и Москитти [Johansson, Moschitti, 2013 : 48] при решении задачи оценки тональности.

Как в случае оригинальной статьи, так и в нашей задаче выделения именованных сущностей, часто границы выражений, представляющих сущности, не являются четко определенными.

Идея состоит в том, чтобы приписать значения от 0 до 1 каждому сегменту в отличие от традиционного подхода, при котором каждый сегмент может считаться либо верно, либо неверно выделенным. Покрытие (c) фрагмента (s) (множество токенов) определяется по отношению к другому фрагменту s′, что указывает, насколько хорошо фрагмент s′ «покрыт» фрагментом s:

$$c(s, s') = \frac{|s \cap s'|}{|s'|}$$

Где |s| - длина фрагмента s, а перечесение s∩s’ представляет множество токенов, которые являются общими для обоих фрагментов. Так как и в оригинальном исследовании, и в нашем случае существует не один, а несколько тегов для фрагментов, то c(s,s′) считается равным нулю, если теги s и s′ различны. Используя покрытие фрагмента, мы определяем покрытие набора фрагментов, s_1, s_2, … s_n по отношению к s’

$$C(\boldsymbol{S}, \boldsymbol{S'}) = \sum_{s_j \in \boldsymbol{S}} \sum_{s'_k \in \boldsymbol{S'}} c(s_j, s'_k)$$

Таким образом, точность и полнота, определяются как пересечение, выделенных фрагментов $\hat{S}$ по отношению к фрагментам-эталонам S:

$$P(S,\hat{S}) = \frac{C(S,\hat{S})}{|\hat{S}|} \qquad R(S,\hat{S}) = \frac{C(\hat{S},S)}{|S|}$$

Где | Ŝ |- число фрагментов в множестве Ŝ.

Например, в тексте «Сергею лазареву в новом клипе сердце и лицо разбила красотка-боксерша» был выделен сегмент «Сергею» с пометой «PER» (Персона), в то время как в аннотированном корпусе помета «PER» присвоена словосочетанию «Сергею лазареву». В этом случае мы предварительно расчитываем коэффициент покрытия, равный в этом случае 0,5 и, с одной стороны, учитываем данную сущность как правильно выделенную при подсчете результатов, а с другой стороны, можем видеть и учесть при подсчёте, что она не является идеально выделенной.

6. Количественные результаты исследования

Результаты эксперимента приведены в таблицах 3 и 4.

Таблица 3. Результаты работы двух систем

		P	R	F1
Микроблоги	Томита	0.63	0.58	0.61
	Gate	0.52	0.49	0.51
Новости	Томита	0.79	0.81	0.80
	Gate	0.76	0.82	0.79

Таблица 4. Результаты на корпусе микроблогов с разбиением по типам именованных сущностей

		P	R	F1
Томита	Org	0.59	0.53	0.56
	Per	0.71	0.68	0.69
	Loc	0.59	0.54	0.56
Gate	Org	0.42	0.37	0.39
	Per	0.61	0.59	0.60
	Loc	0.54	0.51	0.52

7. Выводы, направления дальнейшей работы

Оба инструмента показали неплохие результаты (хотя и значительно ниже state-of-the-art показателей) на корпусе микроблогов.

Несмотря на учет особенностей предметной области и разработку специфических словарей для обработки текстов микроблогов, новостные тексты всё же представляют меньшие трудности для обработки.

Так как оба инструмента являются системами, опирающимися на правила, точность их работы может быть весьма высока, т.е. составленные газеттиры и словари замен были недостаточно полными.

Невысокие показатели системы Gate объясняются отчасти, скорее всего, тем, что специфический модель частеречной разметки с учетом особенностей текстов Твиттера не был использован, а примененный вместо него модуль для русского языка не был построен с учетом данной специфики. Очевидно также, что при таком подходе размер словарей замен (коррекции орфографии и расшифровки аббревиатур) должен быть значительно увеличен.

Также нужно отметить, что выбранные три класса именованных сущностей – Персона, Локация и Организация - не лучшим образом отражают специфику текстов. В частности, сущности типа Организация

вызвали наибольшие проблемы у обеих систем в связи с тем, что, во-первых, были хуже представлены в корпусе, и во-вторых, газеттиры для них оказались недостаточно адаптированными.

Включение же в область исследования других типов сущностей, в частности Продукт (Товар), могло бы положительно повлиять на общий результат.

Анализ результатов показывает, что с точки зрения разбиения на классы ИС результаты, полученные при помощи Томита-парсера были более однородны, система Gate же показала большую вариативность, что говорит о недостаточной адаптации всех модулей системы.

Учёт этих недостатков в дальнейшей работе может способствовать улучшению результатов.

Кроме того, представляется интересным применить созданный корпус, газеттиры и признаки, использованные при написании правил, для тестирования методами машинного обучения, в частности с использованием метода условных случайных полей.

Библиография

1. Антонова А.Ю., Соловьев А.Н. (2013) Использование метода условных случайных полей для обработки текстов на русском языке - Компьютерная лингвистика и интеллектуальные технологии.

2. Глазова М.А. (2010) Использование Марковской модели максимальной энтропии для задачи извлечения собственных имен из текста - Труды 12-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции».

3. Нехай И.В. (2012) Применение n-грамм и других статистик уровня символов и слов для семантической классификации незнакомых собственных имен – сборник докладов «Диалог», том 1.

4. Подобряев А.В. Поиск упоминаний лиц в новостных текстах с использованием модели условных случайных полей - Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции».

5. Brykina M. M., Faynveyts A. V., Toldova S. Yu. (2013) Dictionary-based Ambiguity Resolution in Russian Named Entities Recognition – International Workshop on Computational Linguistics and its Applications, ed. A. Narin'yani, v. 1

6. Collins Michael and Singer, Y. (1999) Unsupervised Models for Named Entity Classification - Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

7. Maynard, D., V. Tablan, K. Bontcheva, H. Cunningham, and Y. Wilks (2003) Muse: a Multi-Source Entity Recognition System - Submitted to Computers and the Humanities.

8. Nadeau D., Sekine S. (2006) A Survey of Named Entity Recognition and Classification - Lingvisticae Investigationes, 3 - 26.

9. Nigam K., Lafferty J., McCallum A. (1999) Using maximum entropy for text classification - In IJCAI Workshop on Machine Learning for Information Filtering

10. Popov B., Kirilov A., Maynard, D. and Manov, D. (2004) Creation of reusable components and language resources for Named Entity Recognition in Russian - Proc. Conference on Language Resources and Evaluation.

11. Rationov L., Roth D. (2009) Design challenges and misconceptions in named entity recognition - Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pages 147–155

12. Rau, Lisa F. (1991) Extracting Company Names from Text - Proc. Conference on Artificial Intelligence Applications of IEEE.

13. Tweet Segmentation and Its Application to Named Entity Recognition. Chenliang Li, Aixin Sun, Jianshu Weng, Qi He. IEEE Trans. Knowledge and Data Engineering, 2015

14. Augmenting Business Entities with Salient Terms from Twitter. Riham Mansour, Nesma Refaei and Vanessa Murdock. In Proc. COLING 2014.

15. Adapting taggers to Twitter with not-so-distant supervision. Barbara Plank, Dirk Hovy, Ryan McDonald and Anders Søgaard. In Proc. COLING 2014

16. Chenliang Li, Aixin Sun. Fine-Grained Location Extraction from Tweets with Temporal Awareness. In Proc. SIGIR 2014

17. Saeid Hosseini, Sayan Unankard, Xiaofang Zhou, Shazia Sadiq. Location Oriented Phrase Detection in Microblogs. In Proc. DASFAA 2014.

18. Chenliang Li, Aixin Sun, Jianshu Weng, Qi He.Exploiting Hybrid Contexts for Tweet Segmentation. In Proc. SIGIR 2013

19. FS-NER: A Lightweight Filter-Stream Approach to Named Entity Recognition on Twitter Data Diego Marinho de Oliveira, Alberto H. F. Laender, Adriano Veloso, Altigran S. da Silva. In Proc. WWW (Companion) 2013.

20. Nerit:Named Entity Recognition for Informal Text. David Etter and Francis Ferraro and Ryan Cotterell and Buzek, Olivia and Van Durme, Benjamin. Tech Report. Johns Hopkins University. 2013

21. Xiaohua Liu, Ming Zhou. Two-Stage NER for Tweets with Clustering. Inf. Process. Manage. 2013

22. Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, Bu-Sung Lee. TwiNER: Named Entity Recognition in Targeted Twitter Stream. In Proc. SIGIR 2012

23. Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, Xiangyang Zhou. Joint Inference of Named Entity Recognition and Normalization for Tweets. In Proc. ACL 2012

24. Bo Han, Timothy Baldwin. Lexical Normalization of Short Text Messages: Makn Sens a #twitter. In Proc. ACL 2011

25. K. Gimpel, N. Schneider, B. O'Connor, D.Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, N. A. Smith.Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In Proc. ACL 2011

29. Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou.Recognizing Named Entities in Tweets. In Proc. ACL-HLT 2011

30. Alan Ritter, Sam Clark, Mausam, Oren Etzioni. Named Entity Recognition in Tweets: An Experimental Study. In Proc. EMNLP 2011

31. Jason J. Jung. Towards Named Entity Recognition Method for Microtexts in Online Social Networks: A Case Study on Twitter. In Proc. ASONAM 2011

32. Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating Named Entities in Twitter Data with Crowdsourcing. In Proc. NAACL-HLT Workshop 2010

33. K. Bontcheva, L. Derczynski, A. Funk, M.A. Greenwood, D. Maynard and N. Aswani. 2013. "TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text". In Proceedings of the International Conference on Recent Advances in Natural Language Processing, ACL.

34. L. Derczynski, A. Ritter, S. Clarke, and K. Bontcheva. 2013. "Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data". In Proceedings of the International Conference on Recent Advances in Natural Language Processing, ACL. 3

35. Leon Derczynski, Diana Maynard, Niraj Aswani and Kalina Bontcheva. Microblog-Genre Noise and Impact on Semantic Annotation Accuracy.

Proceedings of the 24th ACM Conference on Hypertext and Social Media. Pages 21-30.

36. Truc-Vien T. NGUYEN and Alessandro MOSCHITTI. 2012. Structural Reranking Models for Named Entity Recognition. Intelligenza Artificiale, vol. 6, no. 2, pp. 177-190, 2012.

37. Рубцова Ю.В. Метод построения и анализа корпуса коротких текстов для задачи классификации отзывов. Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XV Всероссийской научной конференции RCDL'2013, Ярославль, Россия, 14-17 октября 2013 г. – Ярославль: ЯрГУ, 2013. –С. 269-275.

38. Guillaume Lample et al. Neural Architectures for Named Entity Recognition. Proceedings of NAACL 2016.

39. Daniele Bonadiman et al. Deep Neural Networks for Named Entity Recognition in Italian. Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015.

40. James Hammerton. Named Entity Recognition with Long Short-Term Memory. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4. Pages 172-175.

41. Ronan Collobert et al. Natural Language Processing (Almost) from Scratch. The Journal of Machine Learning Research archive. Volume 12, 2/1/2011. Pages 2493-2537

42. Leon Derczynski et al. Analysis of Named Entity Recognition and Linking for Tweets. Information Processing & Management 51(2):32-49. October 2014.

43. Rinat Gareev et al. 2013. Introducing Baselines for Russian Named Entity Recognition. Conference: Proceedings of the 14th international conference on Computational Linguistics and Intelligent Text Processing - Volume Part I.

45. Darwish, Kareem and Wei Gao. “Simple Effective Microblog Named Entity Recognition: Arabic as an Example.” LREC (2014).

46. Pikakshi Manchanda. Entity Linking and Knowledge Discovery in Microblogs. ISWC-DC 2015 The ISWC 2015 Doctoral Consortium, 25

47. Malmasi S., Dras M. (2016) Location Mention Detection in Tweets and Microblogs. In: Hasida K., Purwarianti A. (eds) Computational Linguistics. Communications in Computer and Information Science, vol 593. Springer, Singapore.

48. Richard Johansson, Alessandro Moschitti. Relational Features in Fine-Grained Opinion Analysis. Computational Linguistics. September 2013, Vol. 39, No. 3, Pages: 473-509

49. Sysoev A. A., Andrianov I. A. Named Entity Recognition in Russian: the Power of Wiki-Based Approach . Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2016”

59. Popov A. M., Adaskina Yu. V., Andreyeva D. A., Charabet Ja., Moskvina A. D., Protopopova E. V., Yushina T. A. Named Entity Normalization for Fact Extraction Task. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2016”

51. Stepanova M. E., Budnikov E. A., Chelombeeva A. N., Matavina P. V., Skorinkin D. A. Information Extraction Based on Deep Syntactic-Semantic

Analysis. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2016”

52. Petra Saskia Bayerl, Karsten Ingmar Paul. What Determines Inter-Coder Agreement in Manual Annotations? A Meta-Analytic Investigation. Computational Linguistics. December 2011, Vol. 37, No. 4, Pages: 699-725

53. Bikel D. M., Miller S., Schwartz R., Weischedel R. Nymble: A highperformance learning name-finder. In Proc. of ANLP-97, 1997. P. 194–201.

54. Kaiser K., Miksch S. Information Extraction. A survey. Technical Report: Vienna University of Technology, 2005.

55. McCallum A., W. Li Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proc. of CoNLL-03, 2003.

56. Ponzetto S. P., Strube M. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In Proc. of HLT-NAACL-06, 2006. P. 192–199.

57. Tjong Kim Sang E. F. Introduction to the CoNLL-2002 shared task: Language-independent Named Entity Recognition. In Proc. of CoNLL-02, 2002.

58. Tjong Kim Sang E. F., De Meulder F. Introduction to the CoNLL-2003 shared task: Language independent Named Entity Recognition. In Proc. of CoNLL03, 2003. P. 142–147.

59. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named Entity Recognition through Classifier Combination. In Proceedings ofCoNLL-2003.

60. Jason P. C. Chiu and Eric Nichols. Named Entity Recognition with Bidirectional LSTM-CNNs.CoRR, abs/1511.08308, 2015.

61. Yang, Z., Salakhutdinov, R., and Cohen, W. (2016). Multi-task cross-lingual sequence tagging from scratch.CoRR, abs/1603.06270.12

62. Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T.-Y. (2014). Rc-net: A general framework for incorporating knowledge into word representations. In Proceedingsof the 23rd ACM International Conference on Conference on Information and KnowledgeManagement, pages 1219–1228. ACM.

63. Simon Carter, Wouter Weerkamp, and Manos Tsagkias.2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. Language
Resources and Evaluation, pages 1–21.

64. Y. Choi, E. Breck, C. Cardie. Joint extraction of entities and relations for opinion recognition. Proceedings of the 2006 Conference on Empirical Methods in Natural Language.

65. E Breck, Y Choi, C Cardie. Identifying Expressions of Opinion in Context.
IJCAI 7, 2683-2688, 2007.