Кривальцевич Е.В.

ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА (NATURAL LANGUAGE PROCESSING) ПРИ ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИИ NLTK (NATURAL LANGUAGE TOOLKIT) НА БАЗЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Язык – средство передачи информации, информация заключена в тексте (не в языке), текст «построен» с использованием языка, языковой системы. Характеристики языка определяются задачей эффективно обеспечивать порождение и анализ текста (извлечение информации из текста), т.е. речевую коммуникацию. Изменяются ли эти характеристики в зависимости от особенностей коммуникативной ситуации? Коммуникация может быть устной или письменной.

Язык, обеспечивающий эффективную устную коммуникацию, не может не отличаться от языка, обеспечивающего письменную коммуникацию. Каждый из носителей письменного языка (успешно овладевший письменным языком) может по праву называться билингвом: человеком, владеющим двумя – устным и письменным – языками и умеющим переключаться с одного языка на другой (с одного кода на другой) в зависимости от требований коммуникации. Следующий тезис: информация заключена в тексте (не в языке), но текст строится и анализируется с использованием языка. Значит, легко допустить, что тексты существенно разного типа накладывают свои требования на используемый язык. Речь идет, прежде всего, о текстах, различающихся по степени и типу информационной нагруженности: о текстах разных функциональных стилей.

Обычно выделяют следующие функциональные стили (одна из самых грубых классификаций): разговорный (бытовой диалог), литературно-художественный, газетно-публицистический (новостной), научный, деловой (официально-деловой). Нас интересует, прежде всего, (1) степень и тип информационной насыщенности, (2) основной тип контекста и (3) жесткость

композиционной структуры (два последних фактора рассматривается в следующей главе).

Как-то укоренилось, что термин «прикладная лингвистика» в зарубежной и отечественной науке имеют существенно различное значение. В зарубежной науке лингвистика «прикладывается», прежде всего, к такой безусловно прикладной задаче, как обучение языку. Наше понимание термина ближе всего к компьютерной или вычислительной / машинной / инженерной лингвистике (наша специальность «Прикладная и математическая лингвистика» (в номенклатуре ВАК) за рубежом скорее всего найдет себе аналоги на факультетах Computer Science).

Когда и зачем нужны лингвисты? Лингвисты несколько лучше представляют себе «физическую» природу объекта моделирования. Языковая система уникальна в том смысле, что она полностью не починяется законам ни естественнонаучного, ни гуманитарного познания. Язык многие рассматривают как творение человека, но это в существенной степени заблуждение. Пожалуй, так никогда не скажет лингвист. Язык – объект принципиально особого свойства. Он сосуществует в природе совместно с человеком (ср. разнообразные варианты рассмотрения гипотезы лингвистической относительности, т.е. степени взаимообусловленности человека, языка и социума (цивилизации). Для моделирования языковой системы используются инструменты моделирования, пришедшие из физики, из экономики (и/или социологии), из физиологии, из философии и семиотики (теории о знаках). Лингвистика – хорошая лингвистика – должна уметь оценить рассматриваемый объект во всех этих плоскостях (быть междисциплинарной), конечно, если лингвистика – действительно наука о языке. Вернее сказать, это наука о языке (языковой системе) и тексте, формах и способах функционирования этой системы. Может ли на начальном этапе – этапе постановки задачи – компьютерная лингвистика обойтись без лингвиста? Вряд ли. Может ли хотя бы на начальном этапе лингвист обойтись без инструментария смежных дисциплин? Безусловно, нет.

Наряду с единичными текстами, которыми и раньше занимались лингвисты, объектом лингвистики становятся и коллекции текстов, и информационные потоки как объекты нового информационного пространства.

В качестве единицы анализа (письменного) текста в работах используются, прежде всего, такие стандартные единицы, как лексема и словоформа. Когда и какая из этих единиц важнее – решать исследователю, и выбор задается целью и задачами работы.

Впрочем, отметим, что роль словоформы как основной единицы восприятия (анализа) текста подтверждается психолингвистическими экспериментами (особенно для звучащей речи). Для звучащего текста в качестве основной единицы первичного анализа используются фонетические слова.

При восприятии и порождении (анализе и синтезе) текста неизбежно используются единицы разного масштаба, разной степени связанности и разных уровней иерархии. Эти единицы «задаются» характеристиками языка и контекста, предпочтение тех иных единиц имеет ярко выраженную вероятностную природу. В качестве такого рода оперативных единиц могут выступать как синтаксические, так и лексические единицы (под последними понимаются разнообразные обороты, единицы, эквивалентные слову и т.д. – см., напр., и словарь оборотов www.ruscorpora.ru/obgrams.html).

В современной лингвистике, ориентированной, с одной стороны, на функциональность и антропоцентричность описания, а с другой стороны – на возможности корпусной лингвистики, уже практически очевидна необходимость использования основных положений грамматики конструкций и близких к ней научных направлений. Подход «GxC» (грамматики конструкций) начал разрабатываться с 1970х годов и чрезвычайно популярен в разных направлениях современной лингвистики (подробную библиографию см. в http://constructiongrammar.org/).

В рамках парадигмы корпусных и когнитивных исследований нас интересует изучение лексико-грамматических явлений (вернее было бы даже

сказать: лексических и морфолого-синтаксических явлений) при восприятии и порождении (анализе и синтезе) текста. Поэтому для нас наиболее интересным является объединение идей, заложенных в моделях грамматики конструкций и различных контекстно-ориентированных моделях (от широко известной «Контекстуальной теории значения» (Contextual Theory of Meaning) Ферса.

Как известно, в процедурах обработки текста происходит максимальная опора на контекст. Причем понятие «контекст» также рассматриваться в разных смыслах. Для нас контекст предполагает широкое понимание: минимальный контекст, в котором реализуются лексические и морфолого-синтаксические явления; текстовый контекст, включающий в себя фрагменты текста вплоть до текста целиком; контекст коллекции (базы текстов), предполагающий учет текстов определенного типа (заданного функционального стиля, отобранной коллекции текстов и т.д.).

Можно было бы добавить еще одно понимание контекста: как совокупности текстового опыта человека, а также тем самым – знание языка (на основании опыта по восприятию и порождению текстов). Такое понимание «широкого контекста» в известной степени моделируется в создании и последующем изучении Национальных корпусов.

Процедуры обработки текста носят вероятностный характер. Безусловно вероятностный характер носит обработка (восприятие, понимание) текста человеком. О вероятностном характере процедур обработки текста мы можем говорить в отношении многих систем автоматического понимания текста (ср., напр., системы кластеризации новостных текстов на новостных порталах или машинный перевод, основанный на статистическом анализе). Возможны, наконец, процедуры автоматического анализа текста, моделирующие стратегии обработки текста человеком.

Степень связанности конструкций, по всей видимости, зависит от вероятностной модели, описывающей появление этой конструкции в ходе процедур обработки текста. Вероятные оценки могут быть получены лишь на основании статистических данных. Причем статистические характеристики

должны описывать данные в зависимости от перечисленных выше типов контекста.

Попытки последовательно учитывать контекст (причем – как указывалось выше – разные типы контекстов) ставят перед исследователем дополнительные задачи. Обычно получаемые в работах списки коллокаций лишь в некоторой степени могут быть соотносимы с исследованием тех особенностей, которые не просто заложены в языке (всех текстах на этом языке), но в существенной степени зависят от типа контекста (напр., от функционального стиля текстов, конкретной коллекции или отдельного текста по отношению к этой коллекции).

Реализовать контекстно-ориентированный подход можно с использованием различных статистических мер, позволяющих автоматически выделить из текстов коллокации и ранжировать их по степени неслучайности в соответствии со значениями выбираемых мер. При этом нечеткое и интуитивное понятие контекста принимает черты объективности – в узком смысле под контекстом понимается та коллекция, на которой проводится исследование. Возможность варьировать коллекции (например, выбирая коллекции текстов разных функциональных стилей или даже отдельные тексты из этих коллекций) позволяет получать списки коллокаций, различающие различные контексты. Именно текстовый материал, реализация лексико-грамматических и синтаксических проявлений, оказывается базой для исследования.

А поскольку тексты написаны на естественных языках, мы считаем, что лучше уточнить значение термина «естественный язык» в контексте данного доклада. Под «естественным языком» мы будем понимать язык, используемый людьми для ежедневного общения; например, такие языки как английский, португальский, хинди.

В отличие от искусственных языков, таких как языки программирования или математических обозначений, естественные языки эволюционировали от поколения к поколению и их трудноограничить четкими правилами. Мы

будем понимать Natural Langugage Processing (Обработку естественного языка – ОЕЯ для краткости) в широком смысле, чтобы включить все типы компьютерного манипулирования естественным языком.

С одной стороны все может быть так же просто, как и подсчет частоты использования слов для сравнения различных стилей написания. С другой стороны ОЕЯ включает «понимание» завершенных человеческих высказываний, хотя бы в той степени, с которой появляется возможность дать полезных ответ на них.

Технологии, основанные на ОЕЯ, становятся все более распространенными. К примеру, телефоны и мобильные компьютеры поддерживают предсказывание текста и распознавание рукописи, движки веб-поиска дают доступ к информации, скрытой в неструктурированном тексте, машинный перевод позволяет получать текст написанный на китайском и читать его на испанском. Путем предоставления более естественного интерфейса человек-машина и более сложного доступа к хранимой информации обработка языка стала играть центральную роль в многоязыковом информационном сообществе.

Изучение данной технологии позволит студентам филологического факультета работать с языковой информацией, проводить ее анализ и разрабатывать программы для данных целей: обеспечение базами данных в сфере искусства и гуманитарных наук – управление большими языковыми корпусами, исследование лингвистических моделей, построение роботизированных систем для выполнения лингвистических задач с технологическими приложениями; обеспечение базами данных в сфере науки и инженерии – применение различных техник в моделировании данных, поиске информации, а также открытие знаний об анализе естественного языка, применение лингвистических алгоритмов и информационных структур в роботизированном ПО обработки текстов.

Для обработки естественного языка, как один из возможных вариантов, используется язык программировния Python совместно с откытой библиотекой

источников под названием Инструментарий Естественного Языка (Natural Language Toolkit NTLK). NTLK включает обширный набор ПО, информации, документации, доступные для свободного скачивания с www.ntlk.org. Дистрибутивы выполнены для платформ Windows, Macintosh и Unix.

Python был выбран в качестве языка выполнения NLTK поскольку он достаточно прост для изучения, его синтакс и семантика достаточно ясны и у него хорошая функциональность в обработка строковых переменных.

Python является простым и в тоже время мощным языком программирования с великолепной функциональностью для обработки лингвистической информации. Python может быть установлен бесплатно с www.python.org. Установщики доступны для любых платформ.

Как интерпретируемый язык программирования Python способствует интерактивному исследованию. Как объектно-ориентированный язык программирования Python легко позволяет инкапсулировать и повторно использовать данные и методы. Python идет с обширной стандартной библиотекой, которая включает в себя инструментарий для графического программирования и обработки числовых данных, что означает, что он может быть использован для широкого спектра нетривиальных приложений. Python – идеален для обучения новичков и работы опытных программистов.

Python широко применяется в промышленности, научных исследованиях и обучении по всему миру. Python часто хвалят за его способы облегчения производительности, качетсва и надежности ПО. История успехов Python размещена на www.python.org/about/succes/.

Natural Language Toolkit (NLTK – Пакет Программ для обработки ествественного языка) широко используется для обучения студентов, изучающих обработку естественного языка в сфере лингвистики или вычислительной техники.

Относительно легко обучать ОЕЯ (обработке ествественного языка) в качестве отдельной дисциплины для разных групп студентов. Лингвисты могут быть обучены программе, создавая проекты, в которых студенты

манипулируют с их собственными данными. Специалисты по вычислительной технике могут изучить методы автоматической обработки текста, участвуя в проектах по глубинному анализу текста (text-mining) и по созданию чат-роботов (chatbot – программа-робот системы групповых дискуссий в Internet).

NLTK был раразработан для того, чтобы дать студентам возможность углубить свои знания и умения в области ОЕЯ. В частности, NLTK дает возможность начать курс, которые покрывает существенное количество теории и практики для аудитории, состоящей как из лингвистов, так и из специалистов в области компьютерных наук. NLTK – пакет модулей, созданных при помощи языка программирования Python, выложенных в Internet на основе общедоступной лицензии на сайте www.nltk.org

NLTK предоставляется вместе с большой коллекцией корпусов, исчерпывающей документацией и сотнями упражнений, которые делают NLTK уникальным пакетом программ, в плане предоставления всесторонних корпусов для развития у студентов вычислительного понимания языка. Код NLTK основан на 100 000 строках кода в Python и поддерживает доступ к корпусам, токенизирование, морфологический поиск, тегирование, фрагментирование, синтаксический анализ, кластеризацию, классификацию, моделирование языка, семантическую интерпретацию, стандартизацию и другие функции.

Как пример развития в сфере науки, NLTK используется в более чем 60 вузах в 20 странах, список которых можно просмотреть на сайте NLTK.

С момента зарождения в 2001, NLTK претерпел значительные изменения, которые основывались на опыте преподавательской деятельности в нескольких университетах, в том числе на информации от многих преподавателей и студентов. По прошествии этого периода, серии практических онлайн-обучений NLTK разрослись до всеобъемлющей онлайн-книги.

Как упоминалось в статье Loper & Bird (2002), приоритеты кода NLTK фокусируются именно на роли обучения. Когда код хорошо написан, студент,

который не понимает математику скрытой марковской модели (алгоритм непрерывного распознавания речи), может извлечь пользу из наблюдения за тем, как осуществляется алгоритм. Подобный акцент ставится на расширяемость, поскольку это помогает обеспечению того, что код растет как связанная целостность, а не как непредсказуемые и случайные приложения.

Не смотря на то, что эффективность нельзя игнорировать, она заняла второе место наряду с простотой и ясностью кодирования. В том же духе, мы пытались избежать программистских хитростей, поскольку это типично препятствует ясности кода. В конце концов, широта сферы действий не была доминирующим вопросом NLTK, что дает множество возможностей для студенческих проектов и общественного участия.

Одно издание впитало в себя значительное количество внимания в области наименования ориантированных на пользователя функций NLTK. В большей степени, система наименований является пользовательским интерфейсом в инструментарии, и кажно, чтобы пользователи были способны угадать, какое действие может быть представлено данной функцией.Следовательно, правила наимнования должны быть четкими и семантически прозрачными. В то же время, есть компенсирующее давление в плане относительно коротких слов, поскольку чрезвычайное многословие может также помешать восприятию и удобству эксплуатации. Дополнительная сложность в том, что принимание\заимствование\выбор объектно-ориентированного стиля программирования может быть хорошо мотивировано по ряду причин, но тем не мнее является сложностью для студента-лингвиста. Например, хотя лучше всего осуществлять метод экземпляра WordPunctTokenizer().tokenize(text) (для некоторой строки введения text), упрозенная версия тоже предоставляется: wordpunct_tokenize(text).

Большое количество упражнений и проектов, которые студенты могут выполнить, сильно расширяется включением большой коллекции корпусов, вместе с удобными программами для чтения корпусов. Эта коллекция, которая

на данный момент состоит из 45 корпусов, включает в себя структурно проанализированные, пост-тегированные, простые тексты, категоризированные тексты, и лексиконы.

В создании программ для чтения корпусов был сделан акцент на простоте, логичности и эффективности. Корпусные объекты (corpus objects), такие как nltk.corpus.brown и nltk.corpus.treebank, определяют общие методы для чтения корпусного контета, абстрагируясь от идиосинкратического формата файлов, чтобы предоставить унифицированный интерфейс.

Объекты корпуса предоставляют методы для загрузки контента корпуса различными путями. Общие методы включают: raw() для необработанного контекта корпуса; words() для списка токенизированных слов; sents() для того же списка, сгруппированного в предложения; tagged_words() для списка из пары слово-тег; tagged_sents() для тех же пар, сгруппированных в предложения; parsed_sents() для дерева синтаксического разбора. Опциональные параметры могут быть использованы для ограничения количества информации, запрашиваемой в корпусе, к примеру, определенная часть или индивидуальный файл в корпусе.

Большинство методов, используемых в программах для чтения корпусов, возвращают изображение корпуса (corpus view), которые действует как список текстовых объектов, но поддерживает быстроту реакции и эффективность памяти посредством загрузки тех пунктов из файла, которые являются необходимой базой. Таким образом, когда мы печатаем изображение корпуса, мы загружаем только первый блок корпуса в память, но когда мы работаем непосредственно с объектом из корпуса, мы загружаем весь корпус:

```
>>> nltk.corpus.alpino.words()
['De', 'verzekeringsmaatschappijen', 'verhelen', …]
>>> len(nltk.corpus.alpino.words())
139820
```

Когда пользователь скачивает NLTK-библиотеку, ему впоследствии будет необходимо скачать дополнительные данные, к примеру, коллекцию

литературных текстов. Это можно осуществить используя определенные команды в интерпретаторе Питона. В последствии с этими текстами можно будет осуществлять различные операции, к примеру подбор конкорданса, который покажет частоту употребления слова в определенном тексте, либо может подобрать слова, похожие по значению или употребляемые в похожем контексте. Также осуществимы операции по определению длины текста. Длина текста определяется по числу токенов. Токеном считается любое недублируемое слово или пунктуационный знак. Также возможно составлять списки слов.

Текст можно рассматривать как совокупность знаков на странице. С другой стороны, текст – это совокупность глав, состоящих из подглав, в которых есть параграфы и так далее. В целях изучения обработки текста при помощи языка Python, мы будем рассматривать текст как совокупность слов и пунктуации.

NTLK может обеспечить исследователя базовыми классами для представления информации, относящейся к обработке естественного языка; стандартными интерфейсами для выполнения задач таких, как маркировка частей речи, синтаксический разбор, классификация текста; стандартными исполнениями для каждого задания, которое может быть скомбинировано для решения комплекса проблем (комплексной проблемы).

NTLK сопровождается обширной документацией. Сайт www.ntlk.org предоставляет API документацию по каждому из модулей, классов и функций из инструментария, специфицируя параметры и раскрывая примеры использования. Сайт также предоставляет множество HOWTO с обширными примерами и тестовыми заданиями, предназначенными для пользователей, разработчиков, инструкторов.

ЛИТЕРАТУРА:

1. Steven Bird, Ewan Klein, Edward Loper, Jason Baldridge “Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics:

Multidisciplinary Instruction with the Natural Language Toolkit ”. Columbus, Ohio, USA, June 2008. – P.62 –70.

2. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. — М.: МИЭМ, 2011.

3. Steben Bird, Ewan Klein, Edward Loper “Natural Language Processing with Python”. O’Reilly, 2009.