**Содержание**

**­Введение**

Анализ текстов на естественном языке – одна из передовых и приоритетных областей IT-индустрии. Уже сейчас разработки в этой сфере имеют огромную прикладную значимость.

В настоящее время автоматический анализ текста становится все более актуальным, и набор необходимых прикладных задач в этой области быстро растет. Самая острая потребность в них сегодня заключается в повышении доступности информации. Мы живем в эпоху огромного роста объемов информации, и ее количество превращается в фактор ограничения. Обычный человек просто не может воспринимать все данные, в которых нуждается, и это делает задачу автоматического обзора текста, аннотаций и иллюстраций жизненно важной для дальнейшего развития. Это особенно актуально для научных и учебных текстов. Современные технологии позволяют нам разрабатывать алгоритмы для преобразования массивов текстов в более воспринимаемую форму.

Однако количество исследований в этой области невелико, особенно для русского языка. Это связано как с высокой сложностью задачи, так и с отсутствием размеченных корпусов для нее.

Целью данной работы является разработка программы для полуавтоматической генерации размеченных корпусов на русском языке, в дальнейшем пригодных для семантического парсинга текстов.

**1 Постановка задачи и обзор предметной области**

**1.1 Понятие корпуса**

Корпус — это коллекция взаимосвязанных документов (текстов) на естественном языке. Корпус может быть большим или маленьким, но обычно состоит из десятков и даже сотен гигабайт данных в тысячах документов. Корпусы могут быть аннотированными, то есть текст или документы могут быть снабжены специальными метками для алгоритмов обучения с учителем (например, для фильтров спама), или неаннотированными, что делает их кандидатами на тематическое моделирование и кластеризацию документов (например, для изучения изменений в темах, скрытых в сообщениях, с течением времени) [2].

**1.2 Понятие дискурса**

Дискурс – это сложное, многозначное понятие, используемое в семантике, социолингвистике, дискурсивной психологии и других лингвистических и окололингвистических областях науки [5].

Под дискурсом мы будем понимать связную последовательность лингвистических объектов: слов или предложений, – совокупный смысл которых больше простой суммы смыслов этих объектов по отдельности [6].

Дискурс, как и большинство лингвистических явлений, имеет иерархическую природу [8]. Фактически, дискурс – это комбинация паттернов, описывающих структуру текста. Часто под дискурсом понимаются единицы смысла, состоящие более чем из одного предложения. Минимальной единицей дискурса будем считать словосочетание, то есть два или более синтаксически согласованных и/или синтаксически связанных слова.

**1.2 Теория риторических структур**

Одна из основных идей для преобразования массивов текстов в более воспринимаемую форму – представить текст в качестве риторической структуры. Согласно этому подходу, первоначально текст делится на непересекающиеся фрагменты, а именно элементарные дискурсивные единицы (EDU), иногда их называют клаузами. Кроме того, последовательные EDU взаимосвязаны риторическими отношениями. Эти части известны как элементы, из которых строятся большие фрагменты текстов и целые тексты. Каждый фрагмент по отношению к другим фрагментам выполняет свою определенную роль. Текстовая связность формируется с помощью тех отношений, которые моделируются между фрагментами в тексте. Каждая EDU добавляет что-то для реализации коммуникативного намерения говорящего/автора (отсюда термин «риторическая»).

Теория риторических структур (RST) (Манн и Томпсон, 1987, 1988) основана на анализе более ста текстов. Манн и Томпсон утверждают, что можно проанализировать большинство текстовых типов в терминах иерархического дерева риторических отношений. Такой анализ подразумевает:

1) разбиение текста на элементарные дискурсивные единицы (ЭДЕ);

2) установление между дискурсивными единицами риторических отношений;

3) построение риторических деревьев, отображающих структуру исследуемого дискурса.

Анализ основан на предположении, что некоторые текстовые единицы являются более важными (значимыми) для текста, чем другие [9]. Важная единица – это ядро, которое считается наиболее приоритетной частью высказывания, в то время как менее важная единица, называемая сателлитом, объясняет ядра и считается вторичной. Ядро содержит основную информацию, а сателлит хранит дополнительную информацию о ядре. Таким образом, риторическая структура текста является способом описания функционального аспекта текстовых элементов.

EDU связаны между собой риторическими отношениями (РО), связанные между собой они образуют группы. Группы же в свою очередь связываются между собой теми же самыми риторическими отношениями. Так объединение частей идет иерархически вплоть до дискурса в целом.

Типы риторических отношений бывают двух видов:

* симметричные (ядро + ядро (+ ядро …));
* асимметричные (ядро + сателлит).

Дискурс устроен иерархически, и для всех уровней иерархии используются одни и те же риторические отношения. Сейчас выделяют около сорока риторических отношений, в таблицах 1.1 и 1.2 приведены наиболее распространенные из них:

Таблица 1.1 – Список асимметричных отношений.

|  |  |
| --- | --- |
| *Английский термин* | *Русский термин* |
| Evidence | Свидетельство |
| Volitional cause | Волитивная причина |
| Restatement | Переформулировка |
| Concession | Уступка |
| Volitional result | Волитивный результат |
| Antithesis | Антитезис |
| Elaboration | Развитие, детализация |
| Non-volitional cause | Неволитивная причина |
| Solutionhood | Решение |
| Motivation | Мотивация |
| Non-volitional result | Неволитивный результат |
| Enablement | Обеспечение возможности |
| Justify | Обоснование |
| Circumstance | Обстоятельство |
| Purpose | Цель |
| Condition | Условие |
| Background | Фон |
| Interpretation | Интерпретация |
| Evaluation | Оценка |
| Otherwise | Альтернатива |
| Summary | Резюме |
| Means | Средство |

Таблица 1.2 – Список симметричных отношений.

|  |  |
| --- | --- |
| *Английский термин* | *Русский термин* |
| Sequence | Последовательность |
| Contrast | Контраст |
| Joint | Конъюнкция («и») |

Многие РО похожи на обстоятельственные отношения в сложноподчиненном предложении – RST фактически распространяет обстоятельственные отношения на дискурс в целом. При этом интерпретационные решения в ТРС не зависят от синтаксической структуры.

Приведем пример, в паре клауз:

1. риторическое отношение цели, при этом первая часть является главной и представляет собой ядро, а вторая является зависимой, сателлитом:

*Иван вышел рано, чтобы не опоздать на встречу.*

1. симметричное отношение – последовательность:

*Пришёл, увидел, победил.*

RST – наиболее удачная теория дискурса, так как:

1. предоставляет общий формат для описания глобальной и локальной структуры дискурса;
2. позволяет в более общем виде описать другие результаты – например, процесс реферирования;
3. позволяет описать некоторые жанровые явления: например, рассказ – это такой дискурс, который на верхнем уровне имеет отношение «последовательность»;
4. позволяет единым образом описать многие языковые явления.

Отношения RST применяются рекурсивно в тексте до тех пор, пока все единицы в этом тексте не станут составляющими в отношении RST. Результатом такого анализа является то, что структура RST обычно представлена ​​в виде деревьев с одним отношением верхнего уровня, которое охватывает другие отношения на более низких уровнях.

Хотя теория риторических структур (RST) является одной из наиболее широко используемых теорий организации текста, недостаточное количество помеченных корпусов и нехватка списков маркеров и коннекторов для определения отношений являются существенным препятствием на пути применения этого подхода для русских научных текстов [3].

**1.3 Penn Discourse Treebank**

*Penn Discourse Treebank (PDTB)* состоит из 385 статей Wall Street Journal, аннотированных структурой дискурса в структуре RST, а также созданных человеком выдержек и рефератов, связанных с исходными документам, представляет собой крупномасштабный корпус, снабженный информацией, касающейся структуры и семантики дискурса. Хотя существует много аспектов дискурса, которые имеют решающее значение для полного понимания естественного языка, PDTB фокусируется на кодировании дискурсивных отношений. PDTB стремился поддерживать нейтральный к теории подход в отношении характера представления структуры дискурса на высоком уровне, чтобы корпус мог использоваться в различных теоретических рамках. Теоретически нейтральность достигается путем сохранения аннотаций дискурсивных отношений на «низком уровне»: каждое дискурсивное отношение аннотируется независимо от других отношений, то есть зависимости между отношениями не отмечены.

PDTB предназначен для извлечения ряда выводов, связанных с дискурсивными отношениями, для широкого спектра приложений NLP, таких как анализ, извлечение информации, ответы на вопросы, обобщение, машинный перевод, генерация, а также исследования на основе корпуса в лингвистике и психолингвистике.

PDTB строится непосредственно поверх Penn Treebank и Propbank, тем самым поддерживая извлечение полезных синтаксических и семантических функций и обеспечивая более богатую основу для разработки и оценки практических алгоритмов.

Корпус PDTB содержит аннотации четырех типов соединительных элементов: подчиняющие соединения, координирующие соединения, связные наречия и неявные связки.

В настоящее время аннотации выполняются четырьмя аннотаторами. Индивидуальная аннотация выполняется по одному соединению за раз. WordFreak, инструмент аннотирования, используемый аннотаторами, идентифицирует все экземпляры данного соединения в корпусе, которые затем аннотируются независимо четырьмя аннотаторами. Таким образом, аннотаторы быстро получают опыт работы с этим связующим звеном и лучше понимают его характеристики предиката-аргумента. Точно так же для аннотации неявных связок все экземпляры идентифицируются по одному тексту за раз.

Для выполнения этой задачи аннотаторы должны прочитать весь текст, чтобы они могли принимать обоснованные и надежные решения о неявных связях и их аргументах. Кроме того, после определения аргументов каждой неявной связки аннотаторы предоставляют, если возможно, явную связку, которая лучше всего выражает предполагаемое отношение.

Краткий обзор классов соединительных элементов, которые аннотируются:

1. Подчиняющие соединения вводят предложения, которые синтаксически зависят от основного предложения. Наиболее распространенными типами отношений, которые они выражают, являются временные (например, «когда», «как только»), причинно-следственные (например, «потому что»), концессионные (например, «хотя», «даже тогда»), цели (например, «так что») и условные (например, «если», «если»). Пункты, введенные с подчиненным соединением, могут быть переделаны относительно основного пункта, как показано в (1).
2. *Поскольку [засуха сократила запасы в США], [у них более чем достаточно места для хранения нового урожая], и это позволяет им ждать повышения цен.*
3. Координирующие соединения содержат соединительные элементы, такие как «и», «но» и «или».
4. Связные наречия выражают дискурсивное отношение между двумя событиями или состояниями, например, «однако», «следовательно», «тогда» и т. д. В этот класс также включены предложные фразы, которые выражают аналогичные отношения, такие как, «в результате», «в дополнение», «фактически» и т. д. Пример (2) показывает аннотацию экземпляра связного наречия «в результате».
5. *... [многие аналитики ожидали, что цены на энергоносители также будут расти на уровне потребителей]. В результате [многие экономисты ожидали, что индекс потребительских цен увеличится значительно больше, чем он сделал].*
6. Неявные связки идентифицируются между смежными предложениями, которые не связаны явным соединением. Аннотация неявных связок предназначена для захвата дискурсивных отношений, которые неявно выражены между смежными предложениями. Аннотаторам предлагается предоставить явную связку, которая наилучшим образом описывает предполагаемое отношение. Например, явная связка, представленная в (3), «В отличие от этого».
7. *... [6 миллиардов долларов, которые около 40 компаний ожидают получить в течение года, заканчивающегося 31 марта, сопоставимы с только 2,7 миллиардами долларов, привлеченными на рынке капитала в предыдущем финансовом году]. (В отличие от этого) [В 1984 году, до того как г-н Ганди пришел к власти, было собрано всего 810 миллионов долларов].*

Поскольку берутся отношения между абстрактными объектами, требуется, чтобы аргумент содержал хотя бы один предикат вместе со своими аргументами. Следовательно, аргументом может быть одно предложение, последовательность предложений или их комбинации.

Таким образом, PDTB включает аннотации дискурсивных связок и их аргументов. Высокая точность – порядка 90%, указывает на то, что дискурсивные соединительные элементы и их аргументы раскрывают четко определенный уровень структуры дискурса, который можно надежно аннотировать.

Однако корпуса состоят в основном из новостей и их трудно масштабировать до текста других жанров, в частности, до научных и образовательных текстов. Более того, исследователи, создающие аннотированный корпус для русского языка, обнаружили, что список типов отношений должен быть расширен [13].

**1.4 GENIA**

GENIA – семантически аннотированный корпус для биотекстирования. Корпус GENIA разрабатывается для предоставления справочных материалов, позволяющих методам NLP работать с биологическими текстами.

На данный момент выпущена версия GENIA corpus 3.0, состоящая из 2000 тезисов MEDLINE, с более чем 400 000 слов и почти 100 000 аннотаций для биологических терминов.

GENIA представляет собой сборник статей, извлеченных из базы данных MEDLINE. В корпусе GENIA статьи кодируются в схеме разметки на основе XML-based, где каждая статья содержит свой ID, заголовок и реферат в указанном порядке, а все тексты в рефератах сегментированы на предложения, конфигурация проиллюстрирована на рисунке 1.1.

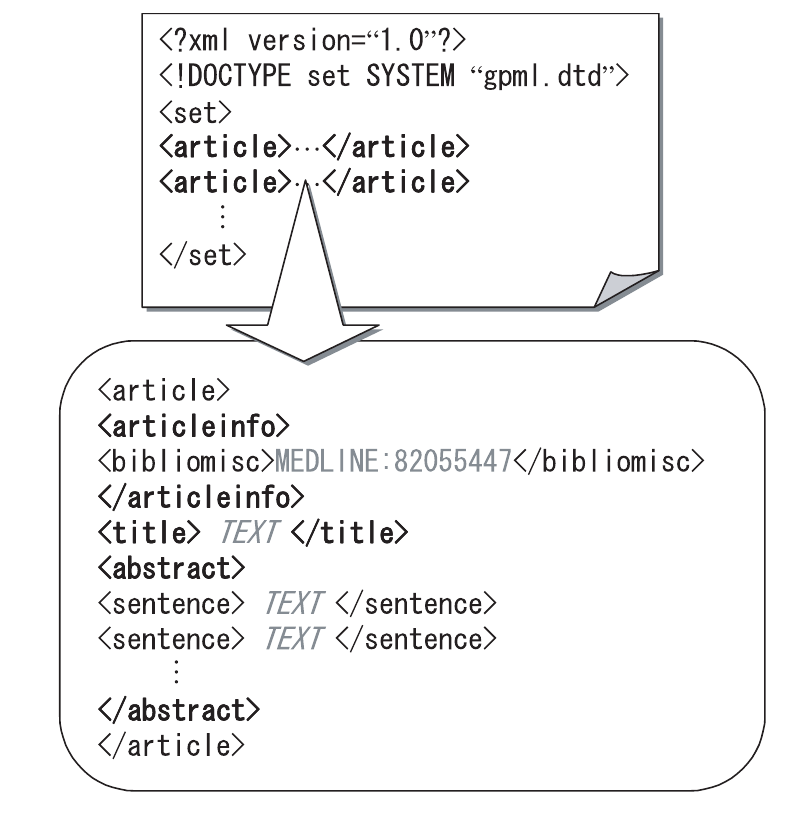


Рисунок 1.1 – Конфигурация корпуса GENIA

Основная ценность корпуса GENIA исходит из его аннотации: все рефераты и их названия были размечены двумя экспертами в области биологически значимых терминов, и эти термины были семантически аннотированы дескрипторами из онтологии[[1]](#footnote-1) GENIA.

Онтология GENIA представляет собой таксономию[[2]](#footnote-2) 47 биологически значимых номинальных категорий. На рисунке 1.2 показана иерархия онтологии GENIA, где в крайнем левом столбце перечислены три основных понятия: биологический источник, биологическое вещество и другие. Здесь «другое» фактически не является биологическим понятием, а подготовлен для терминов, которые рассматриваются как биологические понятия, но не отождествляются с какими-либо другими понятиями в онтологии. Более классифицированные концепции перечислены справа от соответствующих основных концепций. Понятия, выделенные жирным шрифтом, являются терминальными понятиями, и они образуют фактический набор тегов для семантической аннотации.

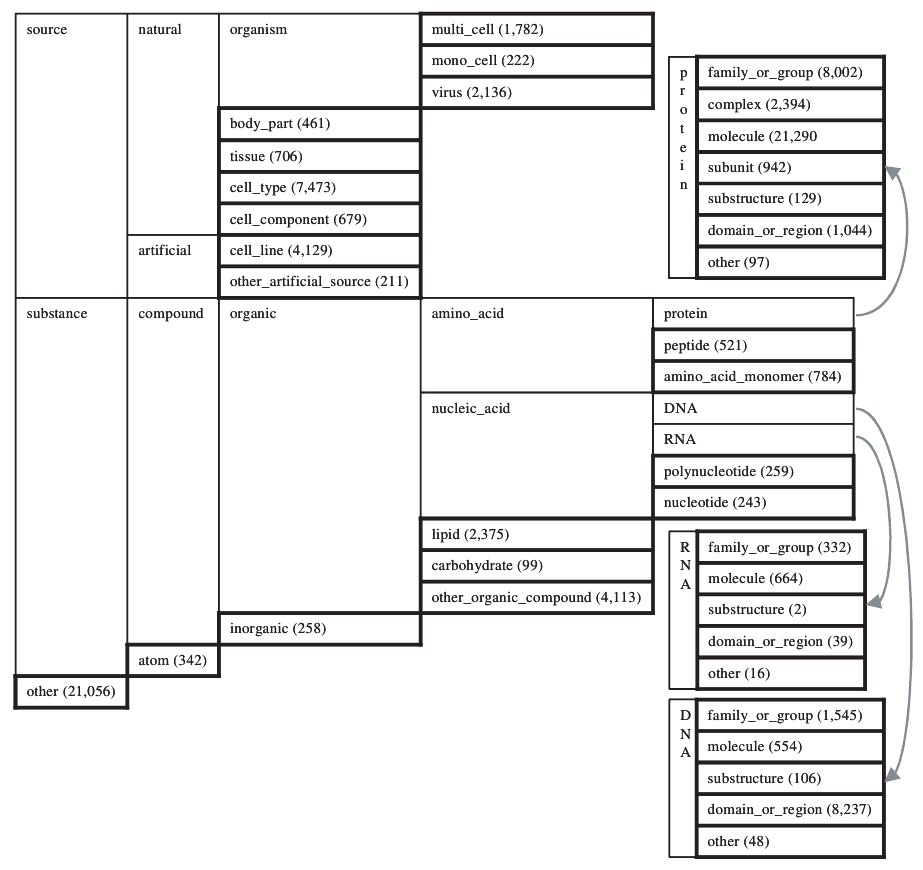


Рисунок 1.2 – Онтология GENIA

Онтология GENIA кодируется в DAML+OIL языке онтологии на основе XML и поставляется вместе с корпусом GENIA.

В корпусе GENIA биологические термины аннотированы. Семантически они определены как термины, которые можно отождествить с любыми терминальными понятиями в онтологии GENIA. Однако синтаксически они не просто определены. Часто термины, особенно имена биологических сущностей, сравнивают с именами именованных сущностей в основном собственными существительными, в то время как термины, включающие имена биологических сущностей, являются в основном общими существительными, и они могут появляться в тексте с различными спецификаторами или квалификаторами. Таким образом, требуется политика включения или исключения спецификаторов и квалификаторов в терминах. В схеме аннотации спецификации не включены в термины, и включение квалификаторов оставлено на усмотрение экспертов.

**1.5 SAPIENT**

SAPIENT – веб-приложение для аннотации на основе предложений полных статей с семантической информацией. SAPIENT позволяет экспертам аннотировать научные статьи на английском языке предложение за предложением, а также связывать предложения вместе, образуя, таким образом, участки важных частей дискурса. Как часть системы, разработан разделитель предложений с поддержкой XML (SSSplit), который сохраняет разметку XML и идентифицирует предложения путем добавления встроенной разметки. SAPIENT использовался в систематическом исследовании для аннотации научных статей с концепциями, представляющими основную информацию о научных статьях (CISP), для создания корпуса из 225 аннотированных статей.

Важным аспектом системы является то, что, хотя аннотация основана на предложениях, система обслуживает идентификаторы, которые связывают воедино предложения, относящиеся к одной и той же концепции. SAPIENT также включает в себя функцию OSCAR3 для автоматического распознавания именованных химических объектов и работает в браузере, что делает его независимым от платформы. SAPIENT принимает в качестве входных данных полные научные статьи в формате XML, разбивает их на отдельные предложения, отображает их и позволяет пользователю аннотировать каждое предложение одной из 11 концепций CISP, а также связывать предложение с другими предложениями, относящимися к тому же экземпляру выбранной концепции.

SAPIENT разработан в форме веб-приложения, чтобы сделать его независимым от платформы и облегчить его включение в качестве части рабочего процесса в Интернете.

В текущей реализации SAPIENT раскрывающееся значение типа соответствует выбору одной из 11 общенаучных концепций, а именно «Предпосылки», «Заключение», «Эксперимент», «Цель», «Гипотеза», «Метод», «Модель», «Мотивация», «Объект исследования», «Наблюдение», «Результат». Эти метки происходят из набора метаданных (Основная информация о научных концепциях (CISP). Идентификаторы концепции однозначно идентифицируют экземпляр концепции, но не предложение, то есть идентификаторы концепции обозначают и связывают вместе экземпляры одной и той же семантической концепции, распределенные по разным предложениям, которые могут быть в разных частях документа.

Различие между идентификаторами предложений и идентификаторами понятий является важной характеристикой системы. Это означает, что система не обязательно предполагает соответствие «1-1» между предложением и понятием, а скорее то, что понятия могут быть представлены интервалами часто непересекающегося текста. Следовательно, SAPIENT косвенно позволяет аннотировать дискурсивные сегменты за пределами уровня предложения, а также отслеживает совместные предложения.

Несмотря на то, что SAPIENT изначально был разработан для работы с концепциями CISP, его можно использовать для аннотирования документов в соответствии с любой схемой аннотаций на основе предложений.

Причиной разработки собственного сплиттера предложений было то, что широко распространенные сплиттеры предложений не могли правильно обрабатывать XML. Разметка XML содержит полезную информацию о структуре и форматировании документа в виде встроенных тегов, что важно для определения логической структуры документа [14].

**1.6 Специфика Википедии**

Википедия – один из наиболее доступных источников данных. В отличие от других возможных источников Википедия имеет следующие достоинства:

– Большой объем. Русский сегмент энциклопедии содержит более 1 551 000 статей [9].

– Документы Википедии имеют научную или близкую к научной лексику и структуру.

– Свободная доступность и бесплатность (распространяется под лицензией Creative Commons Attribution-ShareAlike 3.0 Unported) [9] .

– Актуальность и регулярная пополняемость. Википедия пополняется и поддерживается в актуальном состоянии силами огромного сообщества.

– Частичная структурированность (разметка). Тексты из Википедии снабжены достаточно подробными и стандартизованными метаданными. Также имеется подробная иерархия категорий.

Например, если мы возьмём статьи со схожей тематикой рисунки 1.3 и 1.4, то увидим, что они имеют одинаковую или схожую структуру построения документа с похожими названиями заголовков.

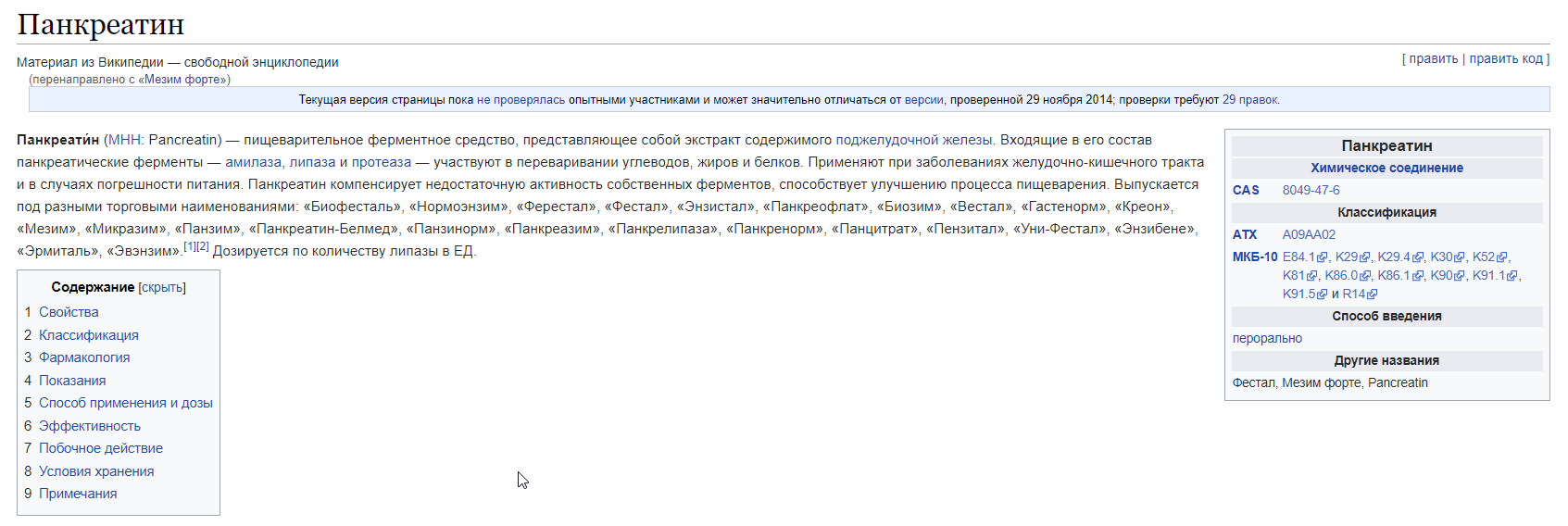
****

Рисунок 1.3 – Структура документа с названием «Панкреатин»

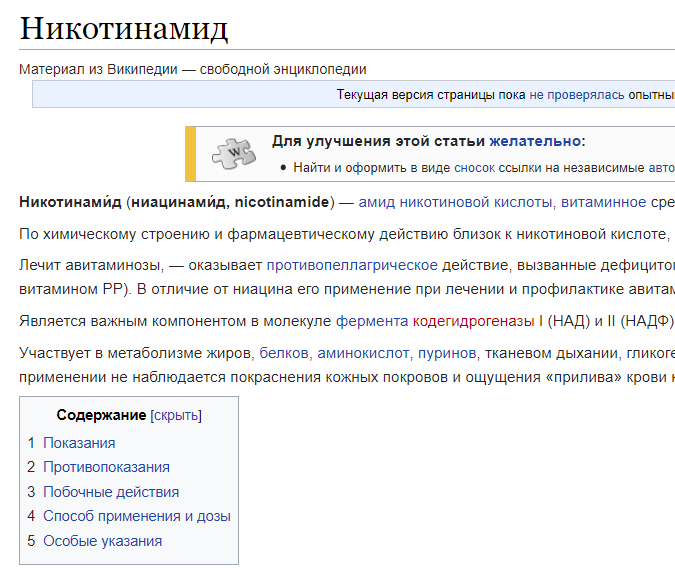


Рисунок 1.4 – Структура документа с названием «Никтотинамид»

Можем увидеть, что имеются одинаковые заголовки «Показания», «Способ применения и дозы», «Побочные действия» и таких примеров 1000. Данное преимущество открывает возможности для легкого чтения и выбора нужных отрывков текста при составлении связанного одной тематикой корпуса.

Для работы с дампом[[3]](#footnote-3) Википедии была использована разработанная В. А. Крайвановой в рамках магистерского проекта библиотека парсинга дампа Википедии. На вход библиотеке подаётся список документов Википедии, размеченный с помощью XML, и файл, содержащий индексы всех категории в SQL формате.

**1.7 Постановка задачи**

Целью данной работы является разработка программы для полуавтоматической генерации размеченных корпусов для семантического парсинга текстов.

Для полноценного анализа предоставленных данных, было выявлено

несколько задач:

1. Провести анализ существующих корпусов с семантической разметкой;
2. Изучить возможности использования Википедии для генерации корпусов;
3. Спроектировать сценарий генерации корпуса;
4. Разработать архитектуру программы;
5. Обозначить форматы конфигурационных файлов;
6. Реализовать программу.

Пользователь программы – специалист по компьютерной лингвистике.

**2 Проектирование системы**

**2.1 Построения корпуса на основе Википедии**

2.1.1 Категории

На вход будущего приложения подаётся полный дамп статей Википедии и индексы категорий, находящиеся в открытом доступе [11] и содержащие более 4 400 000 статей на русском языке.

Каждая из статей в Википедии привязана к той или иной ветке в дереве категории по принципу иерархичности, что позволяет легко её вычислить в огромном перечне статей электронной энциклопедии [12]. На рисунке 2.1 пример принадлежности статьи «Витамин В12» различным существующим в Википедии категориям.



Рисунок 2.1 – Список категорий статьи «Витамин В12»

Первым шагом необходимо выделить категории необходимые для построения желаемого корпуса, а также исключить из списка те, категории, которые могут объединить в себе огромное количество других категорий, создавая цикл. Цикл может объединить в конечный корпус либо просто большое количество ненужных статей, либо включит в выборку абсолютно все статьи из Википедии. Например, такая ситуация может возникнуть если не исключить «Символы» из категории «Информатика».

2.1.2 Типы документов

Каждая статьи Википедии принадлежит к тому или иному типу документа, которых опытным путем было обнаружено и сформировано 13. Каждый тип содержит в себе несколько шаблонов и категорий принадлежащих ему документов. Например, список таких шаблонов и категорий для сформированного типа документов «DEVICE»:

1. Шаблоны (templates):
2. «Судно»;
3. «Космический аппарат»;
4. «Карточка цифрового фотоаппарата»;
5. «Карточка фотоаппарата»;
6. «Сотовый телефон»;
7. «Паровоз»;
8. «Тепловоз»;
9. «Танк2»;
10. «Танк»;
11. «Infobox Weapon»;
12. «Карточка оружия»;
13. «Гоночный автомобиль»;
14. «автомобиль»;
15. «Карточка ЛА»;
16. «Подводная лодка».
17. Категории (categories):
18. «Патроны\_по\_обозначению».

Убедиться, к какой именно категории или шаблону относится статья, можно, нажав на странице «править код» (рисунок 2.2).



Рисунок 2.2 – Пример просмотра типа документа (Шаг 1)

В открывшемся окне в текстовом поле можем видеть уже введённый текст для шаблона «автомобиль» (рисунок 2.3).

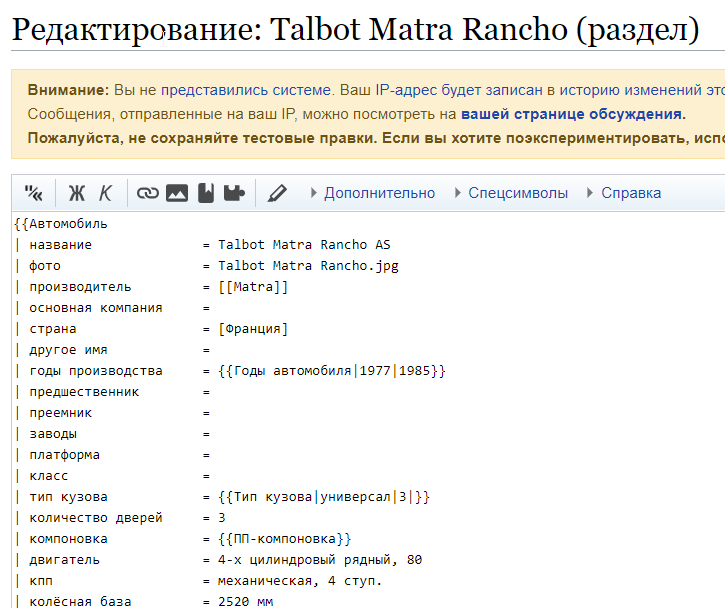


Рисунок 2.3 – Пример просмотра типа документа (Шаг 2)

Соответственно, если в какой-либо корпус не нужно включать конкретный тип документов, то его нужно исключить из общего списка нужных статей. Далее необходимо получить непосредственно все статьи, которые попадут в корпус.

2.1.3 Функциональная тема

*Функциональными элементами* текста являются слова, словосочетания, знаки препинания и другие элементы текста с точки зрения их роли в построении дискурсивной структуры текста. Список функциональных элементов, конечно же, зависит от темы и жанра текста, и в этой статье мы подразумеваем те, которые используются в научных, образовательных и энциклопедических текстах (SEET). Тем не менее, результаты могут быть обобщены на другие типы текстов: технические, экономические, юридические и новости.

Важно отметить, что помимо универсальных отношений RST, SEET имеет специфический аспект семантики, помещенной между темой и функциональной структурой. Этот аспект включает в себя особенности, характерные для стиля SEET. Например, предложение: «Окружность – замкнутая плоская кривая, которая состоит из всех точек на плоскости, равноудаленных от заданной точки» (Круг - это форма замкнутой плоскости, которая состоит из всех точек на плоскости, которые находятся на заданном расстоянии от заданного пункт) представляют собой определение понятия «Окружность» (Круг). Другая группа предложений может сформировать описание алгоритма или математическое доказательство. С одной стороны, такие виды текстовых областей образуют текстовую структуру и, следовательно, являются функциональными элементами. С другой стороны, в приведенном выше предложении есть тема «Определение окружности» (определение круга), поэтому определение является частью темы предложения. Таким образом, такой аспект может быть описан как *функциональная тема* (FT) текстового промежутка [3].

Наборы FT в SEET различны в различных областях науки, и ярким примером является различие между структурой статей Википедии для химического элемента, вида животных и алгоритма сортировки. Заголовки образцов этих групп представлены в Таблице 2.1. Несмотря на то, что не каждый заголовок из статьи в Википедии может быть классифицирован как FT, некоторые из них являются хорошими полупомеченными кандидатами на корпус текстовых интервалов FT.

Таблица 2.1 – Примеры структур статей

|  |  |  |
| --- | --- | --- |
| **Рубидий (Rubidium)** | **Западная серая белка (Western gray squirrel)** | **Сортировка пузырьком  (Bubble sort)** |
| 1 История (History)  2 Происхождение названия (Origin of the name)  3 Нахождение в природе (Occurrence in the nature)  4 Получение (Production)  5 Физические свойства (Physical properties)  6 Химические свойства (Chemical properties)  7 Применение (Application)  8 Изотопы (Isotopes)  9 Примечания (Notes)  10 Ссылки (References)  11 Литература (Literature) | 1 Описание (Description)  2 Распространение (Distribution)  3 Питание (Diet)  4 Размножение (Reproduction)  5 Естественные враги (Predation)  6 Значение (Competition)  7 Примечания (Notes)  8 Литература (Literature) | 1 Алгоритм (Algorithm)  2 Реализация (Implementation)  3 Пример работы алгоритма (Example of the algorithm work)  4 Примечания (Notes)  5 Ссылки (References)  6 Литература (Literature) |

Поэтому необходимо обозначить функциональную тему для будущего корпуса и выбрать к ней подзаголовки из Википедии. Например, для «Информатики» возьмём 4 FT: определение, теорема, алгоритм и приложение. Названия FT даны в соответствии с включенными в них подзаголовками Википедии. Определения FT включают довольно широкий класс описательных предложений. Теоремы включают в себя не только формулировки теорем, но и их фон: предположения, гипотезы и логические цепочки доказательств. Алгоритмы включают в себя любые последовательности процессов или действий. Приложения содержат применение концепций, описанных в статьях с точки зрения практики. Обратим внимание, что эти типы не являются взаимоисключающими, например, совершенно очевидно, что некоторые предложения из теоремы могут быть очень похожи на определения, а алгоритмы аналогичны доказательствам теорем и приложениям.

Конечно, качество этого корпуса невысокое по сравнению с корпусом ручной работы. Например, в статье «Прилив и заливка» в разделе «Терминология» есть фрагмент, описывающий поведение приливных волн, которое более точно соответствует FT алгоритма, нежели определению. То есть модель, построенная на этом корпусе, будет существенно уступать модели, построенной на помеченной вручную модели. В дальнейшем корпус может быть улучшен путем самостоятельного применения модели, когда предложения из корпусов классифицируются с использованием предложенного алгоритма, а при пересчете моделей FT используется только тот из них, которые были классифицированы по их первоначальному FT.



Рисунок 2.4 – Сценарий построения корпуса на основе Википедии

**2.2 Пользовательский сценарий**

Пользователь программы – специалист по компьютерной лингвистике, так как построение корпусов требует от пользователя соответствующих навыков и знаний, то рядовой пользователь может не справиться с программой должным образом.

Построение корпуса для пользователя было решено обеспечить по шагам.

Шаг 1. Пользователь вводит название нового корпуса и префикс для файлов конфигурации, которыми сам сможет воспользоваться для редактирования и отладки. Пока не введён префикс, дальнейшие шаги невозможны из-за необходимости сохранять действия в файлах конфигурации.

Шаг 2 и 3. Независимые шаги: выбор категорий и исключение типов документов. Результат помешается в файл конфигурации с введённым ранее префиксом.

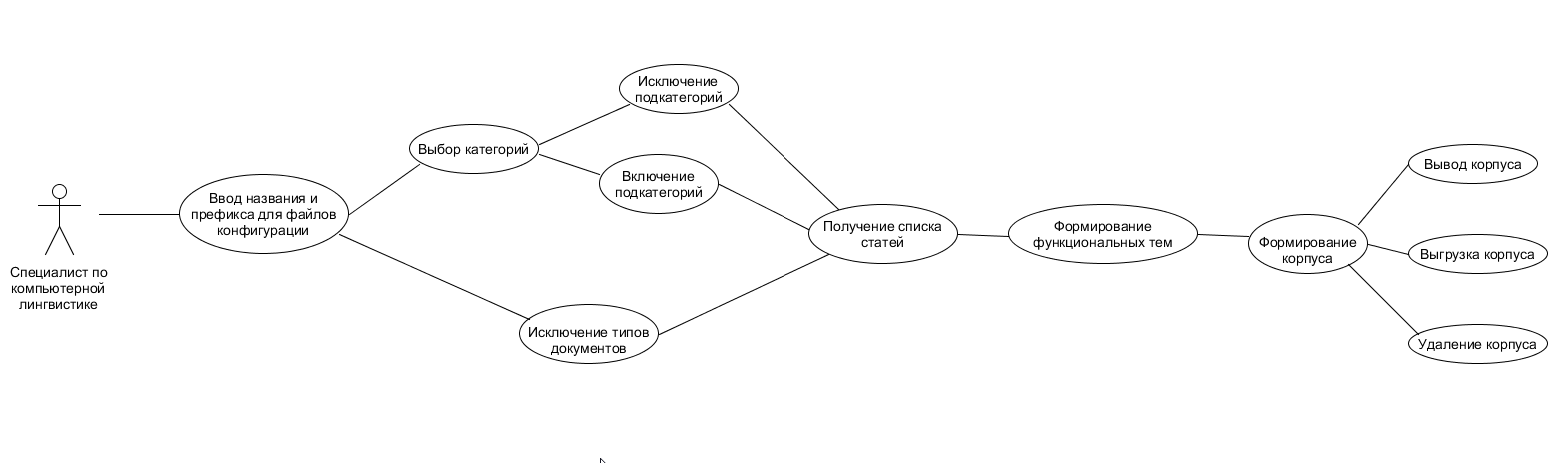
Шаг 4 и 5. Независимы друг от друга, но могут быть выполнены только после выбора категорий: выбор подкатегорий и исключение подкатегорий. Выбор подкатегорий необходим, если нужно рассмотреть небольшую часть огромных категорий.

Шаг 6. Может быть выполнен, как после 2 и 3, так и после 4 и 5, шагов: получение списка статей, которые попадут в корпус. Пользователю необязательно видеть все статьи целиком, поэтому выводиться будут лишь заголовки.

Шаг 7. Формирование пользователем необходимых ему функциональных тем.

Шаг 8. Формирование корпуса из полученных статей (шаг 6) с применением к ним функциональных тем (шаг 7).

Шаг 9, 10, 11. Пользователь имеет полное право распоряжения с полученным корпусом: выгрузка, вывод на экран или вовсе его удаление.

Рисунок 2.5 – Use-case диграмма

**3 Проектирование ПО**

**3.1 Выбор технологий**

Python – высокоуровневый язык программирования общего назначения. Синтаксис языка минималистичен, легко читаем и легок в изучении. Он является наиболее распространенным инструментом в машинном обучении. Также он имеет множество полезных библиотек [16].

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation [17].

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения [18].

Jinja — это шаблонизатор для языка программирования Python. Он подобен шаблонизатору Django, но предоставляет Python-подобные выражения, обеспечивая исполнение шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода. Лицензирован под BSD лицензией.

Шаблонизатор Jinja позволяет настраивать теги, фильтры, тесты и глобальные переменные. Также, в отличие от шаблонизатора Django, Jinja позволяет конструктору шаблонов вызывать функции с аргументами на объектах [19].

AJAX, Ajax (от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL).

**Заключение**

В ходе выполнения работы было реализовано web-приложение для составления корпусов предложений на основе сайта Википедия, добавлены функции просмотра, выгрузки и удаления результатов. Корпуса могут использоваться как для обучения моделей, нейронных сетей, так и для различных статистических исследований.

**Список использованных источников**

1. Демченко А. Г. Разработка программного комплекса для автоматического построения размеченных корпусов предложений [Электронный ресурс] / А. Г. Демченко, В. А. Крайванова // XVI Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых «Наука и молодежь – 2019». Секция «Информационные технологии». Подсекция «Программная инженерия». / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2019. – C. 62-63. – Режим доступа: https://www.altstu.ru/media/f/pi2019v1.pdf.
2. Бенгфорт Б., Билбро Р., Охеда Т. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка [Текст] / Б. Бенгфорт, Р. Билбро, Т. Охеда. – СПб.: Питер, 2019. — 368 с.
3. Krayvanova V., Batura T., Bruches E. Semi-automatic method for building corpora with functional topic labelling [Electronic resource] / V. Krayvanova, T. Batura, E. Bruches – Preprint, 2019.
4. The Penn Discourse Treebank [Electronic resource] / E. Miltsakaki, R. Prasad, A. Joshi, B. Webber – 2004. – Access mode: https://www.researchgate.net/publication/2903359\_The\_Penn\_Discourse\_Treebank.
5. Дискурс [Электронный ресурс] // Wikipedia. — 2017. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81% D0%BA%D1%83%D1%80%D1%81.
6. Webber B. Discourse Structure and LanguageTechnology [Text] / B. Webber, M. Egg, V. Kordoni // Journal of Natural Language Engineering / под ред.R. Mitkov. — 2012. — Т. 01. — С. 1—40.
7. Toldova S., Pisarevskaya D., Kobozeva M., Vasilyeva M. The Cues for Rhetorical Relations in Russian: “Cause—Effect” Relation in Russian Rhetorical Structure Treebank Computational Linguistics and Intellectual Technologies [Electronic resource] // Proceedings of the International Conference “Dialogue 2018”. – Moscow, 2018. – Access mode: http://www.dialog-21.ru/media/4338/toldovas.pdf.
8. Макерова С. Р. Грамматика текста грамматика дискурса [Текст] / Историческая и социально-образовательная мысль. — 2013. — № 6 / 2013.
9. Forsbom E. Rhetorical Structure Theory in Natural Language Generation [Electronic resource] // E. Forsbom. – 2005. – Access mode: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2668&rep=rep1&type=pdf.
10. Википедия: Описание [Электронный ресурс] // Википедия. — 2019. — Режим доступа: http://ru.wikipedia.org.
11. Wikimedia Downloads [Электронный ресурс] // Wikimedia Downloads. – 2019. – Режим доступа: https://dumps.wikimedia.org/.
12. Дерево категорий. Служебная страница [Электронный ресурс] // Википедия. – 2019. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE\_%D0%BA%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D0%B9.
13. Lynn C., Marcu D., Okurowski M.E. RST Discourse Treebank LDC2002T07 [Электронный ресурс] // Web Download. Philadelphia: Linguistic Data Consortium, 2002. – Режим доступа: http://www.sovren.com.
14. Liakata, M., Claire Q., Soldatova L. Semantic annotation of papers: interface & enrichment tool (SAPIENT) [Text] / Proceedings of BioNLP-09. – 2009. – P. 193-200.
15. Tholpadi G., Bhattacharyya C., Shevade S. Corpus-Based Translation Induction in Indian Languages Using Auxiliary Language Corpora from Wikipedia [Electronic resource] // ACM Trans. Asian Low-Resour. Lang. Inf. Process. – March 2017. – Access mode: https://doi.org/10.1145/3038295.
16. Python [Электронный ресурс]. – Режим доступа: https://www.python.org, свободный. – Загл. с экрана.
17. Django [Электронный ресурс]. – Режим доступа: https://www.djangoproject.com, свободный. – Загл. с экрана.
18. Bootstrap 4 [Электронный ресурс]. – Режим доступа: https://getbootstrap.com/. – Загл. с экрана.
19. Jinja [Электронный ресурс]. – Режим доступа: http://jinja.pocoo.org/. – Загл. с экрана.
20. Ajax-запрос [Электронный ресурс]. – Режим доступа: http://jquery.page2page.ru/index.php5/Ajax-%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81. – Загл. с экрана.
21. GitHub [Электронный ресурс]. – Режим доступа: https://github.com/. – Загл. с экрана.
22. Kim J.-D., Ohta T., Tateisi Y., Tsujii J., GENIA corpus — a semantically annotated corpus for bio-textmining [Электронный ресурс] // Bioinformatics. Volume 19. Issue suppl\_1. – 3 July 2003. – P. 180–182. – Режим доступа: https://doi.org/10.1093/bioinformatics/btg1023.

1. Онтология в информатике — это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. [↑](#footnote-ref-1)
2. Таксономия — учение о принципах и практике классификации и систематизации сложноорганизованных иерархически соотносящихся сущностей. [↑](#footnote-ref-2)
3. Дамп – XML-размеченное собрание всех статей Википедии в одном файле. [↑](#footnote-ref-3)