ФГБОУ ВО «ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ТЕОРИИ ВЕРОЯТНОСТЕЙ И АНАЛИЗА ДАННЫХ

(подпись соискателя)

Ткач Станислав Сергеевич

Магистерская диссертация

Применение лексических цепочек для разрешения лексической многозначности на основе Русского Викисловаря

Направление 090402 — Информационные системы и технологии

Научный руководитель:
к.т.н., рук.
Лаборатории информационных компьютерных технологий
Института прикладных математических исследований
А. А. Крижановский

_____
(подпись руководителя)

Петрозаводск — 2016

Оглавление

Введение

Актуальность темы

Одной из основных задач обработки текстов является разрешение лексической многозначности. Целью данной задачи является установление значений слов, основанных на контексте, в котором они употребляются. Разрешение лексической многозначности требуется практически во всех языковых областях, в том числе таких как: информационный поиск, машинный перевод, извлечение информации и контент-анализ и др. [7].

Для того, чтобы решить данную задачу, нужно определить все толкования слов и отношения между этими толкованиями и контекстом употребления слов. Основной источник толкований - это различные толковые словари и энциклопедии. Для того, чтобы установить связи между толкованиями, создаются такие структуры, как семантические сети и тезаурусы, но так как создание таких ресурсов является трудоемким процессом, исследователи в области обработки языка заинтересовались возможностью использования для решения данной задачи таких ресурсов, как Веб, онлайн-словари и энциклопедии, которые создаются и постоянно обновляются огромным числом различных пользователей.

Викисловарь – это уникальный, значимый и богатый ресурс для автоматической обработки текста. Данный ресурс популярен в связи с тем, что он постоянно пополняется новыми данными и в нем содержатся толкования слов, описание их фонетических и морфологических свойств, семантические отношения, ко многим словам подобраны иллюстрации [15].

Структура статьи Викисловаря, содержащая значения слова, синонимы, гипонимы, гиперонимы и примеры употребления, позволяет

использовать ее с целю применения метода построения лексических цепочек для разрешения лексической многозначности, который заключается в назначении и использовании семантических связей между различными словами контекста.

Цель работы

Целью работы является разработка алгоритма для разрешения лексической многозначности, использующего метод построения лексических цепочек и Русский Викисловарь в качестве машиночитаемого словаря, также разработка приложения “Nerpa”, реализующего данный алгоритм.

Для достижения цели работы были поставлены следующие задачи:

1. Изучение научной литературы и уже проведенных исследований в данной области;
2. Изучение метода построения лексических цепочек;
3. Разработка алгоритма построения лексических цепочек на основе Русского Викисловаря;
4. Разработка системы “Nerpa”, использующей алгоритм построения лексических цепочек для разрешения лексической многозначности;

Научная новизна и практический вклад

1. Разработан алгоритм построения лексических цепочек на основе данных, полученных из Викисловаря;
2. Разработана система “Nerpa” реализующая следующие функции:

2.1. Строит лексические цепочки для введенного пользователем фрагмента текста, используя разработанный ранее алгоритм;

2.2. Из построенных лексических цепей выбирает наиболее сильную цепочку;

2.3. Извлекает из полученной сильной цепочки толкование слова, введенного пользователем и выводит его на экран;

Практическая значимость

Разработанный алгоритм может использоваться для создания или повышения точности существующих систем, предназначенных для обработки или анализа текстов на естественном языке.

Система “Nerpa” может применяться как самостоятельно для разрешения лексической многозначности, так и стать основой для других систем, предназначенных для обработки и анализа текстовых данных.

Глава 1. Разрешение лексической многозначности

Лексическая многозначность

Лексическая многозначность (полисемия) — это наличие у слова нескольких взаимосвязанных значений, характеризуемых общностью одного или более семантических компонентов [17].

Естественному языку присуща неоднозначность. Например, в английском языке существительное «plant» может означать «зеленое растение» или «завод», аналогично французское слово «feuille» может иметь значение «лист (растения или дерева)» или «лист бумаги». Точное толкование многозначного слова может быть выбрано на основе контекста, в котором оно употребляется и соответственно задача выбора верного значения слова определяется как задача автоматического назначения наиболее подходящего для пользователя толкования данного слова в пределах контекста [9].

Задача WSD

Разрешение лексической многозначности (англ. word sense disambiguation или WSD) — задача определения смысла (значения) слова, которое используется в определенном контексте [7].

Задача разрешения лексической многозначности впервые была сформулирована в качестве отдельной вычислительной задачи в течение первых дней машинного перевода в 1940-х годах, что делает её одной из самых старых проблем компьютерной лингвистики. Уоррен Уивер, в своем знаменитом меморандуме о переводе [12], впервые представил проблему в вычислительном контексте.

В 1970-х WSD была частью семантических систем интерпретации, разработанных в области искусственного интеллекта, но так как системы WSD были в основном основаны на вручную выведенных правилах (rule-based and hand-coded), эти правила полностью зависели от количества имеющихся знаний, получить которые было очень тяжело [7].

К 1980-м стали доступны крупномасштабные лексические ресурсы, такие как Oxford Advanced Learner's Dictionary of Current English (OALD) и выведенные вручную знания были заменены знаниями, которые стали извлекаться автоматически из этих ресурсов [7].

В 1990-х к задаче WSD стали применяться методы изучения с учителем (англ. supervised machine learning techniques [7]).

В 2000-е годы методы обучения с учителем достигли плато в точности, поэтому пришлось сместить внимание в сторону работы с более обобщёнными системами словарных знаний (coarse-grained senses), адаптации к предметным областям (domain adaptation), частичного обучения с учителем (semi-supervised systems) и обучения без учителя (unsupervised corpus-based systems), смешанных методов, а также обработки баз знаний и выведению результатов в виде графов (the return of knowledge-based systems via graph-based methods). Однако, до сегодняшнего дня системы обучения с учителем считаются наиболее эффективными [7].

Пример многозначного слова

В качестве одного из примеров полисемии в русском языке можно привести многозначное русское слово “ключ”. Вот некоторые примеры употребления данного слова:

1. В саду, за малиной, есть калитка, её маменька запирает на замок, а **ключ** прячет. [*А. Н. Островский, «Гроза», 1860 г.*]

2. Много стоило мне усилий, чтобы найти **ключ** к сердцам этих людей. [*М. Е. Салтыков-Щедрин, «Благонамеренные речи»*]

3. Миша быстро надел наушники и, низко наклонившись к фибровому чемоданчику, застучал **ключом**, дробно выбивая точки и тире азбуки Морзе. [*В. П. Катаев, «Катакомбы», 1949 г.*]

4. Юноши, как правило, играют в том же **ключе**, что и их старшие товарищи. [*«Футбол надежды нашей», 1985 г. // «Студенческий меридиан»*]

Исходя из данных примеров, можно увидеть, что в каждом из четырех случаев, слово “ключ” имеет разное значение.

Если фрагмент текста обозначить как $W = w_1, w_2, \ldots, w_n$, где каждое w_i - это многозначное слово, а список возможных толкований слов контекста обозначить как $S = S_{w_1}, S_{w_2}, \ldots, S_{w_n}$, то для каждого w_i из последовательности слов $W = w_1, w_2, \ldots, w_n$ существует последовательность значений $S_{w_i} = S^1_{w_i}, S^2_{w_i}, \ldots, S^n_{w_i}$, где $S_{w_i} \in S$. Задача разрешения лексической многозначности сводится к задаче поиска наиболее вероятного значения $S^j_{w_i} \in S_{w_i}$ для конкретного многозначного слова w_i.

Лексическая цепочка – последовательность слов $G = w_1, \ldots, w_n$, которые имеют схожее значение (толкование). Слово w_i в данной последовательности связано с w_{i+1} лексико-семантическим отношением $\lambda = w_i \leftrightarrow w_{i+1}$ (например, синоним, гипоним и др.). Учитывая контекст W слова w_i, где $w_i \in W$, при данном подходе выбираются те конфигурации значений S_{w_i}, которые максимизируют число последовательных пар связанных семантическими отношениями:

$$\lambda(w_i, w_{i+1}) = \begin{cases} 1, \text{ есть связь} \\ 0, \text{ нет связи} \end{cases}$$

Если обозначить функцию лексических цепей, присоединяющую значение $S^j_{w_i}$ к значениям, выбранным для W как $f(S^j_{w_i}, W)$, а лексическую цепочку как $G_{n-1} = S_{w_1}, S_{w_2}, ..., S_{w_{n-1}}$, то функция добавления нового слова в цепочку будет выглядеть:

$$G_n = f(G_{n-1}, w_n), \text{ где } w_n \leftrightarrow \{S^1_{w_n}, S^2_{w_n}, ..., S^x_{w_n}\} := S^* \qquad (1)$$

Сила цепочки будет иметь следующий вид:

$$F_{y \in S^*}(G_{n-1}, y) = \sum_{x \in G_{n-1}} synonym(x, y) \qquad (2)$$

где:

$$0 \le F(G_n) \le \frac{n(n-1)}{2}$$

и

$$synonym(x, y) = \begin{cases} 1, & x \text{ и } y \text{ синонимы} \\ 0, & \text{не синонимы} \end{cases}$$

Исходя из формул (1) и (2) максимизация будет выглядеть следующим образом:

$$(1), (2) \Rightarrow y^* = arg\ max_{y \in S^*}(F(G_{n-1}, y))$$

Таким образом, возникает проблема выбора одного или нескольких значений слов, которые соответствовали бы тому контексту, в котором употребляется слово. Данная задача является одной из важнейших задач обработки текстов. Для человека процесс устранения многозначности во многом является подсознательным и не представляет особых трудностей, но как вычислительная проблема он представляет собой сложнейшую задачу из области искусственного интеллекта.

Задачи, в которых требуется разрешение лексической многозначности

Машинный перевод – наиболее очевидное приложение, требующее разрешения лексической многозначности. Например английское слово “break” в зависимости от контекста можно перевести как “ломать”, “нарушать”, “ослаблять”, “расторгать” и т. д. Но решение задачи WSD также требуется практически в каждом применении языковых технологий, в том числе таких как:

Информационный поиск: при поиске специфичных ключевых слов, желательно оставлять только документы, в которых эти слова встречаются в нужном смысле, что позволит повысить релевантность получаемых результатов [2];

Контент-анализ: основной подход - это анализ распределения категорий слов в текстовых коллекциях, то есть слов относящихся к определенной теме. Для построения верных распределений категорий необходимо верное установление значений используемых слов [4].

Также решение задачи WSD требуется для обработки речи, текстов, извлечения информации и лексической интерпретации.

Разрешение лексической многозначности становится наиболее важным в новых исследовательских областях, таких как биоинформатика и Семантический Веб (Semantic Web) [7].

Традиционные подходы к решению задачи WSD

Есть четыре традиционных подхода к разрешению лексической многозначности:

- WSD-методы, основанные на знаниях: использование знаний в качестве ресурсов для исследований (словари, тезаурусы, онтологии,

словосочетаний и др.), направленных на определение смыслов слов в контексте [11].

- WSD-методы с учителем (supervised): Как правило, классификатор (эксперт) решает задачу классификации для одного слова. Для того, чтобы «научить» классификатор множеству примеров, в которых целевое слово вручную связано со значением из инвентаря значений эталонного словаря, используется многоразовое обучение [11].
- Полуобучаемые или минимально-контролируемые методы (Semi-supervised): эти методы используют вторичные знания, такие как определения терминов в толкованиях слов или выровненный двуязычный корпус [7].
- WSD-методы без учителя (unsupervised, кластеризация и графы): не требуется выбор значения слов из заранее известных (как это, например, происходит при работе со словарём), а требуется построение кластеров, каждый из которых будет соответствовать значению слова [11].

Заключение по первой главе

В данной главе рассмотрены такие понятия, как лексическая многозначность (полисемия) и разрешение лексической многозначности. Приведено описание задачи разрешения лексической многозначности. Задача автоматического разрешения лексической многозначности существует уже более 60 лет, однако до сих пор полностью не решена.

Приведены четыре традиционных подхода к разрешению лексической многозначности:

- WSD-методы, основанные на знаниях;
- WSD-методы с учителем (supervised);
- Полуобучаемые или минимально-контролируемые методы (Semi-supervised);

- WSD-методы без учителя (unsupervised, кластеризация и графы);

В следующей главе будет рассмотрено понятие лексической связности. Будет описан метод лексических цепочек для разрешения лексической многозначности.

Глава 2. Лексическая связность и лексические цепочки

Лексическая связность

Лексическое единство в тексте появляется в результате возникновения цепей связанных слов, которые способствуют целостности общей темы повествования [3]. Эти лексические цепи строятся из единиц текста, имеющих схожее значение друг с другом, и нахождение структуры текста включает в себя поиск таких единиц текста.

Текст или фрагмент текста – это не просто ряд предложений, каждое из которых повествует о какой-то случайной теме, не связанной с остальными предложениями. Наоборот, каждое предложение или фраза любого текста имеет тенденцию быть «о тех же самых вещах», что и остальные предложения в тексте. То есть у текста есть качество единства. Благодаря этому единству предложения и фразы «склеиваются» и функционируют в тексте как единое целое [3].

Связность достигается через соединения и семантические отношения слов. Связность не является гарантией единства в тексте, она скорее представляет собой устройство для создания единства. Холидей и Хасан в своей работе [6] пишут, что связность – это способ заставить текст "оставаться целым в целом".

Лексическая связность – это связь, которая является результатом семантических отношений между словами [3].

Холлидей и Хасан предоставили классификацию лексической связности, основанную на типе отношений, которые существуют между словами [6]. Есть пять основных классов:

1. Повторение с идентичностью ссылки (reiteration with identity of reference);

Пример 1:

- Мария надкусила *персик;*
- К сожалению, *персик* не созрел;

2. Повторение без идентичности ссылки (reiteration without identity of reference);

Пример 2:

- Мария съела несколько *персиков;*
- Она очень любит *персики;*

3. Связность высших уровней (reiteration by means of superordinate);

Пример 3:

- Мария съела *персик;*
- Она любит *фрукты;*

4. Систематичное семантическое отношение (systematic semantic relation);

Пример 4:

- Мария любит *зеленые* яблоки;
- Ей не нравятся *красные;*

5. Несистематичное семантическое отношение (nonsystematic semantic relation);

Пример 5:

- Мэри провели три часа в *саду* вчера;
- Она *копала* картофель;

Примеры 1, 2, и 3 попадают в класс повторения. Обратите внимание на то, что повторение включает не только идентичность ссылки или

повторение того же самого слова, но также и использование синонимов, гипонимов и гиперонимов. Примеры 4 и 5 попадают в класс словосочетания, то есть, семантических отношений между словами, которые часто употребляются рядом друг с другом в тексте. Класс словосочетаний может быть далее разделен на две категории отношений: систематичный семантический, и несистематичный семантический [3].

Систематичные семантические отношения могут быть классифицированы довольно простым способом. Этот тип отношений включает антонимы, члены упорядоченного множества, такие как {*один, два, три*}, члены неупорядоченного множества, такие как {*белый, черный, красный*}, и части-к-целому, такие как {*глаза, рот, лицо*} [3].

Примером 5 является иллюстрация предложений, где отношение слов *сад* и *копать* несистематично. Этот тип отношений является самым проблематичным, особенно с точки зрения представления знаний. Такие отношения существуют между словами, которые имеют тенденцию употребляться в подобной лексической окружающей среде (similar lexical environments). Слова имеют тенденцию употребляться в подобной лексической окружающей среде, потому что они описывают вещи, которые происходят в аналогичных ситуациях или контекстах в мире. Следовательно, зависящие от контекста примеры, такие как {*почта, служба, почтовые марки, платить*} включены в класс. Другой пример такого типа {*автомобиль, колеса, поворотные*}. Эти слова связаны в контексте, описывающем вождение автомобиля, но если их извлечь из этого контекста, они не будут связаны систематичным отношением [3].

Лексические цепочки

Часто, лексическая связность происходит не просто между парами слов, но последовательно между многими соседними словами,

охватывающими актуальную единицу текста. Эти последовательности связанных слов называются лексическими цепочками [3]. Есть отношение расстояния между каждым словом в цепи и словами, сосуществующими в пределах данного промежутка. Лексические цепочки не останавливаются на границах предложения. Они могут соединять пары как смежных слов, так и находящиеся в разных предложениях, “пересекая” весь текст [3].

Лексические цепи имеют тенденцию разграничивать части текста, у которых есть сильное единство значения [3].

Есть две основных причины, почему лексическая связность важна для систем разрешения лексической многозначности [3]:

1. Лексические цепи обеспечивают легко определяемый контекст, чтобы помочь в разрешении лексической многозначности;
2. Лексические цепи дают возможность для определения последовательности и структуры контекста, и следовательно определения большего значения текста.

Метод построения лексических цепочек

Моррис и Хирст в своей работе [3] представили первую вычислительную модель для лексических цепочек. Цепочки создавались путем взятия нового слова из текста и поиска родственной (связанной) цепочки для слова в соответствии с критериями родства. Недостатком подхода было то, что в одну цепочку могло входить слово с разными значениями (для многозначных слов). Таким образом, выбор подходящей цепочки для слова эквивалентен решению WSD задачи. Метод построения лексических цепочек включает шаги [18]:

1. Выбирается набор слов-кандидатов (существительные и составные существительные). Это кандидаты на включение в цепочки;

2. Строится список всех значений для каждого слова-кандидата (по данным словаря);

3. Для каждого значения каждого слова-кандидата находится связь для каждого слова во всех уже построенных цепочках (слово в цепочке имеет строго определённое значение, задаваемые другими словами в той же цепочке) [1];

4. Слово-кандидат добавляется в цепочки со словами, в которых найдена связь. Смысловая неоднозначность устраняется, то есть в цепочку добавляется не просто слово, а его конкретное значение;

Для иллюстрации метода приводится пример на отрывке текста, который представлен ниже и проверяется, какое значение будет выбрано для слова *machine*. Во-первых, для слова *Mr.* создается узел [лексема «*Mr.*», значение {*mister*, *Mr.*}]. Следующим по тексту существительным, представленным в тезаурусе WordNet, будет слово *person*, у него есть два значения: [лексема «*person1*», значение {*human being*}] и [лексема «*person2*», значение {*grammatical category of pronouns and verb forms*}]. Наличие двух значений у слова *person* разбивает пространство цепочек на два множества интерпретаций: в первой интерпретации используется значение *person1*, во второй – *person2* (рис. 1) [18].

***Mr**. Kenny is the **person** that invented an anesthetic **machine** which uses **micro-computers** to control the rate at which an anesthetic is pumped into the blood. Such **machines** are nothing new. But his **device** uses two **micro-computers** to achieve much closer monitoring of the **pump** feeding the anesthetic into the patient.*

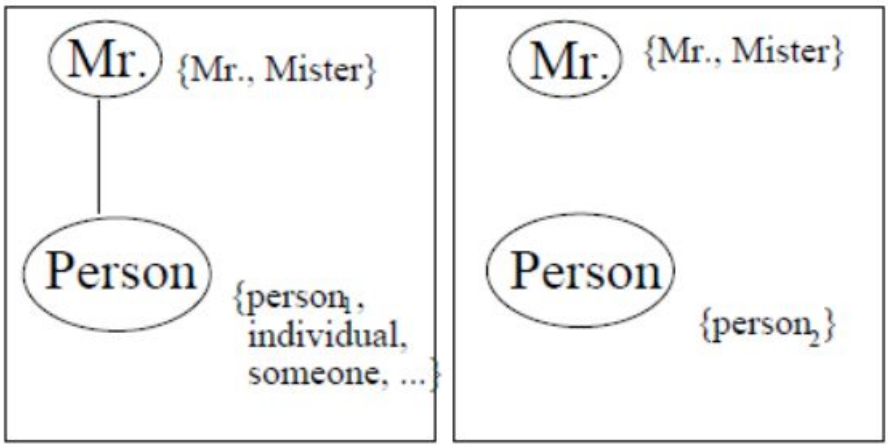

Рисунок 1. Шаг 1, интерпретация 1 (слева) и 2 (справа) [8]

Компонентой – это список взаимоисключающих интерпретаций [8]. Именно посредством компонент выбор одного из значений слов ведет к выбору соответствующей интерпретации, а следовательно, к невозможности других интерпретаций из этой компоненты. Интерпретации 1 и 2 на рис. 1 являются компонентой [18].

Следующее слово *anesthetic* не связано со словами из первой компоненты, поэтому для него создается компонента с одним значением (новая компонента содержит ровно одну интерпретацию) [18].

Следующее слово *machine* имеет 5 значений: от *machine1* до *machine5*. В первом значении *machine1* [лексема «*machine*», значение {*an efficient person*}] слово связано со значениями слов *person* и *Mr.*, поэтому слово *machine* вставляется в первую компоненту. После этой вставки изображение первой компоненты становится таким, как показано на рис. 2. Если продолжить этот процесс и добавить слова *micro-computer*, *device* и *pump*, то количество альтернативных вариантов значительно увеличивается. Самые сильные интерпретации представлены на рис. 3. При условии, что текст связный, лучшей интерпретацией считается та, которая имеет больше всего связей. В данном случае в конце шага 3 выбрана другая интерпретация *machine4* [лексема «*machine*», значение

{*any mechanical or electrical device that performs or assists in the performance*}], что верно отражает значение слова *machine* в этом контексте [18].

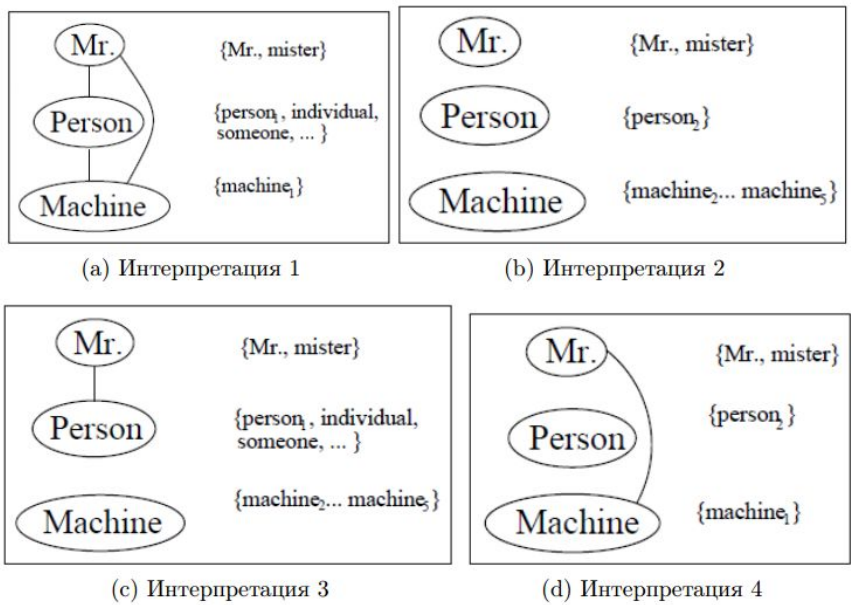

Рисунок 2. Четыре интерпретации на втором шаге [8]

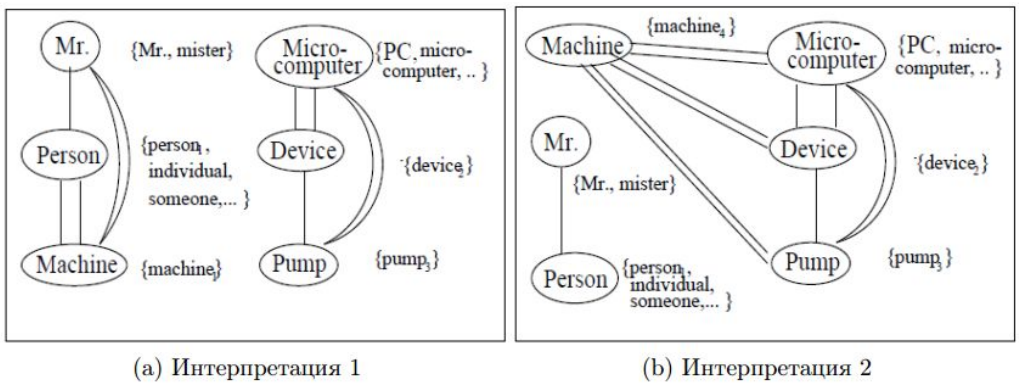

Рисунок 3. Две самые сильные интерпретации, полученные на третьем шаге [8]

Оценка интерпретации определяется как сумма оценок ее цепочек. Оценка цепочки определяется количеством и весом отношений между участниками цепочки. В эксперименте авторы зафиксировали следующий вес: повторения и синонимы – 10, антонимы – 7, гиперонимы и гипонимы

– 4. Описанный алгоритм вычисляет все возможные интерпретации, не допуская противоречий между ними. Когда число возможных интерпретаций превышает определенный порог, слабые интерпретации удаляются, это необходимо для предотвращения экспоненциального роста использования памяти [18].

Заключение по второй главе

В данной главе рассмотрено понятие лексической связности и перечислены 5 основных классов связности:

1. Повторение с идентичностью ссылки (reiteration with identity of reference);
2. Повторение без идентичности ссылки (reiteration without identity of reference);
3. Связность высших уровней (reiteration by means of superordinate);
4. Систематичное семантическое отношение (systematic semantic relation);
5. Несистематичное семантическое отношение (nonsystematic semantic relation);

Приведено понятие лексической цепочки и описан метод построения лексических цепочек. Рассмотрен алгоритм построения цепочки, который включает следующие шаги:

1. Выбирается набор слов-кандидатов (существительные и составные существительные). Это кандидаты на включение в цепочки;
2. Строится список всех значений для каждого слова-кандидата (по данным словаря);
3. Для каждого значения каждого слова-кандидата находится связь для каждого слова во всех уже построенных цепочках (слово в

цепочке имеет строго определённое значение, задаваемые другими словами в той же цепочке);

4. Слово-кандидат добавляется в цепочки со словами, в которых найдена связь. Смысловая неоднозначность устраняется, то есть в цепочку добавляется не просто слово, а его конкретное значение;

В следующей главе будет приведен пример того, как данные Викисловаря можно использовать для построения лексических цепочек, будет описан алгоритм построения лексических цепочек для разрешения лексической многозначности на основе Русского Викисловаря.

Глава 3. Применение метода построения лексических цепочек к решению задачи WSD на основе Русского Викисловаря

Выбор словаря

Для построения лексических цепочек необходимо использование различных толкований слов. Основной источник толкований - это толковые словари и энциклопедии. Для того, чтобы установить связи между толкованиями, создаются такие структуры, как семантические сети и тезаурусы, но так как создание таких ресурсов является трудоемким процессом, исследователи в области обработки языка заинтересовались возможностью использования для решения данной задачи таких ресурсов, как Веб, онлайн-словари и энциклопедии, которые создаются и постоянно обновляются огромным числом различных пользователей.

Текущая ситуация в современной российской лексикографии отражает переходный период от традиционных печатных изданий до крупномасштабных проектов, основанных на больших корпусах и краудсорсинге. Традиционные словари, основанные на ручной выборке и обработке данных, рассматриваются как высококачественные источники, но все же они уступают таким ресурсам как Викисловарь, в объеме и охвате современной лексики [13].

Викисловарь – это уникальный, значимый и богатый ресурс для автоматической обработки текста. Данный ресурс имеет популярность в связи с тем, что он содержит толкования слов, описание их фонетических и морфологических свойств, семантические отношения, а также постоянно пополняется новыми данными [15].

Русский Викисловарь

Викисловарь – это свободно пополняемый многофункциональный многоязычный словарь и тезаурус [15].

В Викисловаре содержатся толкования слов, описание их фонетических и морфологических свойств, семантические отношения, ко многим словам подобраны иллюстрации. Кроме того Викисловарь постоянно пополняется новыми данными.

Есть четыре типа статей в Викисловаре:

- об отдельных словах (лексемах);
- о словообразовательных единицах (морфемах и частях сложных слов);
- о словосочетаниях;
- об аббревиатурах;

Объектом описания в словарной статье выступает лемматизированная, или основная форма слова. В русском и ряде других языков такими формами являются:

- Для существительных — форма именительного падежа единственного числа (например, человек, а не человека, человеком и т. п.);
- Для прилагательных, порядковых числительных и т. п. — форма именительного падежа единственного числа мужского рода (например, хороший, а не хорошая, хорошей, хорошими и т. п.). Исключения могут составлять лексикализованные субстантивированные формы женского или среднего рода типа служащая и т. п.;
- Для глаголов — неопределённая форма (например, ходить, а не хожу, ходим, ходил и т. п.);

Для каждой статьи существует категория, указывающая на тип языковой единицы (для словосочетаний) или часть речи (для слов, лексем). Как правило, в конце статьи указана семантическая категория, которая обозначает предметную область или часть картины мира, к которой описываемая языковая единица относится по смыслу. Например, в статье *“гитара”* возможным вариантом будет *“музыкальные инструменты”*.

Различные толкования слова в Викисловаре часто сопровождаются примерами использования, подтверждающими собой соответствующее толкование, также чаще всего толкования упорядочены:

- по частотности употребления: самые частотные, привычные, обыденные значения приводятся в начале списка; редкие, специальные, жаргонные — в конце;
- по этимологически-хронологическому принципу («первым появился, первым упомянут»): что первично, то и впереди, а переносные и производные значения — после;

Каждое значение сопровождается иллюстрирующим примером (примерами) словоупотребления — одной или несколькими цитатами, в которых описываемое слово употреблено в данном значении.

В секции «Семантические свойства», помимо подраздела «Значение», по умолчанию содержатся четыре подраздела, соответствующие наиболее важным семантическим связям определяемого слова, а именно его синонимам, антонимам, гиперонимам и гипонимам. Кроме того, для отдельных слов в данную секцию включаются и другие типы семантических отношений (в случае их релевантности), например, меронимы, холонимы, конверсивы и т. п.

Статья о слове русского языка содержит следующие разделы:

- морфологические и синтаксические свойства слова (например: «Существительное, женский род, неодушевлённое, 1-е склонение»);
- фонетические свойства (транскрипция в системе IPA/МФА, например ['palkə]; звуковые примеры произношения);
- семантические свойства, включая, в общем случае:
  - значение (список возможных толкований);
  - синонимы;
  - антонимы;
  - гиперонимы;
  - гипонимы;
  - согипонимы;
  - меронимы;
  - холонимы;
- родственные (однокоренные) слова, желательно по частям речи — существительные (палочка), прилагательные (палочный);
- этимология слова;
- устойчивые сочетания и фразеологизмы с участием данного слова (например, «палка о двух концах»);
- переводы слова на иностранные языки;
- примечания, литература;

Таким образом, извлекая из словарной статьи Викисловаря нужную информацию, можно применить алгоритмы для решения задачи WSD. Структура статьи Викисловаря, содержащая значения слова, синонимы, гипонимы, гиперонимы и примеры употребления, позволяет использовать ее с целю применения метода построения лексических цепочек для разрешения лексической многозначности.

Алгоритм построения лексических цепочек на основе Русского Викисловаря

Лексическая цепочка – это последовательность слов w_1, ..., w_n, которые имеют схожее значение (толкование) [10]. Слова связаны друг с другом лексико-семантическим отношением (например, синоним, гипоним, и т.д.).

Данный метод подразумевает, что у отрывков из разговорного или письменного текста есть свойство единства. Синтаксические и лексические средства могут использоваться, чтобы создать ощущение связности между предложениями, явление, известное как текстовая связность [6]. Из всех средств связи лексическая связность, вероятно, наиболее поддающаяся автоматической идентификации. Лексическая связность возникает, когда слова связаны семантически, например в отношениях повторения между термином и синонимом. Формирование лексической цепочки – это процесс соединения семантически связанных слов [5].

В статье [8] с целью реферирования текста строится модель в виде лексических цепочек. Реферирование включает четыре этапа: оригинальный текст делится на блоки (сегменты), строятся лексические цепочки, определяются сильные цепочки, извлекаются важные предложения.

Суть метода заключается в объединении разных частей текста в одно целое, в то, что имеет общее значение (смысл).

В данной статье объединяются различные слова в тексте с целью нахождения общего значения между ними. Таким образом, происходит избавление от лексической многозначности.

В статье [6] описывается два способа формирования лексической связности:

- Лексическая связность повторений (reiteration category) достигается повтором слов, использованием синонимов и гипонимов;
- Лексическая связность словосочетаний (collocation category) определена для слов, которые часто употребляются вместе, то есть встречаются в одних и тех же контекстах;

Слова и фразы, между которыми существует лексическая связность, представляют собой лексическую цепочку (lexical chains). Метод лексических цепочек основан на анализе совместной встречаемости слов и лексических связей между словами.

Достоинство лексических цепочек состоит в том, что их не сложно распознать и построить.

Метод построения лексических цепочек имеет следующий алгоритм:

```
Data: фрагмент текста
Result: лексическая цепочка
while not Eof(Фрагмент) do
    выбор слова-кандидата
    Построение списока всех значений для слова-кандидата
end
while not Eof(список значений слов-кандидатов) do
    поиск связей со словами в уже построенных цепочках
    if найдена связь then
        добавление слова-кандидата в цепочку
    end
end
```

Рисунок 4. Алгоритм построения лексических цепей

1. Выбирается набор слов-кандидатов (существительные и составные существительные). Это кандидаты на включение в цепочки;

2. Строится список всех значений для каждого слова-кандидата (по данным словаря);

3. Для каждого значения каждого слова-кандидата находится связь для каждого слова во всех уже построенных цепочках (слово в цепочке имеет строго определённое значение, задаваемые другими словами в той же цепочке);

4. Слово-кандидат добавляется в цепочки со словами, в которых найдена связь. Смысловая неоднозначность устраняется, то есть в цепочку добавляется не просто слово, а его конкретное значение;

Для иллюстрации метода приведем пример на отрывке текста, представленного ниже, и определим, какие значения будут выбраны для слов «любовь» и «дом».

Любовь к **Родине** – одно из самых мощных, возвышенных **чувств**. Она в полной **мере** проявилась в братской **поддержке жителей** Крыма и Севастополя, когда они твердо решили вернуться в свой родной **дом** [16].

Первым существительным в тексте является слово «любовь», исходя из данных Русского Викисловаря, у него есть семь различных значений:

1. чувство глубокой привязанности к кому-либо, чему-либо; Материнская любовь; Любовь к другу; цит. Люблю отчизну я, но странною любовью! Не победит её рассудок мой. [М. Ю. Лермонтов, «Родина», 1841 г.];
2. чувство расположения, симпатии к кому-либо;
3. чувство горячей сердечной склонности, влечение к другому человеку;
4. чья-то о человеке, внушающем чувство любви (в предыдущем значении);
5. любовные отношения;
6. внутреннее стремление, влечение, склонность, тяготение к чему-либо;

7. пристрастие к чему-либо, предпочтение чего-либо;

Наличие нескольких значений разбивает пространство цепочек на несколько множеств интерпретаций, в каждой из которых используются разные значения слова *«любовь»*. Четыре первых значения слова *«любовь»* связаны со словом *«чувство»* и только в первом значении *«любовь»* (цит. Люблю отчизну я, но странною любовью! Не победит её рассудок мой. [М. Ю. Лермонтов, «Родина», 1841 г.]) связана со словом *«Родина»*, отсюда получаем две интерпретации (Рис.1).

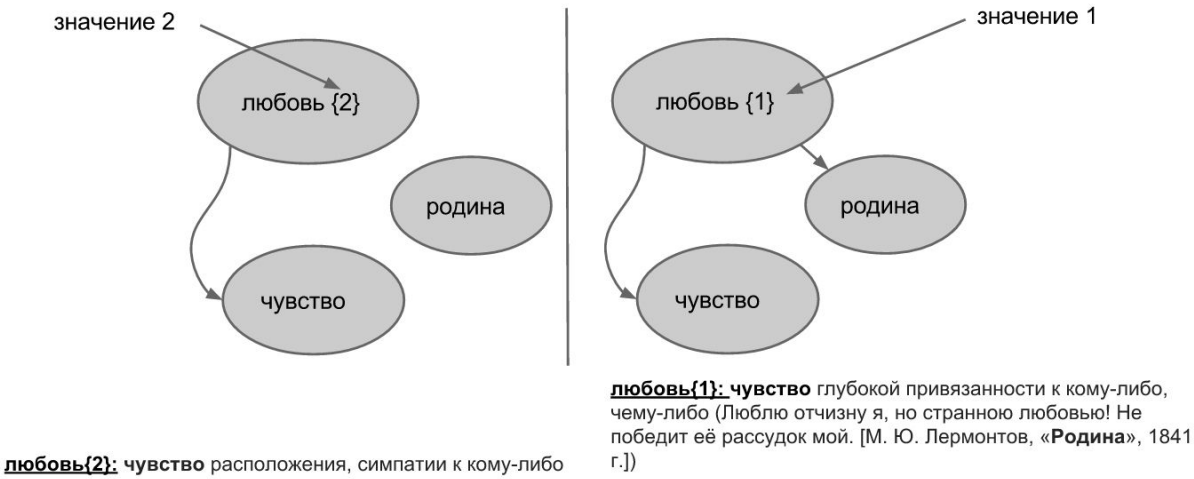

Рис. 1. Шаг 1, интерпретация 1 и 2

Компонентой в работе [8] называют список взаимоисключающих интерпретаций. Именно посредством компонент выбор одного из значений слов ведёт к выбору соответствующей интерпретации, а, следовательно, к невозможности других интерпретаций из этой компоненты. Интерпретации 1 и 2 (Рис. 1) являются компонентой. Следующее слово *«мера»* не связано со словами из первой компоненты, поэтому для него создается компонента с одним значением (то есть новая компонента

содержит ровно одну интерпретацию). Следующее слово *«поддержка»* также не связано со словами из первой компоненты, поэтому для него создается новая компонента с одним значением. Слово *«житель»* имеет единственное значение в Викисловаре:

1. представитель населения; тот, кто живёт где-либо, в чём-либо;

Несмотря на единственное значение у слова *«житель»*, есть еще и единственный гипоним *«гражданин»*, который имеет значения (Wiktionary):

1. лицо мужского рода, принадлежащее к населению какого-либо государства, пользующееся всеми правами и исполняющее все обязанности, установленные законами государства;
2. человек, служащий родине, народу, обществу, заботящийся об общественном благе;
3. официальное обращение к мужчине;

Исходя из значений слова *«гражданин»*, можно сделать вывод, что слово *«житель»* связано со словом *«родина»* через гипоним *«гражданин»*, так как во втором толковании гипонима встречается слово *«родина»*.

Таким образом, получается вторая компонента (Рис. 2). Если продолжить этот процесс и добавить слово *«дом»*, имеющее семь значений, то количество альтернативных вариантов значительно увеличивается. Во втором толковании слова *«дом»* есть слово *«место»*, которое можно связать со словом *«родина»*, так как в единственном толковании *«родины»* есть слово *«место»*. Также во втором толковании слова *«дом»* есть слово *«проживать»*, если мы посмотрим значения этого слова в Викисловаре, то увидим, что первое и второе толкования содержат слово «жить», которое можно связать со словом *«житель»*. Таким образом получается третья компонента (Рис. 3).

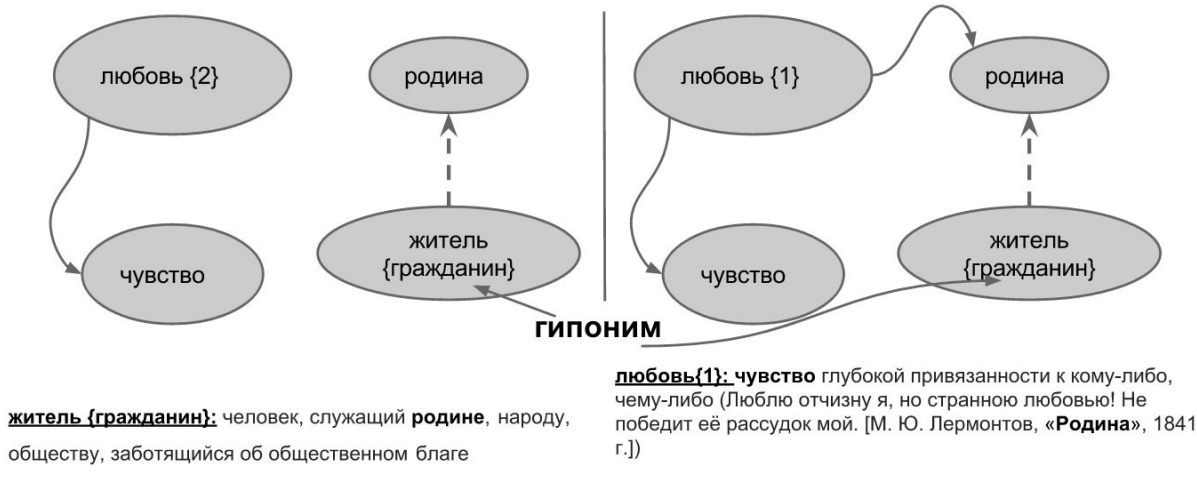

Рис. 2. Шаг 2, интерпретация 1 - 2

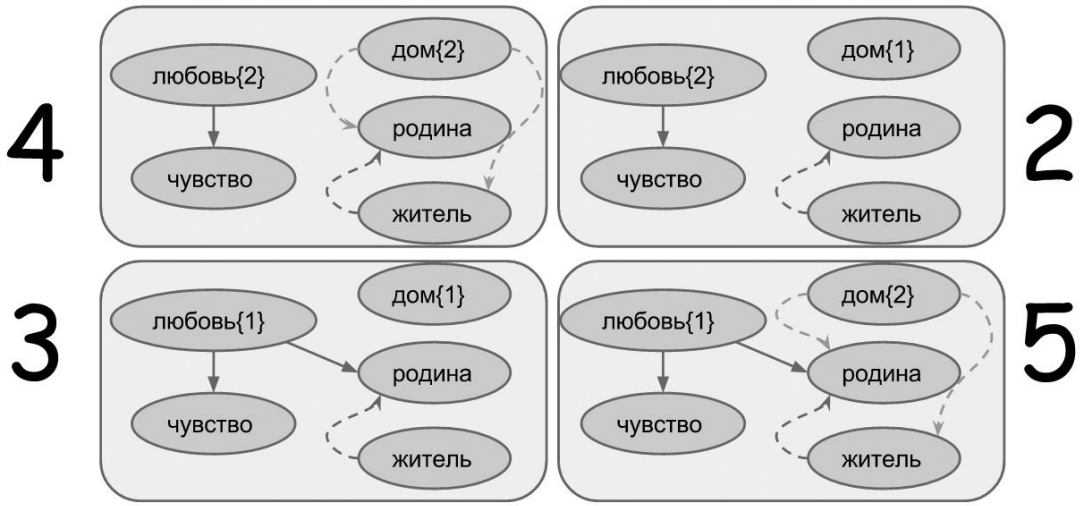

Рис. 3. Шаг 3, интерпретации 1 – 4 (Колличество связей на рисунке уазано большими цифрами 4,2,3,5)

Самые сильные интерпретации представлены на рисунке. При условии, что текст связный, лучшей интерпретацией считается та, которая

имеет больше всего связей (Рис. 4). В данном случае в конце шага 3 выбраны следующие интерпретации интересующих нас слов:

- любовь [лексема «любовь», значение: чувство глубокой привязанности к кому-либо, чему-либо];
- дом [лексема «дом», значение: место, где кто-либо постоянно проживает];

Полученый результат верно отражает значения слов в рассматриваемом контексте.

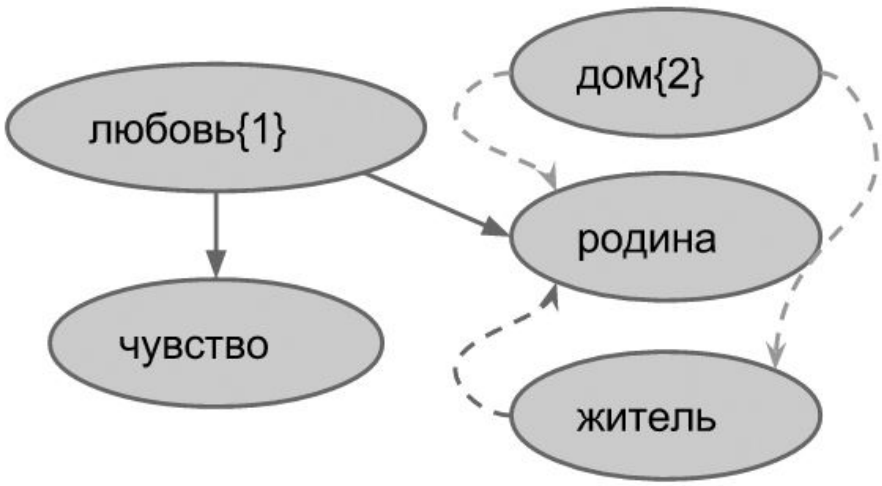

Рис. 4. Окончательная лексическая цепочка, полученная путем выбора самой сильной интерпретации

Заключение по третьей главе

В данной главе рассмотрен метод построения лексических цепочек для решения задачи WSD. Разработан алгоритм построения лексической цепочки с помощью Русского Викисловаря. Было показано как словарные статьи Викисловаря могут использоваться в процессе построения лексических цепей.

В следующей главе будет приведено описание автоматизации алгоритма построения лексических цепочек.

Глава 4. Разработка системы “Nerpa”, реализующей алгоритм построения лексических цепочек

Назначение системы

Система «Nerpa» представляет собой веб-приложение, которое использует метод построения лексических цепочек с целью разрешения лексической многозначности. В качестве машиночитаемого словаря система использует Русский Викисловарь. Суть программы состоит в том, что пользователь вводит в форму целевое слово и фрагмент текста, в котором данное слово употребляется, затем система строит лексическую цепочку из слов введенного текста и исходя из данной цепочки предоставляет пользователю значение слова, соответствующее контексту.

Архитектура системы

Система включает клиентскую и серверную части, для разработки клиентской использовались такие технологии, как HTML5 и CSS, для серверной – Apache 2.4. Основная программная часть системы написана на языке PHP 5, а для управления базой данных использовался продукт MySQL 5.6.

Пользователь вводит в форму целевое слово и фрагмент текста, в котором данное слово употребляется (target word and text), затем фрагмент текста передается в блок lemmatized, где текст разбивается на массив из нормализованных (лемматизированных) слов (word_arr), после этого полученный массив попадает в блок meanings of words, где происходит сопоставление каждого слова со словами в базе данных Викисловаря и получение списков всех значений каждого слова (arr_me). Далее

пользователю выводится список всех значений целевого слова (meanings of target word), а список всех значений слов контекста передается в блок creating chains, где происходит построение различных интерпретаций лексической цепочки. Полученный массив интерпретаций (arr_ch) передается в блок selection of strong chain, где происходит выбор самой сильной интерпретации цепочки и извлечение из данной цепочки значения целевого слова (chain and meaning word), которое затем выводится пользователю. Диаграмма компонетов системы представлена на рисунке 5.

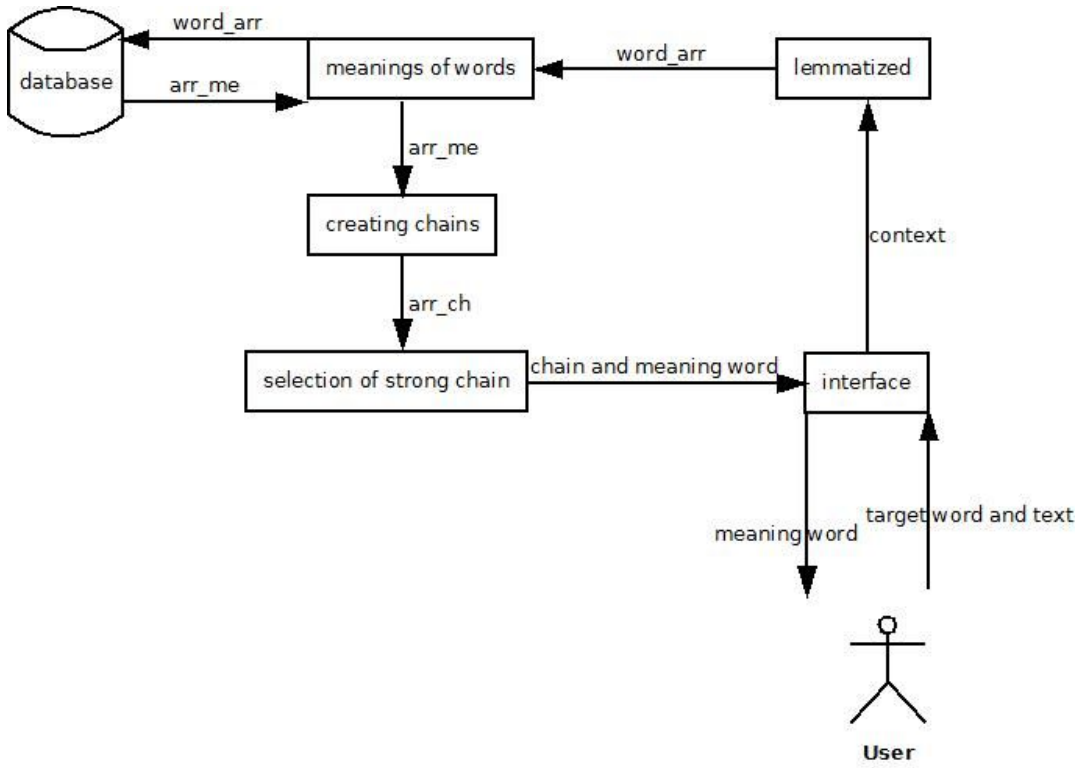

Рис. 5. Диаграмма компонентов системы Nerpa

Подробное описание алгоритма используемого системой для построения лексических цепочек приведено на рисунке 6:

```
Data: целевое слово (target_word), фрагмент текста (context)
Result: толкование целевого слова (sense), лексическая цепочка (lexical_chain)
Step 1:
word_arr = DivideText(context) // получение массива из нормализованных слов контекста
foreach word_arr as words do
    1. Создаем массив mean, состоящий из объектов класса Word
    2. Для каждого нового объекта устанавливаем свойства name (название словарной статьи),
       me (массив всех толкований данного слова из Викисловаря), arr_meaning (массив из
       объектов класса Meaning, содержит значения слов в нормализованном виде)
end
Step 2:
foreach word_arr as words do
    foreach mean as word_obj do
        meanings_array_with_lemma = word_obj->getMeaningsArrayWithLemma(words) //
        получение массива значений, в которых встречается слово words
        foreach meanings_array_with_lemma as arr_w_le do
            создание массива wr, состощего из объектов класса Word, содержащих
            словарную статью с толкованием, имеющим связь с другой словарной статьей
        end
    end
end
Step 3:
foreach wr as arr_wr do
    foreach chains_array as chains do
        if слово принадлежит цепочке then
            добавление в цепочку
        end
        else
            создание новой цепочки
        end
    end
end
```

Рис. 6. Алгоритм построения лексических цепочек на основе Викисловаря

Описание взаимодействия классов системы

Система “Nerpa” разработана с применением объектно-ориентированного подхода в программировании. Структура взаимодействия классов показана на рисунке 7.

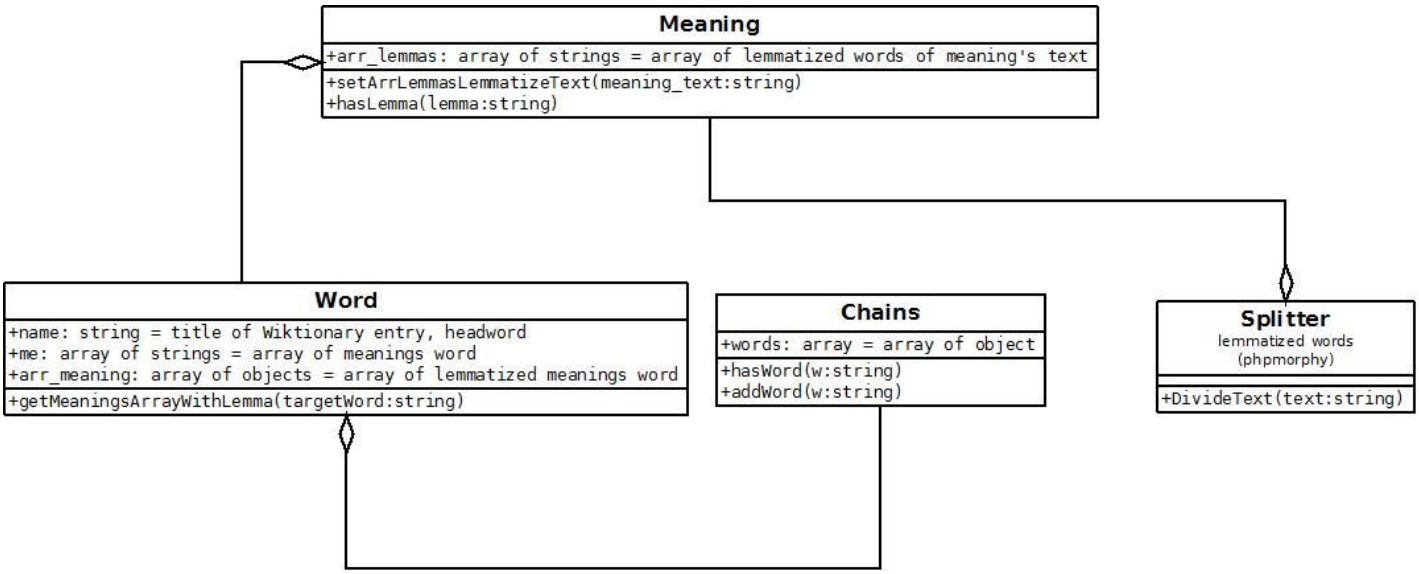

Рис. 7. Структура классов выполняющих анализ контекста и построение лексической цепочки. Классы связаны между собой отношением агрегация

Класс Splitter

Данный класс содержит в качестве метода функцию DivideText, которая в качестве входного параметра получает текстовую строку, а возвращает массив из отдельных слов, входящих в строку. Слова возвращаемого массива приведены в нормальную форму (лемматизированы). Так-же в результирующий массив не попадают стоп слова и знаки препинания. Лемматизация происходит с помощью функции lemmatize, которая относится к библиотеке морфологического анализа phpMorphy.

Метод DivideText класса Splitter используется для приведения к нормальной форме слов вводимого пользователем контекста и слов, получаемых из списков значений Викисловаря.

Класс Meaning

Данный класс содержит в качестве свойства arr_lemmas массив, нормализованных слов, полученный из толкования слова, содержащегося в

классе Word. Также класс Meaning содержит метод setArrLemmasLemmatizeText(meaning_text), который устанавливает свойству arr_lemmas значение массива нормализованных слов, полученного из строки meaning_text. Метод hasLemma(lemma), который проверяет наличие слова lemma в массиве arr_lemmas.

Класс Word

Данный класс содержит свойство name, которое содержит название словарной статьи, свойство me, содержащее массив из всех толкований для данного слова и свойство arr_meaning, которое содержит массив объектов класса Meaning, в которых свойство arr_lemmas принимает значения массивов нормализованных слов для каждого толкования данной словарной статьи. Также данный класс имеет метод getMeaningsArrayWithLemma(lemma), который возвращает массив толкований, в которых употребляется слово lemma.

Класс Chains

Данный класс имеет свойство words, которое содержит массив из слов, входящих в цепочку, метод hasWord(w), который проверяет принадлежность слова w, данной цепочке и метод addWord(w), добавляющий новое слово в лексическую цепочку.

Интеграции между классами

Интеграция между модулями Splitter и Meaning осуществлена с помощью метода setArrLemmasLemmatizeText класса Meaning. Метод setArrLemmasLemmatizeText устанавливает объекту Meaning свойство $arr_lemmas, которое получается путем взаимодействия с методом DivideText класса Splitter.

Интеграция между классами Meaning и Word осуществлена с помощью метода getMeaningsArrayWithLemma класса Word, который ищет

употребление слова lemma в значениях других слов контекста. Данный метод в свою очередь использует метод hasLemma класса Meaning, который проверяет, совпадает ли целевое слово с каким либо из слов в конкретном значении.

Интеграция между модулями Word и Chains осуществлена с помощью метода hasWord класса Chains. Метод hasWord проверяет принадлежность словарного объекта конкретной лексической цепочке путем сравнения имени слова в объекте со словами, используемыми в значениях слов цепочки.

Пользовательский интерфейс системы

Интерфейс системы представлен на рисунке 8:

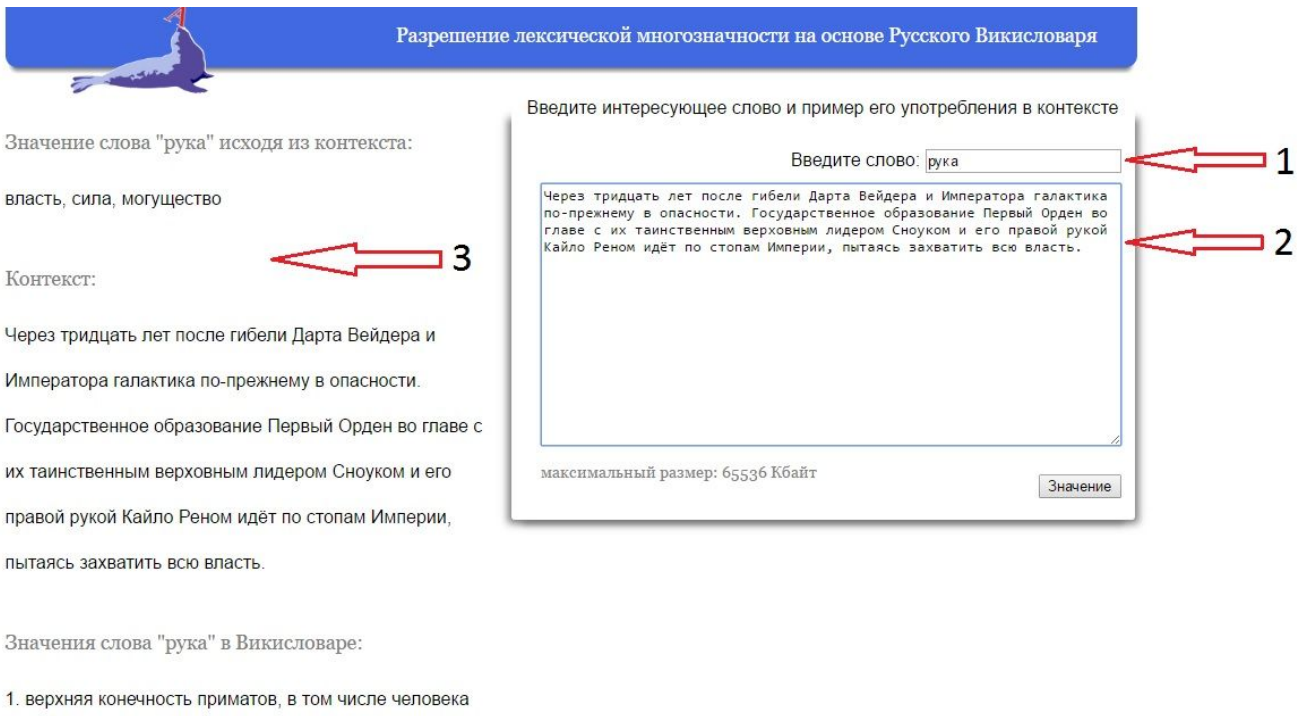

Рис. 8. Интерфейс системы “Nerpa”

Пользовательский интерфейс включает в себя:

1. Текстовое поле для ввода целевого слова;
2. Текстовое поле для ввода контекста;
3. Поле для вывода результатов:

a. Значение целевого слова исходя из контекста;
b. Контекст;
c. Значения целевого слова в Викисловаре;
d. Слова, попавшие в результирующую лексическую цепочку;

Тестирование системы

Во время разработки для проведения тестирования использовалась библиотека PHPUnit.

PHPUnit – это специальный фреймворк, предназначенный для модульного тестирования скриптов языка PHP, разработанный Себастьяном Бергманом. Преимущества PHPUnit:

- Инструменты для создания модульных тестов и организации их в иерархические наборы;
- Интерфейс командной строки для выполнения тестов;
- Провайдеры данных – генераторы наборов данных, для тестирования элементов скрипта, используя различные входные параметры;
- Поддержка тестирования кода, работающего с базой данных.
- Тестирование исключений;
- Поддержка так называемых фиктивных объектов;
- Генераторы отчетов;

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. В противном случае производится заключение о найденной ошибке. В связи со стратегией тестирования, тестирование сценариев более высокого уровня иерархии должно быть приостановлено при нахождении критических ошибок в вызываемых ими сценариях. В случаях возникновения существенных ошибок тестирование может быть продолжено на усмотрение тестировщика. Работы по

тестированию возобновляются после исправления ошибок, вызвавших приостановку тестирования.

Блочное тестирование

Блочное (модульное) - это тестирование методов какого-то класса программы в изоляции от остальной программы.

Объектами для блочного тестирования выбраны: функции divideText, hasWord, addWord, getMeaningsArrayWithLemma.

Было разработано 16 тестов для проверки работоспособности функций системы. Тесты показали, что каждая из функций работает верно.

Интеграционное тестирование

Интеграционное тестирование - это тестирование взаимодействия нескольких классов, выполняющих вместе какую-то работу.

Для проведения интеграционного тестирования применяется стратегия восходящего тестирования.

Порядок интеграции:

1. Splitter + Meaning;
2. Meaning + Word;
3. Word + Chains;

Для проверки интеграций было разработано 13 тестов.

Аттестационное тестирование

Аттестационное тестирование - это тестирование программы в целом. Проводится для всей системы, что подразумевает выполнение действий в пользовательском интерфейсе. Для проведения тестирования в данном случае необходим браузер.

С целью аттестационного тестирования было разработано несколько тестов, которые подробно рассмотрены в пятой главе.

Нагрузочное тестирование

Так как точность решения задачи зависит от количества слов контекста, из которых строится лексическая цепочка, то главным параметром для оценки работы программы было выбрано время, которое требуется для построения лексической цепочки.

В процессе тестирования выполнялась следующая последовательность действий:

1. запуск программы;
2. задание критических значений;
3. оценка времени работы программы;
4. завершение программы;

Результаты тестирования приведены в таблице 1:

Таблица 1. Результаты нагрузочного тестирования

Количество слов	Время обработки
3	< 5 сек
5	< 5 сек
10	< 10 сек
15	< 10 сек
20	< 10 сек
30	< 25 сек
35	<30 сек
40	<30 сек
50	<40 сек

Исходя из полученных результатов данного тестирования, можно сделать вывод, что для оптимальной работы программы необходимо не более 40 - 50 слов контекста, в противном случает время обработки запроса сильно увеличивается.

Покрытие кода тестами

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$$T_{cov} = \frac{L_{tc}}{L_{code}} \times 100\%$$

Где:

T_{cov} - тестовое покрытие;

L_{tc} - количество строк кода, покрытых тестами;

L_{code} – общее количество строк кода;

Тогда:

$$T_{cov} = \frac{L_{tc}}{L_{code}} \times 100\% = \frac{193}{250} \times 100\% = 77.2\%$$

Результаты тестирования показали, что система работает в соответствии с назначением.

Заключение по четвертой главе

В данной главе было рассмотрено:

- Назначение системы “Nerpa”
- Архитектура системы
- Введена схема интеграций системы
- Описание основных классов
- Пользовательский интерфейс
- Подходы и методы для тестирования системы

В следующей главе будут проведены эксперименты, которые более подробно опишут работу разработанной системы.

Глава 5. Эксперименты

Человеческие суждения

Если обратиться к толковым словарям, то выяснится, что у слова “язык” имеется минимум пятнадцать различных толкований (табл. 2).

Таблица 2. Токования слова “язык”

№	Словарь С. И. Ожегова	Большой академический словарь	Викисловарь
1		Орган в полости рта в виде мышечного выроста у позвоночных животных и человека, способствующий пережёвыванию и глотанию пищи, определяющий её вкусовые свойства	Подвижный мускулистый орган в ротовой полости позвоночных животных и человека, служащий для определения вкуса,захватывания, пережёвывания и глотания пищи, а у человека также для артикуляции речи
2		Этот орган человека, участвующий в образовании звуков	

		речи и тем самым в словесном воспроизведении мыслей; орган речи	
3	Исторически сложившаяся система звуковых словарных и грамматических средств, объективирующая работу мышления и являющаяся орудием общения, обмена мыслями и взаимного понимания людей в обществе	Система словесного выражения мыслей, обладающая определённым звуковым и грамматическим строем и служащая средством общения людей.	Исторически сложившаяся система словесного выражения мыслей, обладающая определенным звуковым, лексическим и грамматическим строем, используемая как средство общения и передачи информации в человеческом обществе
4	Совокупность средств выражения в словесном творчестве, основанных на	Разновидность речи, обладающая теми или иными характерными признаками; стиль	Разновидность речи, которой присущи те или иные характерные признаки

	общенародной звуковой, словарной и грамматической системе, стиль		
5	Система знаков (звуков, сигналов), передающих информацию	Система знаков (звуков, сигналов), передающих информацию	
6		Народ, народность	Народ, этнос
7	Пленный, захваченный для получения нужных сведений	Пленный, от которого можно узнать нужные сведения	Пленный, захваченный с целью получения информации, сведений о противнике
8		Металлический стержень в колоколе или колокольчике, который, ударяясь о стенку, производит звон	Подвесная деталь колокола (звонка, колокольчика и т. п.) как правило в виде металлического стержня, которая, ударяясь о стенку, производит звук

9		О том, что имеет удлинённую, вытянутую форму	Объект или предмет, имеющий удлинённую, сужающуюся к концу форму
10			Блюдо, приготовленное из языка [1] животного (преимущественно говяжьего или свиного)
11	Речь, способность говорить		Способность человека говорить, выражать свои мысли
12			Стиль речи, присущий кому-либо, чему-либо
13			Способ, манера выражения, свойственные кому-либо

14	То, что выражает, объясняет собой что-н. (о предметах и явлениях)		То, что выражает собою что-либо, может быть средством общения
15			Отдельное завихрение огня

Был проведен опрос среди десяти человек, который содержал следующие вопросы:

1. Какое значение из таблицы 1 принимает слово “язык” в данном предложении: “*Да отсохнет язык у колокола, если он трезвонит зазря!*”?

2. Какое значение из таблицы 1 принимает слово “язык” в данном предложении: “*Язык колокола после удара должен мгновенно отскочить от звукового пояса, а не «прилипать» к нему (за этим следует следить через ножную педаль) — это значительно влияет на качество звона*”?

3. Какое значение из таблицы 1 принимает слово “язык” в данном предложении: “*Проблем с переговорами у нас не возникло, так что мы быстро нашли общий язык*”?

4. Какое значение из таблицы 1 принимает слово “язык” в данном предложении: “*Языком называют определённый код, систему знаков и правил их употребления*”?

В первом и втором случае все опрашиваемые выбрали значение 8, в третьем случае семь человек из десяти выбрали значение 3, двое выбрали

значение 4, а один человек выбрал значение 6. В четвертом случае девять человек выбрали значение 5 и один человек выбрал значение 3.

Работа системы “Nerpa”

Далее те же самые примеры были протестированы в системе “Nerpa”, после чего были получены следующие результаты:

Пример 1

Целевое слово: “*язык*”

Контекст: “*Да отсохнет язык у колокола, если он трезвонит зазря!*”

Результат: *“язык (деталь колокола, звонка и т. д.)”*

Результат представлен на рисунке 9.

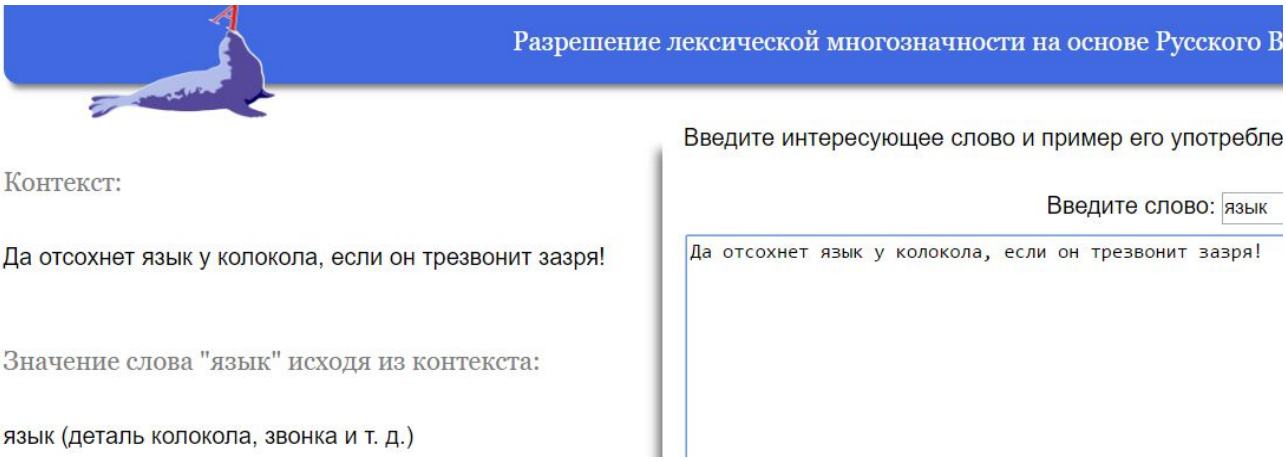

Рис. 9. Результат работы программы для примера 1

Пример 2

Целевое слово: “*язык*”

Контекст: “*Язык колокола после удара должен мгновенно отскочить от звукового пояса, а не «прилипать» к нему (за этим следует следить через ножную педаль) — это значительно влияет на качество звона*”

Результат: *“язык (деталь колокола, звонка и т. д.)”*

Пример 3

Целевое слово: “*язык*”

Контекст: “*Проблем с переговорами у нас не возникло, так что мы быстро нашли общий язык*”

Результат: *“то, что имеет форму языка”*

Пример 4

Целевое слово: “*язык*”

Контекст: “*Языком называют определённый код, систему знаков и правил их употребления*”

Результат: *“язык (знаковая система, средство общения)”*

Следующим экспериментом будет тестирование системы на примере из третьей главы данной работы:

Пример 5

Целевое слово: “*любовь*”

Контекст: “*Любовь к Родине – одно из самых мощных, возвышенных чувств. Она в полной мере проявилась в братской поддержке жителей Крыма и Севастополя, когда они твердо решили вернуться в свой родной дом*”

Результат:

1. *“духовное состояние, порожденное высшей степенью привязанности к чему-либо”*
2. *“пристрастие к чему-либо, увлеченность чем-либо”*

Результат представлен на рисунке 10.

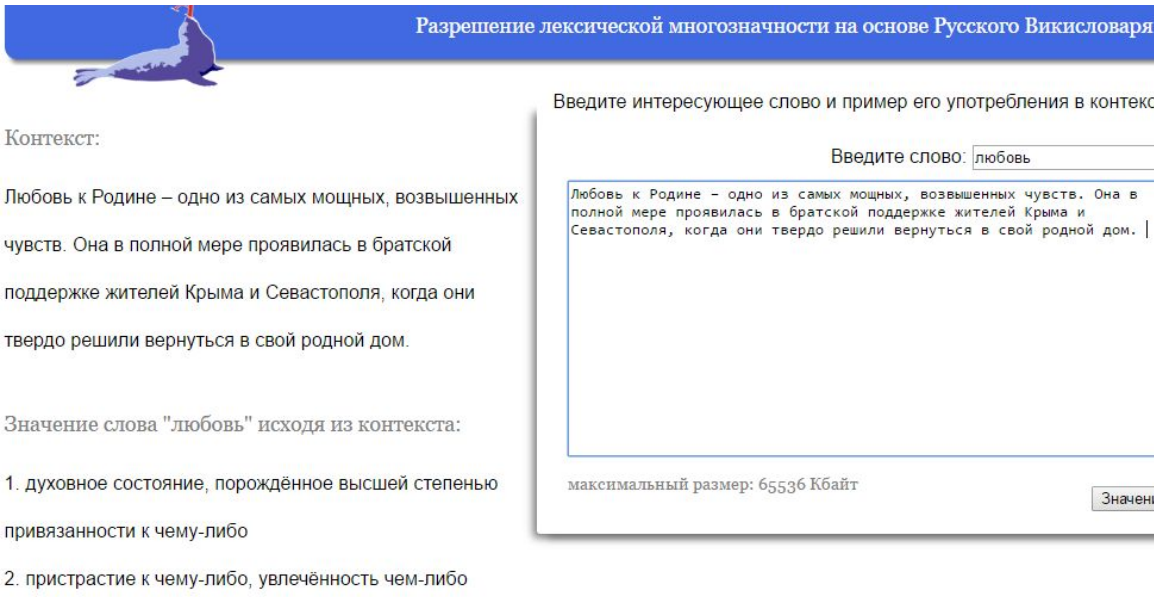

Рис. 10. Результат работы программы для слова примера 5

Пример 6

Целевое слово: “*дом*”

Контекст: “*Любовь к Родине – одно из самых мощных, возвышенных чувств. Она в полной мере проявилась в братской поддержке жителей Крыма и Севастополя, когда они твердо решили вернуться в свой родной дом*”

Результат: *“совокупность жилых или производственных корпусов, а также служебных строений, расположенных на одном земельном участке и имеющих один учётный номер”*

Результаты работы системы и ответы респондентов представлены в таблице 3:

Таблица 3. Результаты опроса и работы системы “Nerpa”

Целевое слово	**№ примера**	**Контекст**	**Номера ответов респондентов по таблице**	**Номера ответов системы по**	**Совпадение ответов респондентов с системой**

			2	**таблице 2**	
язык	1	Да отсохнет язык у колокола, если он трезвонит зазря!	десять человек - значение 8	значение 8	10
	2	Язык колокола после удара должен мгновенно отскочить от звукового пояса, а не «прилипать » к нему (за этим следует следить через ножную педаль) — это значительн			

		о влияет на качество звона			
	3	Проблем с переговорами у нас не возникло, так что мы быстро нашли общий язык	семь человек - значение 3 два человека - значение 4 один человек - значение 6	значение 9	0
	4	Языком называют определённый код, систему знаков и правил их употребления	девять человек - значение 5 один человек - значение 3	значение 5	9

Если считать точность ответа программы как $t = \frac{100\% \times Count(t)}{Count(r)}$, где Count(t) - количество ответов респондентов, совпавших с ответом системы, а Count(r) - количество респондентов, то точность ответа программы будет следующая:

Таблица 4. Точность работы алгоритма программы Nerpa

Номер примера	**Точность**
1	100%
2	100%
3	0%
4	90%

Исходя из полученных результатов можно сделать вывод, что в 1, 2 и 4 примерах ответы системы совпадают с ответами опрошенных людей, но в примере 3 система ошиблась. Также система ошиблась и примерах 5 и 6. Для того, чтобы понять причину ошибки можно проанализировать лексическую цепочку, построенную для примера 3. Система “Nerpa” построила следующую лексическую цепочку:

Слово *язык* в значении [*то,* ***что*** *имеет форму языка*] связано со словом *что* в значении [*употребляется при постановке* ***общего*** *вопроса о предмете, явлении, действии в вопросах о неодушевлённых субъектах*], которое в свою очередь связано со словом *общий* в значении [*касающийся самого основного, не затрагивающий деталей*]. Таким образом, полученная цепочка состоит из трех слов. Ошибка в выборе наиболее подходящего толкования произошла из-за того, что на данный момент система для установления лексической связности учитывает только повторы слов, а такие виды связи, как синонимы, гипонимы и гиперонимы не учитывает. Исправить ошибку можно путем добавления в алгоритм системы эти виды связи. Когда это будет сделано, в лексическую цепочку будет попадать больше слов из контекста и результат работы программы, возможно, будет точнее.

Заключение по пятой главе

В данной главе было проведено четыре аттестационных теста системы “Nerpa” и проведено сравнение работы системы с ответами респондентов путем проведения опроса. Эксперименты показали, что в трех случаях из шести система верно определила значение слова, а в примерах 3, 5 и 6 возникают ошибки. Было определено, что для более точного нахождения значения слова требуется добавление в алгоритм программы таких видов связей, как синонимы, гипонимы, гиперонимы и часто употребляемые вместе слова.

Заключение

В ходе данной работы были изучены такие понятия, как:

Лексическая многозначность (полисемия) — это наличие у слова нескольких взаимосвязанных значений, характеризуемых общностью одного или более семантических компонентов [17].

Разрешение лексической многозначности (англ. word sense disambiguation или WSD) — задача определения смысла (значения) слова, которое принимается в определенном контексте [7].

Лексическое единство в тексте – это результат цепей связанных слов, которые способствуют непрерывности общей темы повествования [3].

Были приведены основные подходы к разрешению лексической многозначности:

- WSD-методы, основанные на знаниях (knowledge);
- WSD-методы с учителем (supervised);
- Полуобучаемые или минимально-контролируемые методы (Semi-supervised);
- WSD-методы без учителя (unsupervised, кластеризация и графы);

Подробно рассмотрен алгоритм построения лексической цепочки для решения задачи WSD.

Основная цель работы достигнута: был разработан алгоритм построения лексической цепочки с помощью Русского Викисловаря. Было показано как словарные статьи Викисловаря могут использоваться в процессе построения лексических цепей. Данный алгоритм может использоваться для создания или повышения точности существующих

систем, предназначенных для обработки или анализа текстов на естественном языке.

Также была разработана система “Nerpa”, реализующая алгоритм построения лексических цепочек для разрешения лексической многозначности на основе Русского Викисловаря.

Эксперименты проведенные с системой показали, что не во всех примерах программа работает верно. Было определено, что для более точного нахождения значения слова требуется улучшение алгоритма путем добавления в него таких видов семантических связей, как синонимы, гипонимы, гиперонимы и часто употребляемые вместе слова. На данный момент система для установления лексической связности учитывает только повторы слов. При увеличении видов связности в алгоритме произойдет увеличение количества и длин получаемых лексических цепочек, а следовательно и точности результатов.

Для улучшения системы в дальнейшем планируется:

1. Реализация в системе таких видов лексической связности, как синонимы, гипонимы, гиперонимы и часто употребляемые вместе слова;
2. Вычисление сильной цепочки с помощью расчета сил связей (повтор слов - 3, синоним, гипоним, гипероним - 2, часто употребляемые вместе слова - 1) и расстояния между словами в контексте (чем дальше слова друг от друга, тем сила связи меньше). На данный момент сильная цепочка вычисляется путем подсчета количества входящих в нее слов;

Литература

1. D. Duong. Automated text summarization. Graduation Thesis. Hanoi University. 2011. 117 p.
2. G. Salton. Automatic Information Organization and Retrieval. — McGraw Hill Text, 1968.
3. J. Morris, G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 1991. Vol. 17, N 1. P. 21–43.
4. K. Litowski. Desiderata for tagging with WordNet sysnsets or MCAA categories // In Proceedings of the ACL-SIGLEX Workshop "Tagging Text with Lexical Semantics: Why, What, and How?"pages 12–17. — Washington, DC, 1997. — April.
5. M. Galley, K. McKeown. Improving word sense disambiguation in lexical chaining. 2003.
6. M. Halliday, R. Hasan. Cohesion in English. 1976. 374 p.
7. P. Edmonds, E. Agirre. Word sense disambiguation. Scholarpedia, 3(7):4358. (2008).
8. R. Barzilay, M. Elhadad. Using lexical chains for text summarization. In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization (Madrid, Spain). 1997. P. 10–17
9. R. Mihalcea. Using Wikipedia for Automatic Word Sense Disambiguation. (2007).
10. R. Navigli. Experiments on the validation of sense annotations assisted by lexical chains. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL, Trento, Italy). 2006. 129–136

11. R. Navigli. Word sense disambiguation: A survey. (2009).
12. W. Weaver. Translation. In Machine translation of languages (1949), pp. 15-23
13. Y. Kiselev, A. Krizhanovsky, P. Braslavski, I. Menshikov, M. Mukhin, N. Krizhanovskaya. Russian Lexicographic Landscape: a Tale of 12 Dictionaries. 2015.
14. А. А. Крижановский, С. С. Ткач. Применение лексических цепочек для разрешения лексической многозначности на основе Русского Викисловаря // Authorea. URL: https://www.authorea.com/users/86022/articles/104927/_show_article
15. А. В. Смирнов, В. М. Круглов, А. А. Крижановский, Н. Б. Луговая, А. А. Карпов, И. С. Кипяткова. Количественный анализ лексики русского WordNet и викисловарей // Труды СПИИРАН. — СПб., 2012. — Т. 23. — С. 231–253.
16. Новогоднее обращение Владимира Путина к гражданам России. RT. 01.01.2015.
17. С. А. Песина. Полисемия в когнитивном аспекте: Монография. — СПб.: Изд-во РГПУ им. А. И. Герцена, 2005. — 325 с.
18. Т. В. Каушинис и др. Обзор методов и алгоритмов разрешения лексической многозначности: Введение. // Труды КарНЦ РАН. No 10. Сер. Математическое моделирование и информационные технологии. 2015. С. 69-98