**Описание процесса проведения исследований**

**Формирование единого словаря строительной отрасли**

В целях формирования подходов к формированию единого словаря строительной отрасли изучался вопрос применения статистических методов. Статистические методы анализа текста могут быть применены для решения следующих задач:

1. Выявление коллизий в определениях терминов, определение которых присутствует в документах в явном виде
2. Формирование перечня потенциальных терминов строительной области
3. Анализ контекста употребления терминов с помощью метода Word2Vec

Первым шагом является формирование набора (корпуса) документов, который будет использоваться для дальнейшего анализа. Файлы документов должны быть в формате Office Open XML (OOXML, Microsoft Word .docx). Отобранные документы переводятся в текстовый формат с кодировкой UTF-8.

Следующим шагом является формирование перечня терминов и их определений на основании разделов “Основные понятия” корпуса документов. Ввиду того, что оформление документов носит разрозненный характер, выделение терминов и их определений в автоматизированном режиме потребует разработки шаблона отдельно под каждое оформление. В данном конкретном случае эффективней данную работу выполнить вручную.

Зачастую в документе термин приводится в полной и сокращенной форме, используемой далее по тексту документа. Например: “закладка материальных ценностей в государственный материальный резерв (далее – закладка)”. В таком случае термином считается именно сокращенная форма. Также нередко в скобках приводится синоним термина. Например: “биоаналогичный лекарственный препарат (биоаналог, биоподобный лекарственный препарат, биосимиляр)”. В таких случаях право выбора термина остается за человеком, выполняющим эту работу. Помимо этого термин зачастую приводится во множественном числе. Таким термины должны быть переведены в единственное число. Результат работы оформляется в виде таблицы с определённой структурой.

Таблица \_\_\_. Структура таблицы, содержащей термины и их определения

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование поля** | **Описание** |
| 1 | Термин | Термин из документа, оформленный в соответствии с вышеизложенными правилами |
| 2 | Исходный термин | Исходный термин |
| 3 | Определение термина | Определение термина из документа |
| 4 | Источник | Код документа (из таблицы \_\_) |

Для целей дальнейшей автоматизированной обработки таблица выгружается в CSV формат и загружается в базу данных.

Исходный текст модуля загрузки приведён в Приложении \_\_ (moduleLoadTermsDefinition.py)

Структура базы данных приведена в Приложении \_\_

Далее, автоматизированным способом, определяются документы, в которых дублируются терминов и определения. Исходный текст запроса к базе данных приведён в Приложении \_\_ (v\_doubles.sql). Затем определяются определяются коллизии в определениях терминов. Исходный текст запроса к базе данных приведён в Приложении \_\_ (v\_def\_mismatches.sql).

После этого автоматизированным способом выполняется частотный анализ корпуса с целью выявления потенциальных терминов. Частотный анализ выполняется с применением языка программирования Python 3.7.2 и библиотек nltk и pymorhy2. При определении потенциальных терминов используются следующие правила:

1. Потенциальный термин состоит из СУЩЕСТВИТЕЛЬНОГО в ИМЕНИТЕЛЬНОМ падеже;
2. Потенциальный термин состоит из ПРИЛАГАТЕЛЬНОГО в ЛЮБОМ падеже и СУЩЕСТВИТЕЛЬНОГО в ИМЕНИТЕЛЬНОМ падеже;
3. Потенциальный термин состоит из СУЩЕСТВИТЕЛЬНОГО в ИМЕНИТЕЛЬНОМ падеже и ПРИЛАГАТЕЛЬНОГО в ИМЕНИТЕЛЬНОМ падеже;
4. Потенциальный термин состоит из СУЩЕСТВИТЕЛЬНОГО в ИМЕНИТЕЛЬНОМ падеже и СУЩЕСТВИТЕЛЬНОГО в РОДИТЕЛЬНОМ падеже;
5. Потенциальный термин состоит из СУЩЕСТВИТЕЛЬНОГО, ПРИЛАГАТЕЛЬНОГО, ПРИЛАГАТЕЛЬНОГО
6. Потенциальный термин состоит из ПРИЛАГАТЕЛЬНОГО, ПРИЛАГАТЕЛЬНОГО, СУЩЕСТВИТЕЛЬНОГО.

Исходный текст модуля частотного анализа приведён в приложении \_\_\_ (moduleFreqEnities\_vTP1202.py).

После определения частоты встречаемости потенциальных терминов необходимо определить число N, которое позволит отнести выделенные слова и словосочетания к категории “часто встречающийся термин”.

1. Для этого воспользуемся законом Ципфа («ранг—частота») который гласит, что если все слова языка (или просто достаточно длинного текста) упорядочить по убыванию частоты их использования, то частота *n*-го слова в таком списке окажется приблизительно обратно пропорциональной его порядковому номеру *n* (так называемому *рангу* этого слова), и выполним ранжирование потенциальных терминов. Исходный текст запроса к базе данных, выполняющий ранжирование приведён в приложении \_\_\_ (v\_termcount.sql, v\_mentions.sql);
2. Далее рассчитываем TF (term frequency — частота вхождения слова — отношение числа вхождений (количества упоминаний) некоторого слова к общему числу слов документа) и С (распределение вероятности встречаемости термина - (Частота вхождения слова х Ранг частоты) / Общее число слов). Исходный текст запроса к базе данных приведён в приложении \_\_\_ (v\_tf.sql);
3. Далее рассчитываем стандартное квадратичное отклонение величины C. Исходный текст запроса к базе данных приведён в приложении \_\_\_ (v\_stddev.sql);
4. На основании рассчитанного стандартного квадратичного отклонения определяем N для каждого документа корпуса. Исходный текст запроса к базе данных приведён в приложении \_\_ (v\_n.sql);
5. Фильтруем список потенциальных терминов в соответствии с N. Исходный текст запроса к базе данных приведён в приложении \_\_ (v\_terms\_n.sql).

При необходимости, полученный список терминов можно дополнительно отфильтровать исключив из него термины, имеющие IDF (*inverse document frequency* — обратная частота документа) близкий или равный 0, что позволит исключить из списка термины, имеющие широкое употребление, выходящее за рамки предметной области строительства и архитектуры. Дальнейший анализ выполнялся без применения фильтра по IDF. Исходный текст запроса к базе данных, выполняющий расчёт IDF приведён в приложении \_\_\_ (v\_idf.sql).

Для анализа контекста упоминания терминов и определение слов-синонимов используется технология Word2Vec. Технология Word2Vec разработана группой исследователей Google в 2013 году. Работу над проектом возглавил Томаш Миколов. Word2Vec собирает статистику по совместному появлению слов в фразах, после чего методами нейронных сетей решает задачу снижения размерности и выдает на выходе компактные векторные представления слов, в максимальной степени отражающие отношения этих слов в обрабатываемых текстах. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), в векторном представлении будут иметь близкие координаты векторов-слов. В настоящем НИР используется реализация Word2Vec из пакета gensim. Таким образом, технологию Word2Vec можно применять для нахождения синонимов, определения контекста употребления термина, исправления опечаток в тексте.

Перед построением моделей Word2Vec для документов необходимо провести дополнительную обработку корпуса документов. Так как Word2Vec оперирует отдельными словами, необходимо выделить термины в документах как отдельные слова. Для этого пробелы в названии терминов заменяются на символ “\_” и сами термины обрамляются символами “%”. Например: “проектно-сметная документация” -> “%проектно-сметная\_документация%”. Далее формируется модель Word2Vec отдельно для каждого документа и сохраняется на локальный диск для дальнейшего использования. Исходный текст модуля формирования моделей Word2Vec на языке Python приведён в Приложении \_\_ (moduleParseText.py).

Для получения списка терминов, семантически близких к заданному используется модуль на языке Python. Исходный текст модуля приведён в Приложении \_\_ (moduleCheckTerm.py). Для работы с модулем необходимо указать искомый термин. Результатом работы модуля является файл содержащей по оси Y - отобранные первые 10 слов. Слова, которые выделены знаками % присутствуют в списке терминов, которые были определены частотным анализом, по оси Х - ID документа. На пересечении указана мера "близости" между искомым термином и отобранными наиболее семантически близкими словами 10 словами в рамках указанного документа. Чем значение ближе к 1 - тем мера близости больше.

Дополнительно проводился частотный анализ глаголов, задающих уровень обязательности требований, указываемых в НПА/НТД.

Для выполнения частотного анализа глаголов использовалась реализация для русского языка библиотеки MaltParser на языке Java. Библиотека MaltParser является синтаксическим и морфологическим анализатором, позволяет получить представление синтаксической структуры анализируемого предложения в виде графа (дерево зависимостей). В настоящем НИР библиотека MaltParser используется для выполнения синтаксического разбора предложения и выделения глаголов, в том числе употребляемых с частицей “НЕ”. Исходный текст программного кода приведён в приложении \_\_ (mp4ru\_test.java)