Разработка алгоритма анализа технической документации

А.А. Андрюкеева, E-mail: andryukeeva@ya.ru

Московский авиационный институт

Аннотация. *Работа посвящена исследованию алгоритмов обработки технической документации. В ходе работы был разработан алгоритм, позволяющий выявить неправильное использование терминологии в документах.*

Ключевые слова: *техническая документация, анализ текста, валидация документации, информационная система, терминология.*

Введение

Сложность современного производства требует наличия множества нормативно-технических документов как внешнего, так и внутреннего происхождения.

Среди множества существующих проблем подготовки технической документации (ТД) [1] наиболее распространенными являются проверка соответствия структуры документации принятым стандартам, требованиям описываемого процесса или предмета, составление многочисленных однотипных документов, выявление сути документа, оперирование большим количеством сокращений и др. Также часто проблемой в содержательной части документов является нарушение терминологии в тексте.

Многие из этих проблем могут решаться автоматически, тем самым существенно сократив трудозатраты на разработку документов и обеспечив возможность больше времени уделить содержательной составляющей документов.

Ранее для решения некоторых этих проблем, была создана информационная система «ТехДок» [2], позволяющая анализировать документ на наличие типичных ошибок и осуществлять валидацию по заданным шаблонам. Система может применяться для создания шаблонов различных видов документов, анализа содержания документа, валидации структуры документа, валидации документа на соответствие шаблону. Также система позволяет проверять стили и шрифты на соответствие параметрам, заданным в шаблоне, и отображает найденные ошибки. Пользователь может составлять стили в редакторе шаблонов. Помимо вышеупомянутых возможностей, система имеет возможность создавать каркас документов по шаблонам автоматически,

что решает проблему составления многочисленных однотипных документов [3].

1. Проблема единообразия используемой терминологии

Оформление технической документации несомненно является важным этапом ее составления. Однако, работа над содержанием - еще более сложный процесс, который трудно автоматизировать.

Технический писатель, человек, который занимается составлением ТД и поддержанием ее в актуальном состоянии, может быть специалистом в одной или нескольких предметных областях, но иногда он может столкнуться с такой, в которой имеет мало опыта или вовсе не разбирается. Отсюда может возникнуть проблема правильного применения терминологии, что в ТД является особенно важным аспектом.

Например, для специалиста, ранее работавшего с модулем системы, отвечающим за пользовательский интерфейс системы [4], могут возникнуть трудности при его переходе на работу с модулем взаимодействия с операционной системой. Еще большую сложность представляют собой переходы между разными предметными областями систем.

2. Алгоритм анализа терминологии

Для решения проблемы использования терминологии разработан следующий алгоритм:

1. Термин, по которому необходимо осуществлять проверку, ставится в начальную форму.
2. Из текста выделяются предложения.
3. Поиск первого вхождения термина в тексте и выделение зависимых слов у первого вхождения.
4. Далее в каждом предложении находятся главные слова и их зависимые.
5. Сравниваются зависимые слова первого вхождения термина в тексте с зависимыми словами главных членов предложения.
6. При условии равенства зависимых слов, сравниваются их главные слова, то есть исходный термин и главное слово в предложении.
7. Если равенство не выполнятся, можно сделать вывод о неправильном применении термина.

Алгоритм основан на том, что при первом вхождении термина он употреблен верно. Сравнение зависимых слов помогает наиболее точно выделить терминологию для текста в его предметной области.

3. Реализация алгоритма анализа терминологии

Алгоритм анализа реализован на языке Java с использованием библиотеки TAWT [5]. Алгоритм реализован в классе TermValidator, который включает в себя в качестве полей экземпляры объектов модулей SyntaxParser и JMorfSdkFactory, и метод findMatch.

В методе findMatch входящими параметрами являются проверяемый термин и текст, в котором необходимо проанализировать употребление данного термина. Метод основан на выявлении опорных оборотов, которые состоят из главного слова и зависимых от него оборотов, то есть дерева предложения.

Алгоритм работы метода следующий:

1. Термин при помощи модуля JMorfSdkFactory приводится к начальной форме.
2. Происходит поиск первого вхождения термина в тексте:
   – модулем SyntaxParser текст разбивается на опорные обороты;
   – в каждом опорном обороте выделяется главное слово в начальной форме и сравнивается с термином;
   – когда такой опорный оборот удовлетворяет условию, записывается его номер и список зависимых оборотов.
3. Для каждого последующего опорного оборота выделяются их зависимые слова и сравниваются с зависимыми от термина. Так как список зависимых слов является деревом, то производится рекурсивная обработка каждого зависимого опорного оборота текста рекурсивно.
4. Когда в тексте найден такой оборот, в котором зависимые равны зависимым термина, то сравнивается главное слово найденного оборота и термин.
5. В случае совпадения слов делается вывод о том, что термин употреблен правильно. В обратном случае – найдена ошибка применения терминологии, которая фиксируется в отчете работы метода.

Например, исходный текст: «В подсистему формирования отчетности должны быть включены механизмы гибкой настройки. В подсистему создания отчетности включены механизмы гибкой настройки.»

Термин, по которому необходима проверка: «формирование».

В тексте будет найдено первое вхождение термина в первом предложении. Зависимым опорным оборотом будет слово «отчетности». В следующем предложении будет найдено такое же слово среди зависимых опорных оборотов, однако, главным в нем будет слово «создание», то есть отличное от термина, поэтому можно предположить,

что допущена ошибка применения «подсистема создания отчетности» вместо «подсистема формирования отчетности».

На рисунке схематично изображены опорные обороты предложений, приведенных в примере. Прямоугольниками отмечены опорные обороты, зеленым – их главные слова, синим – зависимые.

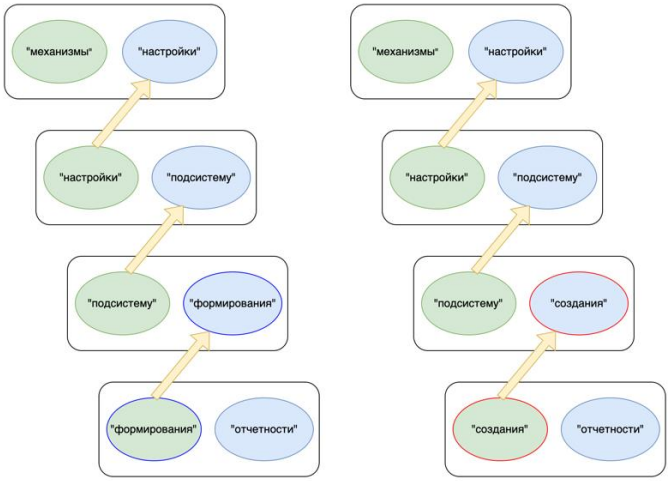

Рисунок. Опорные обороты предложений

Заключение

С помощью предложенного алгоритма можно определять точное несоответствие главных слов зависимых опорных оборотов и термина по всему дереву зависимых слов предложения. В дальнейшем планируется расширить работу алгоритма сравнением слов с применением синонимизации.

Применение предложенного алгоритма позволит реализовать в информационной системе «ТехДок» функцию валидации терминологии, которая даст пользователям возможность быстро и наглядно получить список проблемных мест в документе для оперативного исправления.

Литература

1. ГОСТ 3.1109-82 ЕСТД. Термины и определения основных понятий [Электронный ресурс] : ГОСТ. – Дата обращения: 05.01.2021 – Режим доступа: https://files.stroyinf.ru/Data2/1/4294845/4294845082.pdf

2. ИС «ТехДок» [Электронный ресурс] : приложение. – Дата обращения: 09.01.2021 – Режим доступа: https://yadi.sk/d/-tVo2lvW0V2JLg

3. Андрюкеева А. А. Исследование алгоритмов анализа и генерации технической документации [Электронный ресурс] : статья. – Дата обращения: 12.01.2021 – Режим доступа: https://www.cs.vsu.ru/ipmt-conf/conf/2021/Proceeding_IPMT_2021.pdf

4. ГОСТ Р ИСО/МЭК 19762-1-2011 Информационные технологии [Электронный ресурс] : ГОСТ. – Дата обращения: 10.01.2021 – Режим доступа: https://docs.cntd.ru/document/1200090097

5. Politsyna E., Politsyn S., Porechny A. Solving practical tasks of computer linguistics using the created text processing framework [Электронный ресурс] : статья. – Дата обращения: 15.12.2021 – Режим доступа: https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012129