УДК 004.89

АЛГОРИТМ ВЫДЕЛЕНИЯ ТРЕБОВАНИЙ ИЗ ТЕКСТОВ НОРМАТИВНЫХ ДОКУМЕНТОВ

АРШАНСКИЙ А.Р[1]., БЕГЛАРЯН Н.М.[1], ЧИКУНОВА М.В.[1], МИЩЕНКО И.О.[1], МАКСИМОВ И.В[2]

1 - Акционерное общество "Русатом Автоматизированные системы управления" (Москва, Российская Федерация)

2- Частное учреждение по цифровизации атомной отрасли "Цифрум" (Москва, Российская Федерация)

Аннотация. В статье рассмотрены основные этапы процесса формализации требований, содержащихся в текстах нормативных документов, включающие атомизацию (выделении фрагментов текста из исходных документов) и классификацию фрагментов текста. Разработаны алгоритмы атомизации нормативных документов типа IEC, а также классификации полученных фрагментов текстов и выделения требований. Полученные результаты обладают достаточными параметрами качества для использования в рамках управления требованиями и позволяют сокращать время процесса формализации требований.

Ключевые слова: нормативный документ, требование, МЭК, классификация текста, машинное обучение, обработка естественного языка

AN ALGORITHM FOR REQUIREMENTS EXTRACTION IN TECHNICAL STANDARDS

ALEXEY.R. ARSHANSKIY[1], NANE.M. BEGLARYAN[1], MARY.V. CHIKUNOVA[1], IGOR.O. MISHCHENKO, IVAN.V. MAKSIMOV[2]

1 - JSC «Rusatom Automated Control Systems», (Moscow, Russian Federation)

2 - Cifrum Private Enterprise, Rosatom, (Moscow, Russian Federation)

Abstract. The purpose of the study is to develop an algorithm for the requirement extraction in technical standards. This process contains two stages: document fragmentation and classification of fragments. The results of the study show that the algorithm has good quality and may be used in the requirement management process.

Keywords: technical standard, requirement, IEC, text classification, machine learning, natural language processing

Введение

В последнее время все большее внимание уделяют процессу управления требованиями. Особенно это касается действительно больших проектов, таких как проектирование и строительство атомных электростанций (АЭС), где объем требований исчисляется десятками и сотнями тысяч. Требование — это формулировка ожидаемого свойства, поведения или характеристик продукта. Для организации процесса управления требованиями существует класс информационных систем, которые так и называются - система управления требованиями. Основные требования заказчика к проектированию и сооружению АЭС формируются в EPC-контракте. Помимо этого, для получения лицензии на эксплуатацию АЭС у государственных регулирующих органов, проект должен удовлетворять требованиям национальных и международных стандартов (IEC, IEEE, IAEA, ISO и др.). В связи с чем, первоочередной задачей является выявление требований из текстов нормативных документов (НД) в качестве отдельных самостоятельных сущностей для организации процесса управления требованиями. НД обычно предоставляются в виде электронных документов в формате *pdf*, в них требования описаны в текстовом слабоструктурированном виде. В первую очередь необходимо разделить текст НД на сущности (фрагменты текста), в которых могут содержаться требования. Данная процедура называется атомизацией текста. Кроме требований в тексте НД содержатся и другие данные: информация, заголовки разделов, таблицы, рисунки, пометки. Поэтому необходимо классифицировать сущности.

Общий процесс обработки НД для выделения требований из НД следующий:

1. Атомизация НД на отдельные сущности.
2. Классификация сущностей.

Атомизация нормативных документов

Первой задачей является разделение текста НД на сущности. Под сущностью следует понимать неделимую структурную единицу текстового описания, которая формируется абзацем текста, таблицей или рисунком. Атомизация НД проводилась с помощью синтаксического анализатора. Синтаксический анализатор разбирает разметку и разбивает текст на сущности. Кроме того, немаловажной частью его работы является сквозная нумерация сущностей, которая служит идентификацией. Каждый идентификатор является однозначной ссылкой на требование.

Существует много признаков для определения разметки: положение тех или иных символов, цифр, заглавных и строчных букв. Используя эти признаки в качестве эвристик, мы получили очень простой и легко управляемый синтаксический анализатор. Каждому признаку назначался некий вес, который участвовал в итоговой формуле управления строкой, и уже по сумме этих весов формировались конечные данные. Эвристический метод просто масштабируем ввиду линейности синтаксического анализа. Данный алгоритм также легко адаптируем под новые типы документов.

Предварительная обработка фрагментов текста

Следующим этапом работы алгоритма является процесс классификации полученных фрагментов текста. Поскольку нет явных правил, по которым можно классифицировать фрагменты текста на требования и общую информацию, то в данной работе применялись методы машинного обучения и обработки естественного языка для решения поставленной задачи. Для использования классических моделей машинного обучения необходимо перевести текст в векторное пространство признаков, которое его характеризует. Таким образом, возникает необходимость в процессе векторизации: создании для каждой сущности вектора признаков, который будет использован моделью машинного обучения для классификации сущности.

В данной работе в качестве векторного представления фрагментов текста была выбрана мера TF-IDF [1]. Данная мера определяет статистический вес слова. Для реализации метода требуется провести предварительную обработку данных, заключающуюся в приведении слов к нормальной форме (лемматизация) и удалению лишних символов. Нормальная форма — первое лицо, единственное число, именительный падеж, настоящее время, написание маленькими буквами. Удаление лишних символов заключается в очистке текста от цифр, знаков пунктуации, специальных символов и прочих, не относящихся к написанию слов. Кроме того, удаляются так называемые стоп-слова — не несущие особого смысла. Для английского языка, с которым и проводилась работа, такими, например, являются предлоги, артикли и др.

Таким образом, процесс предобработки данных для дальнейшего использования моделями машинного обучения представлял набор следующих шагов:

1. Очистка текста, которая выполнялась с использованием регулярных выражений [2];
2. Лемматизация слов, которая проводилась с помощью библиотеки *nltk* [3];
3. Перевод в векторное представление фрагментов текст методом TF-IDF реализовывалась с помощью библиотеки *Scikit-learn* [4].

Классификация фрагментов текста

Решение задачи классификации сущностей проводилось с использованием технологий машинного обучения. В качестве базового алгоритма был применен наивный классификатор, который отмечал классом “требование” все сущности, в которых были модальные глаголы (*shall, must, can* и др). С этим базовым классификатором сравнивались все созданные модели машинного обучения. В работе использовались несколько подходов. Первый — это байесовский классификатор [5], второй - градиентный бустинг [6].

Исходные данные для обучения представляли из себя векторное представление сущностей. Размер исходной матрицы признаков (примеры сущностей x признаки) составлял 3760x2580. Обучение классификаторов проводилось на обучающей выборке, которая составляла 50 % от общего объема данных (1880 примеров). В качестве целевого значения была следующая разметка на несколько классов: *Figure* (рисунок), *Heading* (заголовок подраздела), *Information* (информация), *Requirement* (требование), *Title* (заголовок раздела).

Для оценки алгоритмов и сравнения различных моделей использовались такие метрики, как *precision* (точность), *recall* (полнота), *f1* (среднее гармоническое точности и полноты) и *AUC ROC* - площадь под *ROC* кривой ошибок [7], являющейся отображением доли верно классифицированных объектов ко всем объектам. Данные метрики считались для классов “Требование” по схеме один против всех [8] на отложенной тестовой выборке (оставшиеся 50 % от всех данных). Результаты работы моделей приведены в табл. 1.

Таблица 1. Результаты работы моделей классификации

Метод	Precision (точность)	Recall (полнота)	f1-score	AUC ROC
Model verbs	**0.93**	0.86	0.89	0.82
BernoulliNB	0.90	0.94	0.92	0.80
ComplementNB	0.80	**1.00**	0.89	0.64
XGBoost	0.92	0.96	**0.94**	**0.84**

Лучшей моделью по метрике precision является модель, основанная на модальных глаголах, что обусловлено спецификой английского языка и правилами формулирования требований. ComplementNB показал наилучшие результаты по метрике полнота, но, учитывая точность классификации, можно сделать вывод о переобучении данного метода под класс “Требование”. Лучшей моделью по метрике f1, учитывающей и точность, и полноту, является XGBoost. Метрика AUC ROC, равная 0.84 подтверждает наличия преимущества у классификатора XGBoost.

Заключение

Основные результаты проделанной работы следующие:

1. Описан общий процесс работы с требованиями, содержащимися в НД. Он представляет собой последовательное выполнение следующих этапов:

а) Атомизация НД на отдельные сущности.

б) Классификация сущностей.

2. Разработан алгоритм атомизации исходных форматов НД на отдельные сущности, который позволяет выделять фрагменты текстов в табличный вид из слабоструктурированного формата *pdf*.

3. Построены несколько моделей классификации фрагментов текстов НД и проведено тестирование моделей на отложенной тестовой выборке. Лучшие результаты показала модель градиентного бустинга (XGBoost Classifier) со следующими показателями качества: *precision* (точность) = 0.92, *recall* (полнота) = 0.96, *f1* = 0.94.

4. Полученный алгоритм атомизации и классификации фрагментов текстов НД обладает достаточными характеристиками для использования в рамках управления требованиями и позволяет сокращать время процесса формализации требований.

Список литературы

1. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval //Information processing & management. – 1988. – Т. 24. – №. 5. – С. 513-523.

2. Van Leeuwen J. (ed.). Handbook of theoretical computer science (vol. A) algorithms and complexity. – Mit Press, 1991.

3. Loper E., Bird S. Nltk: The natural language toolkit //arXiv preprint cs/0205028. – 2002.

4. Pedregosa F. et al. Scikit-learn: Machine learning in Python //the Journal of machine Learning research. – 2011. – Т. 12. – С. 2825-2830.

5. Rish I. et al. An empirical study of the naive Bayes classifier //IJCAI 2001 workshop on empirical methods in artificial intelligence. – 2001. – Т. 3. – №. 22. – С. 41-46.

6. Friedman J. H. Stochastic gradient boosting //Computational statistics & data analysis. – 2002. – Т. 38. – №. 4. – С. 367-378.

7. Davis J., Goadrich M. The relationship between Precision-Recall and ROC curves //Proceedings of the 23rd international conference on Machine learning. – 2006. – С. 233-240.

8. Rifkin R., Klautau A. In defense of one-vs-all classification //The Journal of Machine Learning Research. – 2004. – Т. 5. – С. 101-141.