УДК 004.912 DOI: 10.14529/mmp250308

ИСПОЛЬЗОВАНИЕ НЕЧЕТКОГО СРАВНЕНИЯ СТРОК ПРИ РЕШЕНИИ ЗАДАЧИ АВТОМАТИЧЕСКОГО ПЕРЕНОСА ФОРМАТИРОВАНИЯ ПОЭТИЧЕСКИХ ПРОИЗВЕДЕНИЙ

Н.Н. Тесля[1]***, Г.Н. Беляк***[2]
[1]Санкт-Петербургский Федеральный исследовательский центр РАН,
г. Санкт-Петербург, Российская Федерация
[2]Институт русской литературы (Пушкинский Дом) РАН,
г. Санкт-Петербург, Российская Федерация

Создание научно-просветительского ресурса «Пушкин Цифровой» связано с необходимостью верстки стихотворных текстов на основе информации о верстке из других изданий. От издания к изданию тексты могут отличаться, и в каждом случае верстка осуществляется заново по правилам данного издания. Ручная верстка требует внимательности и существенных временных и трудовых затрат от специалиста, поскольку требуется сравнить несколько одинаковых текстов в нескольких изданиях. Представленный метод решает две задачи. Во-первых, определяется, насколько отличаются тексты в изданиях, обеспечивая возможность оценить количество ошибок или намеренных трансформаций текста, что является отдельным предметом исследования текстологов. Во-вторых, на основе оценки различия строк и нечеткого их сопоставления формируются правила верстки для каждой строки с учетом того, какие правила используются в ранних изданиях. Метод опробован на 914 текстах лирических произведений А.С. Пушкина, обеспечив корректный полный перенос верстки для 74,55 % текстов, тогда как для 25,45 % этого сделать не удалось и пришлось прибегнуть к ручной верстке.

Ключевые слова: нечеткое сравнение строк; расстояние Левенштейна; форматирование; обработка текста.

Введение

Научно-просветительский ресурс «Пушкин цифровой» (НПР) является уникальным проектом, направленным на сохранение авторского наследия А.С. Пушкина за счет цифровой трансформации предметной области филологических, источниковедческих и библиографических исследований на основе разработки теоретических, технологических основ и платформы автоматизации текстовой и мультимедийной обработки изданий о пушкинской эпохе. Создание НПР связано с необходимостью решения ряда сложных задач, одной из которых является подготовка верстки стихотворных текстов на основе анализа верстки ранее опубликованных текстов. Значительные вариации в оформлении и содержании текстов, которые могут существовать от одного издания к другому, осложняют задачу единообразной и точной верстки. При этом следует учитывать, что каждое новое издание помимо изменения верстки может содержать внутритекстовые изменения, связанные, например, с обнаружением ранее утраченного фрагмента текста или с решением редактора о сокращении текста. Например, в тексте произведения «Морю» составителями убрана практически целая строфа на основании неоднозначности её принадлежности именно к данному произведению (рис. 1). При оформлении текста могут быть также внесены изменения в пунктуацию либо заменены буквы в словах согласно действующим правилам орфографии.

Ручная верстка в таких случаях требует высокой степени внимательности и аккуратности от специалиста, которому необходимо сверять несколько версий текста и

учитывать даже малейшие отклонения в каждой из них. Так, например, при верстке текстов НПР «Пушкин цифровой» базовыми источниками текстов являются два собрания сочинений – полное собрание сочинений в 16 томах (1937–1959) [1] (далее – старое издание), которое доступно в виде цифрового издания на портале *https://feb-web.ru/feb/pushkin*, и полное собрание сочинений в 20 томах (1999 – ...) [2] (далее – новое издание), доступное только в виде печатных книг. Большая часть текстов в собраниях практически идентична, за исключением разницы в пунктуации и орфографии, но некоторые тексты приведены впервые (как правило, это касается ранних и поздних редакций), также часть текстов представлена в переработанном виде. Тексты, имеющие близкое содержание, могут быть сверстаны автоматически на основе верстки цифрового издания, при этом обязательным требованием является сохранение текста, представленного в новом издании.

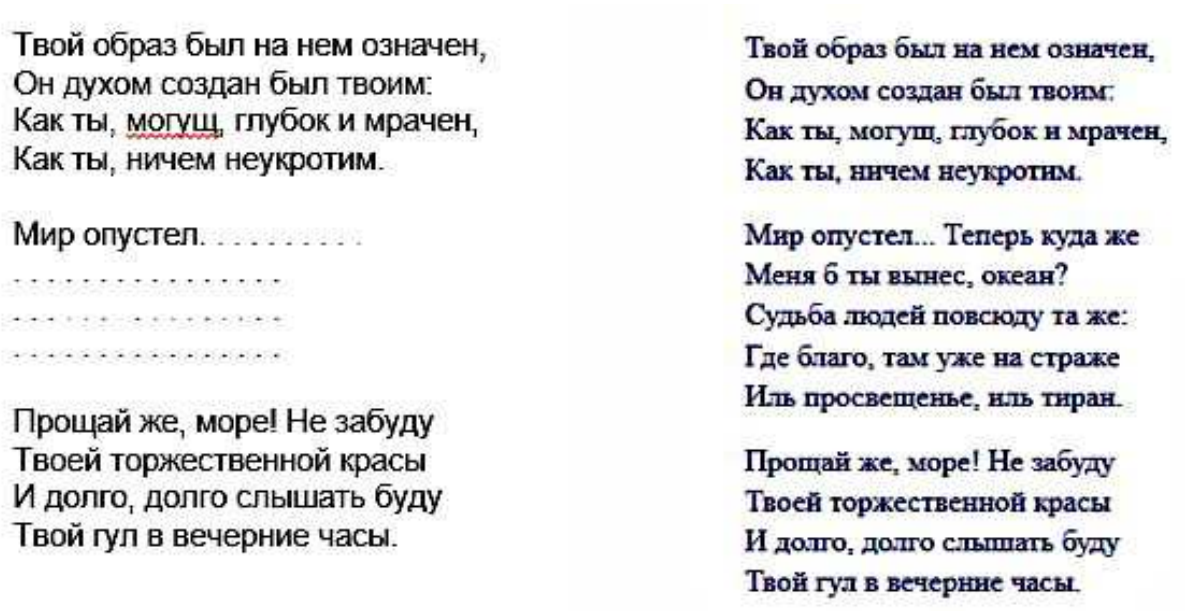

Рис. 1. Различие текстов произведения «Морю» между новым (слева) и старым (справа) собранием сочинений А.С. Пушкина

В данной работе предложен метод автоматического сравнения стихотворных текстов и переноса верстки между текстами, который позволяет облегчить и стандартизировать процесс подготовки контента для НПР «Пушкин Цифровой». В основу метода положена идея поиска и анализа похожих строк с последующим вычислением общей меры сходства текстов. Предложенный метод значительно сокращает время, необходимое для подготовки контента, и повышает точность воссоздания структуры и оформления стихотворных произведений, предоставляя инструмент для единообразного и точного переноса стилистических элементов из одного издания в другое, что показано на примере автоматизации подготовки текстов НПР «Пушкин цифровой».

1. Существующие методы сравнения текстов

Поскольку при копировании верстки необходимо решить задачу сравнения текстов, в данном разделе предлагается обзор существующих исследований и решений в данной предметной области.

Основные направления существующих исследований включают в себя следующие группы методов: основанные на анализе посимвольных различий между строками и

основанные на смысле текста [3]. Эти методы можно также сгруппировать на основе подходов, применяемых в анализе текста, – лингвистические, статистические, основанные на последовательностях, и методы, основанные на машинном обучении. Рассмотрим каждую из групп подробнее. Традиционные методы сравнения текстов основаны на анализе символьных последовательностей для вычисления метрик сходства:

- Вычисление редакционного расстояния (также известного, как расстояние Левенштейна). Определяет минимальное количество операций вставки, удаления и замены для преобразования одной строки в другую. Применяется для выявления различий на уровне символов или слов [4].

- Нечеткое сравнение строк (Fuzzy String Matching). Метод используется для сравнения строк с учетом неточных совпадений. Реализация представлена в алгоритмах, таких как Jaro-Winkler и сравнение n-грамм, также используются различные модификации расстояния Левенштейна, являясь менее строгими по сравению с базовым алгоритмом [5].

Следующей группой походов являются лингвистические, основанные на семантическом, синтаксическом и морфологическом анализе текстов для их сравнения:

- Сравнение текстов на уровне слов и фраз. Методы анализа слов с учетом их морфологических форм и синонимичных конструкций. Тексты разделяются на отдельные слова и проводится анализ семантической близости слов (или n-грам) с учетом порядка их следования в сравниваемых текстах, например, с использованием WordNet [6].

- Методы семантического сравнения. Являются расширением методов сравнения на уровне слов и фраз, но требуют перехода в другое представление текста. Слова сравниваются не напрямую, путем анализа словоформ, а путем трансформации их в векторное пространство (word embeddings) и дальнейшего анализа отношения векторов, чаще всего расстояния между векторами. Используются модели Word2Vec, GloVe, fastText для создания векторных представлений слов и предложений [7].

Расширением методов, основанных на сравнении строковых последовательностей и синтаксическом анализе, являются статистические методы, такие как:

- косинусное сходство (Cosine Similarity). Основано на сравнении текстов в векторном пространстве. Векторы строятся на основе моделей TF-IDF, Bag-of-Words или векторном представлении слов/текстов;

- мера сходства Жаккарда (Jaccard Similarity). Коэффициент, измеряющий схожесть двух текстов на основе пересечения и объединения их множеств слов или фраз [8].

Развитие методов машинного обучения применительно к обработке естественного языка позволяет использовать их в том числе и для анализа сходства текстов, что активно применяется в системах выявления плагиата. Современные подходы используют модели глубокого обучения, архитектуры на основе трансформеров для построения модели языка и последующего сравнения текстов, основываясь на их содержании.

- Сравнение на основе трансформеров (Transformers). Модели вроде BERT, GPT и их производные обеспечивают более глубокое семантическое понимание текстов и позволяют сравнивать их с учетом контекста [9].
- Нечеткое сопоставление текстов с помощью нейронных сетей, в частности использование архитектуры Siamese Neural Networks для сравнения пар текстов и определения меры их сходства [10, 11].
- Sentence-BERT (SBERT). Модифицированная версия BERT для получения эмбеддингов предложений и их дальнейшего сравнения. SBERT значительно ускоряет процесс вычисления семантической близости текстов [12].

Таким образом, исследования по сравнению текстов развиваются в нескольких направлениях: от простых алгоритмов до сложных нейросетевых моделей, что позволяет решать широкий спектр задач – от обнаружения плагиата до текстологического анализа и автоматической обработки больших текстовых коллекций.

2. Постановка задачи

Рассмотрим формализацию задачи сравнения текстов стихотворений для переноса верстки. Пусть даны два множества текстов:

$$T_1 = T_{1,1}, T_{1,2}, ..., T_{1,m}, T_2 = T_{2,1}, T_{2,2}, ..., T_{2,n}, i = [1, m], j = [1, n],$$

где $T_{1,i}$ и $T_{2,j}$ – тексты из множеств T_1 и T_2 соответственно и m и n – количество текстов в этих множествах. Каждый текст $T_{i,j}$ представляет собой множество строк со своими правилами верстки, которые требуется перенести. Таким образом, каждый текст можно записать как:

$$T_{i,j} = s_{i,j,1}, s_{i,j,2}, ..., s_{i,j,k_{i,j}},$$

где $s_{i,j,k_{i,j}}$ – строки в тексте $T_{i,j}$, а $k_{i,j}$ – количество строк в данном тексте.

Задача заключается в выборе двух элементов из произведения множеств текстов и формировании меры сходства между ними на основе сравнения строк в этих элементах. Мера сходства вычисляется как свертка построчных коэффициентов сходства. Если она превышает заданный порог θ, тексты считаются схожими и для каждой из строк возможен перенос верстки из текста $T_{1,i}$ в $T_{2,j}$. Формально для двух выбранных текстов $T_{1,i}$ и $T_{2,j}$ из множества T_1 и T_2 задача сводится к сравнению строк внутри этих текстов с подсчетом построчного уровня сходства и формированием пар схожих строк. Значение свертки коэффициентов сходства текстов используется в качестве основного критерия для принятия решения о переносе верстки.

Таким образом, для кортежа из двух элементов (текстов):

$$(T_{1,i}, T_{2,j}) \in T_1 \times T_2,$$

где $T_{1,i} \in T_1$ и $T_{2,j} \in T_2$, для каждой строки $s_1 \in T_{1,i}$ из текста $T_{1,i}$ и строки $s_2 \in T_{2,j}$ из текста $T_{2,j}$, вычисляется мера сходства. Каждая строка проходит процедуру токенизации с формированием множества токенов:

$$s_1 = w_{1,1}, w_{1,2}, ..., w_{1,d_1}, s_2 = w_{2,1}, w_{2,2}, ..., w_{2,d_2}, l = [1, d_1], p = [1, d_2],$$

где $w_{x,y}$ – слово y в строке s_x. Для пар токенов из множества токенов для строк вычисляется расстояние Левенштейна $L(s_1, s_2)$, на основе которого формируется общая оценка сходства строк. В результате формируется множество кортежей вида

$$(s_1, s_2, L(s_1, s_2)), s_1 \in T_{1,i}, s_2 \in T_{2,j}.$$

Общая мера сходства для двух текстов вычисляется как агрегированная мера сходства между всеми строками из $T_{1,i}$ и $T_{2,j}$:

$$Sim(T_{1,i}, T_{2,j}) = \frac{1}{k_{1,j} \cdot k_{2,j}} \sum_{s_1 \in T_{1,i}} \sum_{s_2 \in T_{2,j}} L(s_1, s_2), \tag{1}$$

где $k_{1,j}$ – количество строк в тексте $T_{1,i}$, $k_{2,j}$ – количество строк в тексте $T_{2,j}$.

3. Реализация метода

Для реализации предложенного метода использовался язык программирования Python, который предоставляет широкий спектр инструментов для обработки текстовой информации и реализации алгоритмов машинного обучения. Разработка отдельных модулей для сравнения строк основывалась на использовании специализированной библиотеки TheFuzz.

Библиотека TheFuzz обеспечивает функциональность для сравнения строк с применением расстояния Левенштейна в качестве базового алгоритма [13]. Она включает несколько модулей для различных сценариев обработки текстовых данных, в том числе метод, применяемый в данной задаче, – сравнение строк с предварительным разделением их на токены и вычислением средней меры сходства между соответствующими частями текстов. Это позволяет учитывать структурные различия между текстами при оценке их схожести.

Для решения задачи был разработан следующий алгоритм (рис. 2). Для каждой пары строк $(s_1, s_2), s_1 \in T_{1,i}, s_2 \in T_{2,j}$ осуществляется фильтрация пустых строк и строк, состоящих только из специальных символов (не являющихся буквами или цифрами). Это необходимый шаг, поскольку данные строки не имеют семантики в векторной модели и могут быть обработаны и учтены простым сравнением строк.

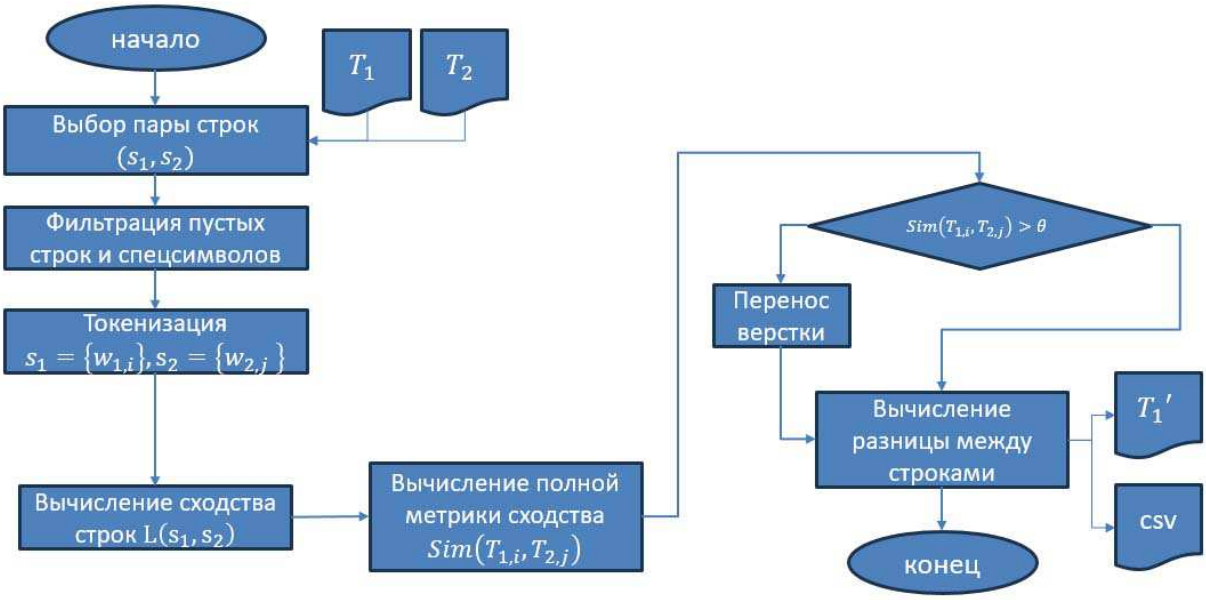

Рис. 2. Схема алгоритма сравнения стихотворных текстов

После фильтрации пустых строк оставшиеся пары строк приводятся к нижнему регистру, фильтруются на наличие пробелов и спецсимволов окончания строки, после чего сравнивается со строкой s_1 с использованием функции token_set_ratio библиотеки TheFuzz, реализующей оценку строк на основе разделения на токены. В качестве

контрольной оценки используется функция WRatio, автоматически игнорирующая специальные символы, но не осуществляющая токенизацию строк.

В случае, когда найдено совпадение строк, для каждой совпавшей пары вычисляется посимвольная разница с использованием функции ndiff библиотеки difflib, которая впоследствии может быть использована как материал для анализа работы алгоритма, так и для оценки различий в текстах специалистом-текстологом.

На основе множества найденных пар совпадающих строк вычисляется общая степень сходства текстов по (1). В случае превышения порога θ тексты считаются похожими и для них осуществляется построчный перенос правил верстки.

4. Оценка результатов

Метод опробован на 842 произведениях А.С. Пушкина, тексты которых взяты из старого и нового изданий. Для анализа подготовлено 554 текста из нового собрания сочинений и 844 из старого, при этом всего для произведений подготовлено 914 текстов без форматирования (из старого и нового собрания сочинений). В старом собрании сочинений имеется шесть уникальных текстов, которые были опубликованы в справочном томе и требуют ручного форматирования, поэтому они исключаются из рассмотрения. В новом собрании количество таких текстов составляет 20. Таким образом, из 914 текстов 888 имеют пример верстки из старого издания и в дальнейшем именно они будут рассматриваться как основной набор данных, используемых для оценки предложенного метода. Превышение количества текстов над количеством произведений вызвано тем, что многие произведения имеют несколько редакций, соответственно, при переносе верстки необходимо создать пару из правильных редакций. За счет построчного сравнения этот процесс выполняется автоматически при работе предложенного метода.

Итого, из всех произведений для 346 была успешно перенесена верстка между подготовленным корпусом и текстами из старого собрания сочинений (346 текстов), за исключением упомянутых ранее шести уникальных. Еще для 316 произведений из нового собрания сочинений тексты из корпуса по новому собранию сочинений успешно сопоставились с текстами из старого собрания (316 текстов) И для 180 произведений порог сходства был ниже установленного значения в 0,95, что означает невозможность однозначно перенести параметры верстки (216 текстов). В процентном соотношении для 74,55 % текстов удалось успешно перенести параметры верстки, тогда как для 25,45 % этого сделать не удалось и пришлось прибегнуть к ручной верстке.

Отдельно стоит отметить, что для полностью сопоставленных текстов только в 261 из 662 не найдено никаких различий. Остальные тексты отличаются наличием или отсутствием знаков препинания, отличием в несколько букв. Например, отличие текстов произведения «Простишь ли мне ревнивые мечты...» между старым [14] и новым собранием сочинений [15] помимо пунктуации заключается в 14 строке – «Ни слова мне, ни взгляда... друг жестокой!» (старое собрание) – «Ни слова мне, ни взгляда... друг жестокий!» (новое собрание). Причиной этих изменений может быть разный источник текста для печати, но каждый конкретный случай должен исследоваться специалистами, и результаты работы метода могут быть использованы как основание для дальнейших исследований.

Заключение

В ходе исследования был разработан и опробован метод автоматизированного переноса параметров верстки для стихотворных текстов. Метод основан на построчном сравнении текстов и формированием интегральной оценки сходства текстов. Каждая

строка сравнивается путем разбиения на токены и анализа посредством векторизации и оценки семантического сходства с использованием векторной модели языка BERT и посимвольного сравнения с использованием расстояния Левенштейна. Анализ работы проведен на корпусе произведений А.С. Пушкина, включающем 842 произведения из старого и нового собраний сочинений, общей численностью 914 текстов. Метод позволил успешно перенести параметры верстки для 662 текстов (74,55 %), что значительно ускоряет процесс подготовки материалов для научно-просветительского ресурса «Пушкин Цифровой».

Однако для 25,45 % текстов порог сходства оказался ниже установленного значения, что потребовало ручной обработки. Основными причинами расхождений были различия в пунктуации, опечатки, а также несоответствия в отдельных строках. Эти случаи выявляют сложность редакционных изменений и подчеркивают необходимость их детального изучения текстологами.

Работа выполнена в рамках реализации Государственного задания FFZF-2023-0001.

Литература

1. Пушкин, А.С. Полное собрание сочинений: В 16 т. / А.С. Пушкин. – М.: Изд-во академии Наук СССР, 1937–1959.
2. Пушкин, А.С. Полное собрание сочинений: В 20 т. / А.С. Пушкин. – СПб.: Наука, 1999.
3. Jiapeng Wang. Measurement of Text Similarity: a Survey / Jiapeng Wang, Yihong Dong // Information. – 2020. – V. 11, № 9. – Article ID: 421. – 17 p.
4. Rani, S. Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity / S. Rani, J. Singh // Computing, Analytics and Networks: First International Conference (ICAN 2017). – Chandigarh, 2018. – P. 72–80.
5. Pikies, M. Analysis and Safety Engineering of Fuzzy String Matching Algorithms / M. Pikies, J. Ali // ISA Transactions. – 2021. – V. 113. – P. 1–8.
6. Kenter, T. Short Text Similarity with Word Embeddings / T. Kenter, M. De Rijke // Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. – 2015. – P. 1411–1420.
7. Mikolov, T. Efficient Estimation of Word Representations in Vector Space / T. Mikolov, Kai Chen, G. Corrado, J. Dean // arXiv: Computation and Language. – 2013. – URL: https://arxiv.org/abs/1301.3781
8. Thada, V. Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieved Documents using Genetic Algorithm / V. Thada, V. Jaglan // International Journal of Innovations in Engineering and Technology. – 2013. – V. 2, № 4. – P. 202–205.
9. Patricoski, J. An Evaluation of Pretrained BERT Models for Comparing Semantic Similarity across Unstructured Clinical Trial Texts / J. Patricoski-Chavez, K. Kreimeyer, A. Balan, K. Hardart et al. // Informatics and Technology in Clinical Care and Public Health. – 2022. – P. 18–21.
10. Neculoiu, P. Learning Text Similarity with Siamese Recurrent Networks / P. Neculoiu, M. Versteegh, M. Rotaru // Proceedings of the 1st Workshop on Representation Learning for NLP. – 2016. – P. 148–157.
11. Amin, K. Advanced Similarity Measures Using Word Embeddings and Siamese Networks in CBR / K. Amin, G. Lancaster, S. Kapetanakis et al. // Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference. – 2020. – P. 449–462.

12. Reimers, N. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks / N. Reimers, I. Gurevych // arXiv: Computation and Language. – 2019. – URL: https://arxiv.org/abs/1908.10084

13. Thefuzz – Fuzzy String Matching in Python. – URL:https://github.com/seatgeek/thefuzz

14. Пушкин, А.С. Простишь ли мне ревнивые мечты... / А.С. Пушкин // Полное собрание сочинений: В 16 т. – Т. 2, кн. 1. Стихотворения, 1817–1825. Лицейские стихотворения в позднейших редакциях. – М.; Л.: Издательство АН СССР. – 1947. – С. 300–301.

15. Пушкин, А.С. Простишь ли мне ревнивые мечты... / А.С. Пушкин // Полное собрание сочинений: В 20 т. – Т. 2, кн. 2. – СПб: Наука. – 2016. – С. 91–92.

Николай Николаевич Тесля, кандидат технических наук, лаборатория интегрированных систем автоматизации, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (г. Санкт-Петербург, Российская Федерация), teslya@iias.spb.su.

Гавриил Николаевич Беляк, кандидат филологических наук, отдел пушкиноведения, Институт русской литературы (Пушкинский Дом) РАН (г. Санкт-Петербург, Российская Федерация), gabriel.belyak@gmail.com.

Поступила в редакцию 19 декабря 2024 г.

MSC 68T50 **DOI: 10.14529/mmp250308**

USING FUZZY STRING COMPARISON FOR AUTOMATED TRANSFER OF FORMATTING IN POETIC WORKS

***N.N. Teslya*[1], *G.N. Belyak*[2]**
[1]St. Petersburg Federal Research Center of the Russian Academy of Sciences, Saint-Petersburg, Russian Federation
[2]Institute of Russian Literature (Pushkinskij Dom) of the Russian Academy of Sciences, Saint-Petersburg, Russian Federation
E-mail: teslya@iias.spb.su, gabriel.belyak@gmail.com

The creation of the scientific and educational resource ≪Pushkin Digital≫ is driven by the necessity of typesetting poetic texts based on layout information from other editions. From one edition to another, texts may vary, and in each case, typesetting is performed anew according to the rules of the specific edition. Manual typesetting demands attentiveness and significant time and effort from a specialist, as it requires comparing several identical texts across multiple editions. The proposed method addresses two tasks.

First, it determines the extent to which the texts differ between editions, enabling an assessment of the number of errors or deliberate transformations of the text, which is a separate subject of study for textual scholars. Second, based on an evaluation of line differences and their fuzzy alignment, the method generates typesetting rules for each line, taking into account the rules applied in earlier editions.

The method was tested on 914 lyrical works by A.S. Pushkin, successfully ensuring the correct and complete transfer of typesetting for 74,55% of the texts. However, for 25,45% of the cases, this proved unfeasible, requiring manual typesetting instead.

Keywords: fuzzy string comparison; levenshtein distance; formatting; text processing.

References

1. Pushkin A.S. *Polnoe sobranie sochinenii: 16 t.* [Complete Works: In 16 Volumes]. Moscow, Leningrad, Izdatelstvo AN SSSR, 1937–1959. (in Russian)
2. Pushkin A.S. *Polnoe sobranie sochinenii: 20 t.* [Complete Works: In 20 Volumes]. St. Petersburg, Nauka, 1999–... (in Russian)
3. Wang Jiapeng, Dong Yihong. Measurement of Text Similarity: A Survey. *Information*, 2020, vol. 11, no. 9, article ID: 421, 17 p. DOI: 10.3390/info11090421
4. Rani S., Singh J. Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity. *Computing, Analytics and Networks: First International Conference (ICAN 2017)*, 2018, pp. 72–80.
5. Pikies M., Ali J. Analysis and Safety Engineering of Fuzzy String Matching Algorithms. *ISA Transactions*, 2021, vol. 113, pp. 1–8.
6. Kente T., De Rijke M. Short Text Similarity With Word Embeddings. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1411–1420.
7. Mikolov T. Efficient Estimation of Word Representations in Vector Space. *arXiv: Computation and Language*, 2013. Available at: https://arxiv.org/abs/1301.3781. DOI: 10.48550/arXiv.1301.3781
8. Thada V., Jaglan V. Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm. *International Journal of Innovations in Engineering and Technology*, 2013, vol. 2, no. 4, pp. 202–205.
9. Patricoski J. et al. An Evaluation of Pretrained BERT Models for Comparing Semantic Similarity Across Unstructured Clinical Trial Texts. *Informatics and Technology in Clinical Care and Public Health*, 2022, pp. 18–21.
10. Neculoiu P., Versteegh M., Rotaru M. Learning Text Similarity With Siamese Recurrent Networks. *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016, pp. 148–157.
11. Amin K., Lancaster G., Kapetanakis S. et al. Advanced Similarity Measures Using Word Embeddings and Siamese Networks in CBR. *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference*, 2020, pp. 449–462. DOI: 10.1007/978-3-030-29513-4_32
12. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. *arXiv: Computation and Language*, 2019. Available at: https://doi.org/10.48550/arXiv.1908.10084. DOI: 10.48550/arXiv.1908.10084
13. Thefuzz – Fuzzy String Matching in Python. Available at: https://github.com/seatgeek/thefuzz
14. Pushkin A.S. *"Prostish li mne revnivye mechty..."* [Will You Forgive My Jealous Dreams...]. *Polnoe sobranie sochinenii: V 16 t.* [Complete Works: In 16 Volumes], Moscow; Leningrad, Izdatelstvo AN SSSR, 1937–1959, Vol. 2, book 1. *Stikhotvoreniia, 1817–1825. Litseiskie stikhotvoreniia v pozdneishikh redaktsiiakh* [Poems, 1817–1825. Lyceum Poems in Later Editions], 1947, pp. 300–301. (in Russian)
15. Pushkin A.S. *"Prostish li mne revnivye mechty..."* [Will You Forgive My Jealous Dreams...]. *Polnoe sobranie sochinenii: V 20 t.* [Complete Works: In 20 Volumes], Vol. 2, Book 2. *Stikhotvoreniia Kniga vtoraia (Iug, 1820–1824)* [Poems. Second Book (South, 1820–1824)], 2016, pp. 91–92. (in Russian)

Received December 19, 2024