Труды Карельского научного центра РАН. 2023. № 4. С. 16–23
Transactions of the Karelian Research Centre RAS. 2023. No. 4. P. 16–23
DOI: 10.17076/mat1773

УДК 81.32

ПОИСК ПОЧТИ ПОХОЖИХ ТЕКСТОВ В ЛИНГВИСТИЧЕСКОМ КОРПУСЕ ВЕПКАР

Ф. Ю. Быков[1], А. А. Крижановский[2*]

[1] *Институт математики и информационных технологий, Петрозаводский государственный университет (пр. Ленина, 33, Петрозаводск, Республика Карелия, Россия, 185910)*
[2] *Институт прикладных математических исследований КарНЦ РАН, ФИЦ «Карельский научный центр РАН» (ул. Пушкинская, 11, Петрозаводск, Республика Карелия, Россия, 185910), *andew.krizhanovsky@gmail.com*

При построении лингвистических корпусов разработчикам требуется очищать корпусы от текстовых дубликатов. В статье представлен небольшой обзор способов поиска почти похожих текстов в различных корпусах. Разработан алгоритм и программа поиска почти похожих текстов на основе подсчета числа общих биграмм. Проведены эксперименты на текстах Открытого корпуса вепсского и карельского языков ВепКар. Из 100 найденных программой пар наиболее похожих текстов эксперт подтвердил около половины случаев сходства. С помощью рангового расстояния Кендалла подсчитано, какая из трех рассмотренных метрик сходства текстов упорядочивает пары похожих текстов наиболее близко к экспертному. Разработанная программа и в дальнейшем будет использоваться в корпусе текстов ВепКар.

Ключевые слова: корпусная лингвистика; почти похожие тексты; ранговая корреляция Кендалла

Для цитирования: Быков Ф. Ю., Крижановский А. А. Поиск почти похожих текстов в лингвистическом корпусе ВепКар // Труды Карельского научного центра РАН. 2023. № 4. С. 16–23. doi: 10.17076/mat1773

Финансирование. Финансовое обеспечение исследований осуществлялось из средств федерального бюджета на выполнение государственного задания КарНЦ РАН (Институт прикладных математических исследований КарНЦ РАН).

F. Yu. Bykov[1], A. A. Krizhanovsky[2*]. SEARCH FOR NEAR-DUPLICATE TEXTS IN THE LINGUISTIC CORPUS VEPKAR

[1] *Institute of Mathematics and Information Technologies, Petrozavodsk State University (33 Lenin St., 185910 Petrozavodsk, Karelia, Russia)*
[2] *Institute of Applied Mathematical Research, Karelian Research Centre, Russian Academy of Sciences (11 Pushkinskaya St., 185910 Petrozavodsk, Karelia, Russia), *andew.krizhanovsky@gmail.com*

Developers of linguistic corpora need to spot and eliminate text duplicates. An overview of approaches to searching for near-duplicate texts in various corpora is presented in this article. An algorithm and a program for searching for near-

duplicate texts (based on the number of common bigrams) have been developed. Experiments were carried out with texts from the Veps and Karelian Open Corpus VepKar. The program found 100 pairs of the most similar texts and offered them to an expert, who confirmed 42 cases to be duplicates. Three metrics of text similarity were considered. The metric that was the closest to the expert's output in its pairwise text alignments was identified using Kendall's rank distance. The newly developed program will be a useful tool for editors of the VepKar text corpus.

K e y w o r d s: corpus linguistics; near-duplicate texts; Kendall rank correlation

F o r c i t a t i o n: Bykov F. Yu., Krizhanovsky A. A. Search for near-duplicate texts in the linguistic corpus VepKar. *Trudy Karel'skogo nauchnogo tsentra RAN = Transactions of the Karelian Research Centre RAS.* 2023;4:16–23. doi: 10.17076/mat1773

F u n d i n g. The studies were funded from the federal budget through state assignment to the Karelian Research Centre RAS (Institute of Applied Mathematical Research, Karelian Research Centre RAS).

Введение

Задачу поиска почти похожих текстов или нечетких дубликатов при сборе данных в Интернете для работы поисковых систем решают достаточно давно [2]. В работе [15] почти похожими (near-duplicate) считают такие тексты, которые разделяют (имеют) очень большую долю общего словаря, где словарь – это множество лемм документа. В работе [7, с. 55] тексты, имеющие общее содержание (например, плагиат, разные версии документа, аннотация), называют co-derived (буквально: *совместно или одновременно полученный из чего-либо*). Два текста являются co-derived, если (1) часть текста извлечена из другого текста или (2) оба текста содержат фрагмент третьего текста [7, с. 56].

Такая же задача поиска почти похожих текстов встает при отборе текстов на этапе создания лингвистического корпуса или в ходе его последующей очистки от дубликатов. Рассмотрим, как эта задача решалась при построении различных корпусов.

При обходе сайтов Интернета и автоматическом построении двух корпусов текстов на немецком и итальянском в 1 млрд слов авторы работы [6] полностью исключали из рассмотрения дубликаты, поскольку обычно они находились на одном сайте и содержали однотипные предупреждения, тексты лицензий и тому подобное. При постобработке экспериментально было решено выбирать из каждого текста 25 5-грамм, при этом тексты считались дубликатами, если у них совпадали хотя бы две последовательности из пяти слов [6]. В корпусе британского английского языка ukWaC похожими считались тексты, если у них также совпадали хотя бы две из 25 последовательностей 5-грамм [10, с. 49].

Отметим, что n-граммы – непрерывные подпоследовательности из n элементов, идущие внахлест. N-граммы называют *шинглами*, когда элементами являются слова. Если подпоследовательность состоит из двух слов, то ее называют биграммой, если из трех, то – триграммой, и так далее.

В веб-корпусе чешского языка объемом в 2,7 млрд слов дубликаты искали и удаляли на уровне абзацев, а не текстов [16, с. 312]. При этом авторы корпуса мирились с тем, что тексты после такой обработки будут содержать пробелы, поскольку корпус можно было скачивать только со случайно перемешанными предложениями внутри одного текста, чтобы избежать лицензионных вопросов [16, с. 314]. Для ускорения обработки выполнялось сравнение не всех пар текстов, а сравнивали n-граммы документа со всеми уже добавленными в корпус текстами. Из текста извлекались 8-граммы (последовательности из 8 слов). Абзац считался дубликатом, если он содержал более 30 % уже встреченных 8-грамм. После удаления дубликатов объем корпуса уменьшился на 20 % [16, с. 312].

Для поиска дубликатов в веб-корпусе словенского языка, содержащем 1,2 млрд токенов, была использована программа Onion [14] с параметрами: уровень порога 0,5 и последовательности 5-грамм. Дубликаты текстов удалялись полностью. Дубликаты-абзацы были оставлены, но помечены, что позволило сохранить целостность текстов [9, с. 36].

Постановка задачи

Наша задача заключается в том, чтобы найти почти похожие тексты в Откры-

том корпусе вепсского и карельского языков ВепКар (dictorpus.krc.karelia.ru). Опишем три особенности задачи.

Во-первых, относительно небольшой объем корпуса, а именно 1,5 млн слов. Поэтому неактуальны проблемы, связанные с недостатком памяти при поиске дубликатов. Однако размеры корпуса достаточно велики, чтобы искать почти похожие тексты вручную, для этого нужны программные средства.

Во-вторых, корпус создается не автоматически, как в примерах выше, а вручную. Из первых двух пунктов следует, что мы достаточно бережно относимся к текстам корпуса и не можем себе позволить удалять дубликаты без ручной проверки. Таким образом, нужно разработать алгоритм поиска почти похожих текстов, который будет выдавать кандидатов для последующей ручной работы эксперта.

В-третьих, авторы дают разрешение на включение своих текстов в корпус, поэтому тексты предоставляются пользователям сайта ВепКар полностью, что делает неприемлемым удаление дубликатов на уровне абзацев. То есть мы будем искать только нечеткие дубликаты текстов.

Обзор способов поиска почти похожих текстов

На примере 65 млн последовательностей из 8 слов (8-грамм), извлеченных из статей «Лос-Анджелес таймс», было обнаружено, что только 908 тыс. 8-грамм повторяются хотя бы раз, то есть менее 1,4 % от исходного числа n-грамм [14, с. 66].

Наблюдения такого рода изложили авторы алгоритма SPEX [7] в 2004 году. Вообще, в алгоритмах поиска дубликатов из связной и непрерывной последовательности, которую представляет собой текст, выбирается несколько фрагментов, то есть n-грамм, – это и будет *отпечаток документа* (fingerprint). Общая гипотеза алгоритмов поиска дубликатов в том, что если тексты имеют достаточно большое число общих кусочков, то маловероятно, что эти тексты появились независимо друг от друга. Идея алгоритма SPEX в том, чтобы при построении отпечатка выбирать в документе только те n-граммы, которые встречаются два и более раз [7]. То есть нет смысла для поиска похожих документов брать уникальные n-граммы, которые больше не встречаются в коллекции документов.

Недостатком алгоритма SPEX является то, что для больших корпусов, включающих более 10 млрд слов, требования к памяти становятся запредельными [14, с. 68]. В этом случае альтернативой является алгоритм поиска дубликатов с суффиксным массивом, где размер требуемой памяти является константой и не зависит от размера корпуса [14, с. 66].

В работе А. В. Крюковой [3] рассматривается поиск почти похожих текстов с помощью того же вычисления близости на основе n-грамм (Word N-Gram Containment Measure и Word N-Gram Jaccard Measure) и с помощью строковых метрик: расстояние Левенштейна, Greedy String Tiling, самая длинная общая подстрока, косинусный коэффициент [3]. В ходе ее экспериментов на небольшом корпусе русских текстов лучший результат показали метрики N-Gram Containment Measure и косинусный коэффициент [3].

- *N-Gram Containment Measure.* Обратим внимание, что эта мера, использованная в работе А. В. Крюковой [3], была взята из работы [8], в которой эта мера называется *containment*, см. далее формулу (2).

- *Косинусный коэффициент.* При переводе документов в векторное пространство (vector space model) документу соответствует вектор. Каждый элемент вектора соответствует словарному слову. Если это слово встречается в тексте, то значение элемента в векторе равно весу этого термина в тексте, вычисленному, например, по схеме TF-IDF. Таким образом, *косинусный коэффициент* сходства двух документов A и B вычисляется как косинусный коэффициент между их векторными представлениями $\overrightarrow{V}(A)$ и $\overrightarrow{V}(B)$ [12, с. 121]:

$$sim(A, B) = \frac{\overrightarrow{V}(A) \cdot \overrightarrow{V}(B)}{|\overrightarrow{V}(A)||\overrightarrow{V}(B)|}.$$

Таким образом, один из лучших результатов показала метрика сходства *containment* [3]. Эта метрика предложена в работе [8], в которой рассматриваются две метрики сравнения текстов на основе n-грамм: *containment* и *resemblance* (дословный перевод: «содержание» и «сходство»). Обе метрики дают значение от 0 до 1, чем ближе к 1, тем больше общих n-грамм есть в текстах A и B, тем более тексты похожи.

Для вычисления метрик containment и resemblance нужно разбить каждый из сравниваемых текстов на токены. Затем нужно построить n-граммы из соседних слов. Таким образом, обработанному тексту A будет соответствовать множество n-грамм $S(A, \omega)$, где ω – размер n-грамм.

Метрика resemblance сходства текстов A и B вычисляется по формуле [8, с. 23]:

$$resemblance(A, B) = \frac{|S(A, \omega) \cap S(B, \omega)|}{|S(A, \omega) \cup S(B, \omega)|}. \quad (1)$$

Метрика containment – содержание текста A в тексте B вычисляется по формуле [8, с. 23]:

$$containment(A, B) = \frac{|S(A, \omega) \cap S(B, \omega)|}{|S(A, \omega)|}. \quad (2)$$

От перестановки параметров значение метрики containment может измениться из-за знаменателя, так как тексты могут быть разной длины. Метрикой $containment_1$ будем называть содержание текста A в тексте B, т. е. $containment_1 = containment(A, B)$, а метрикой $containment_2$ – содержание текста B в A.

Алгоритм и программа поиска почти похожих текстов

Для поиска почти похожих текстов был разработан Алгоритм, вычисляющий метрики containment и resemblance на множестве текстов.

Алгоритм реализован в программе Texiclast, написанной на языке PHP[1]. Программа Texiclast извлекает тексты из базы данных корпуса ВепКар. Тексты приводятся к нижнему регистру, разбиваются на токены. По токенам с помощью библиотеки `php-text-analysis`[2] генерируются биграммы, то есть пары соседних слов (см. строки 6–7 Алгоритма).

Метрики containment и resemblance для текстов T_i и T_j вычисляются по формулам (1) и (2) в строках 8–12. Полученные значения сохраняются в строке 13.

Алгоритм. Вычисление метрик ($containment_1$, $containment_2$ и $resemblance$) сходства текстов с помощью n-грамм

```
1  Input: texts T.
2  Output: array R: (T_A, T_B,
3          similarity metrics).
4  for i ← 0; i++; i<|T| do
5  |  for j <- i+1; j++; j<|T| do
6  |  |  B_i ← bigrams(T_i)
7  |  |  B_j ← bigrams(T_j)
8  |  |  I ← |B_i ∩ B_j|
9  |  |  U ← |B_i ∪ B_j|
10 |  |  cont_1 ← I/|B_i|
11 |  |  cont_2 ← I/|B_j|
12 |  |  resemblance ← I/U
13 |  |  R ← (T_A, T_B, cont_1, cont_2, resemblance)
```

Эксперимент по поиску похожих текстов в корпусе ВепКар

Тексты корпуса ВепКар хранятся в базе данных, содержащей 4178 текстов на вепсском и карельском языках с тремя наречиями (собственно карельским, ливвиковским и людиковским). Найдем дубликаты для текстов на собственно карельском наречии.

При помощи SQL запроса "`SELECT * FROM ‘texts‘ WHERE lang_id = 4;`" было найдено 1180 текстов на собственно карельском наречии. Поле `lang_id` содержит идентификатор языка, собственно карельское наречие имеет номер 4. Результаты вычислений метрик $containment_1$, $containment_2$ и $resemblance$ для текстов на собственно карельском наречии представлены в таблице на рис. 1, строки упорядочены по убыванию значений метрики $containment_1$.

Для сравнения метрик были выбраны первые наиболее похожие 100 пар текстов, отсортированные по значению метрики $containment_1$ (см. полную таблицу [1]). Тексты, значение метрик у которых равно 1, полностью идентичны. Возьмем два текста, у которых значение метрики максимально большое, но меньше 1, то есть тексты различаются. Например, сравним тексты с идентификаторами 1652 и 2116. Программа WinMerge показывает (рис. 2), что различие заключается в одном символе: скорее всего, это опечатка.

Программа WinMerge позволяет сравнивать и объединять как отдельные файлы, так и целые папки с файлами. Программа показывает построчное сравнение двух файлов. Если строки частично совпадают, то программа WinMerge показывает посимвольное сравнение. Важным достоинством программы является визуализация сходства и различия в текстах, развитая система быстрых клавиш (Keyboard shortcut).

WinMerge – это настолько старый и тщательно разрабатываемый проект в мире открытого программного обеспечения, что тексты программы WinMerge использовались при анализе связности кода [13], при изучении эволюции внутреннего словаря программистов, отражающегося в именах переменных и комментариях [5], в задаче поиска и визуализации концептов в исходном коде программы [17].

[1]Исходный код программы сравнения текстов Texiclast доступен онлайн: `https://github.com/saimur420/Search-for-similar-texts-in-the-linguistic-corpus`.

[2]Библиотека PHP Text Analysis: `https://github.com/yooper/php-text-analysis`.

id текста 1	id текста 2	Метрики			Кол-во совпавших n-gram	Дубликат?
		Containment 1	Containment 2	Resemblance		
3495	3090	1	1	1	1614	да
1649	2113	1	1	1	1106	да
2966	3136	1	1	1	1101	да
2963	3044	1	1	1	856	да
1659	2123	1	1	1	158	да
3482	3038	1	1	1	85	да
3501	3150	1	1	1	59	да
3480	3036	1	1	1	43	да
1652	2116	0,9974	0,9974	0,9948	772	да
2118	1654	0,9973	0,9973	0,9946	1480	да
3082	3127	0,9960	0,9973	0,9933	739	да
1650	2114	0,9941	0,9926	0,9868	673	да
2641	2377	0,9933	0,9933	0,9868	298	да
3489	3067	0,9920	0,9920	0,9841	247	да
1644	2109	0,9919	0,9880	0,9801	246	да
3502	3141	0,9873	0,9873	0,9749	311	да
4042	4043	0,9866	0,9866	0,9735	147	да
3481	3037	0,9826	0,9826	0,9658	113	да
3494	3089	0,9821	0,9821	0,9649	440	да
3498	3152	0,9809	0,9809	0,9625	308	да
3497	3134	0,9776	0,9817	0,9601	697	да
3490	3070	0,9766	0,9785	0,9561	501	да
3483	3039	0,9701	0,9848	0,9559	130	да
3499	3153	0,9661	0,9661	0,9344	1966	да
3503	3138	0,9648	0,9691	0,9360	439	да
2087	2603	0,9622	0,9770	0,9409	1273	да
1058	1205	0,9606	0,9442	0,9090	609	да
1076	1233	0,9592	0,9673	0,9291	799	да
3479	3024	0,9585	0,9585	0,9204	185	да
1090	1169	0,9551	0,9635	0,9218	766	да
3496	3103	0,9495	0,9543	0,9082	376	да
3488	3063	0,9432	0,9396	0,8893	249	да
2338	3183	0,9381	0,9164	0,8641	318	да
1730	1748	0,9305	0,9457	0,8832	174	да
3477	3015	0,9179	0,9179	0,8483	179	да
1658	2122	0,8736	0,9636	0,8457	159	да
1924	1980	0,7430	0,8362	0,6486	240	нет

Рис. 1. Степень сходства текстов, вычисленная с помощью метрик $containment_1$, $containment_2$ и *resemblance*, экспертная оценка наличия текстов-дубликатов (см. полную таблицу [1])
Fig. 1. Degree of texts similarity, calculated using the metrics $containment_1$, $containment_2$ and *resemblance*, expert assessment of duplicate texts presence (see full table in [1])

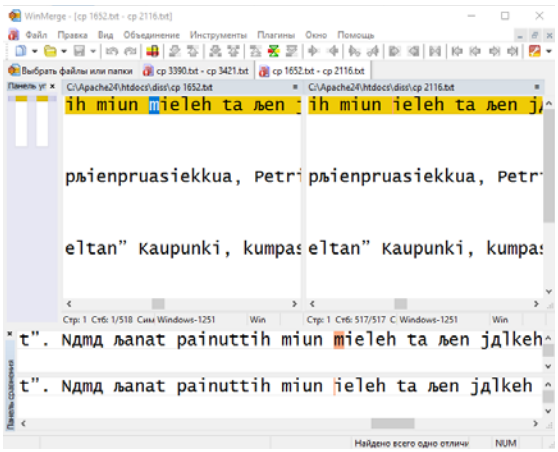

Рис. 2. Сравнение текстов 1652 и 2116 в программе WinMerge
Fig. 2. Comparison of texts 1652 and 2116 in the WinMerge program

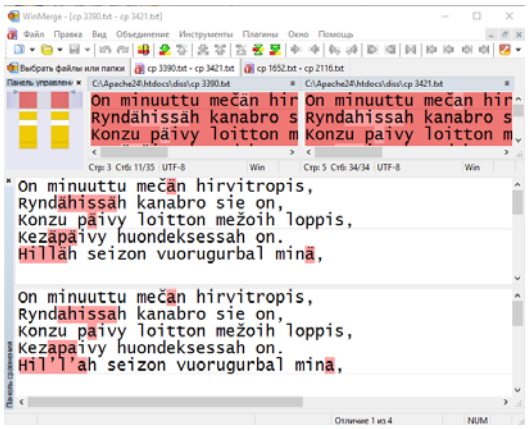

Рис. 3. Сравнение текстов ВепКар на карельском языке с идентификаторами 3390 и 3421 в программе WinMerge
Fig. 3. Comparison of VepCar texts in Karelian with identifiers 3390 and 3421 in the WinMerge program

Сравним еще одну пару текстов с идентификаторами 3390 и 3421 с меньшей степенью сходства (*resemblance* $= 0,474$). В этих текстах есть различие в написании букв «а» и «ä», но видно, что это один и тот же текст (рис. 3).

Экспертная оценка дубликатов текста

Для оценки полученных результатов были выбраны результаты сравнения 100 пар текстов с наибольшим значением метрики $containment_1$. Для удобства работы экспертов была создана таблица со ссылками на тексты в корпусе ВепКар и полями для заполнения (рис. 1). Эксперт вручную проверил найденные пары текстов и отметил дубликаты и почти похожие тексты с указанием, какой текст удалить, а какой оставить. Из 100 предложенных пар, найденных программой, эксперт нашел 42 дубликата.

Последние несколько пар текстов, например тексты 3900[3] и 3647[4] (не попали на рис. 1), – это фрагменты Евангелия от Марка и от Матфея. Это разные тексты, но они описывают один и тот же случай. Поскольку в них используются похожие фразы, то метрика *containment* показала достаточно высокое значение сходства этих текстов (0,4249).

Такой неожиданный результат нахождения сходства разных текстов Библии подсказывает, что разработанную программу Texiclast можно использовать для того, чтобы кластеризовать тексты ВепКар и выявить близкие по смыслу группы текстов в корпусе.

[3] См. http://dictorpus.krc.karelia.ru/en/corpus/text/3900.
[4] См. http://dictorpus.krc.karelia.ru/en/corpus/text/3647.

Ранговое расстояние Кендалла

Ранговый коэффициент корреляции Кендалла (а также τ-*расстояние Кендалла* или *ранговое расстояние Кендалла*) оценивает степень сходства между двумя упорядоченными наборами данных. Этот коэффициент зависит от числа таких перестановок пар объектов, которые одну последовательность (упорядоченное множество, далее кратко: *упорядочение*) преобразуют в другую [4, с. 1].

Одна из интерпретаций τ-расстояния Кендалла – это вероятность наблюдения согласованных пар (concordant, n_c – число пар) и несогласованных пар (discordant, n_d – число пар) переменных [11, с. 473]. Две переменные являются согласованными, если их порядок в двух упорядочениях сохраняется. Другими словами, τ-расстояние Кендалла показывает долю согласованностей и противоречий между двумя наборами упорядоченных данных:

$$\tau = \frac{(\text{N согл. пар}) - (\text{N несогл. пар})}{(\text{N пар})} = \frac{n_c - n_d}{\binom{n}{2}}, \text{ где} \binom{n}{2} = \frac{n(n-1)}{2}, \quad (3)$$

т. е. $\binom{n}{2}$ – количество сочетаний из n по 2, число всевозможных пар из множества размера n.

В работе [11] исследуется вопрос, насколько коррелирует ранговое расстояние Кендалла с оценками людей в задачах упорядочения информации. Применение ранговой корреляции подходит для того случая, когда есть экспертное (идеальное) упорядочение и есть упорядочения, полученные с помощью программы, которые нужно оценить [11, с. 473]. В нашем случае для 100 пар текстов есть ответ эксперта (столбец «Дубликат» на рис. 1) и значения трех метрик.

Чтобы выяснить, какая из рассмотренных метрик лучше подходит для поиска дубликатов, подсчитаем ранговую корреляцию Кендалла между значениями метрик сходства и ответами эксперта и сравним результаты. Получили, что τ-расстояние Кендалла от метрик $containment_1$, $containment_2$ и *resemblance* до ответов эксперта равно 0,427, 0,423 и 0,438 соответственно. При $n = 92$ (число рассматриваемых пар текстов)[5] значение стандартизованной оценки (z-value) для корреляции Кендалла, вычисляемое по формуле

$$\text{z-value} = \frac{3(n_c - n_d)}{\sqrt{n(n-1)(2n+5)/2}},$$

получилось соответственно 6,02, 5,97 и 6,18 с уровнем значимости $p < 0,01$.

Поясним с помощью рис. 4, что для каждой из трех метрик была получена последовательность пар текстов, упорядоченная по значению метрики. То есть на рис. 4 в первом столбце номера значений всегда упорядочены (от максимального значения сходства в первой строке), а вот оценки экспертов будут упорядочены по-разному. Мы подсчитали число согласованных (n_c) и несогласованных (n_d) пар этого упорядочения относительно ответов эксперта. Затем вычислили τ-расстояние Кендалла по формуле (3). Из этой формулы и рис. 4 видно, что для полного соответствия ($\tau = 1$) требуется такое упорядочение значений метрики $containment_1$, чтобы сначала (сверху) в столбце оценки эксперта шли только единицы, затем – только нули.

Containment 1	Эксперт	Concordant	Discordant
1	1	4	0
2	1	4	0
3	1	4	0
4	0	0	3
5	1	3	0
6	0	0	2
7	0	0	2
8	1	1	0
9	0	0	1
10	1	0	0

Рис. 4. Пример подготовки данных для вычисления τ-расстояния Кендалла между упорядочением пар текстов по значению метрики *containment* и экспертным значением сходства, где Concordant – это число согласованных пар, Discordant – число несогласованных пар

Fig. 4. An example of preparing data for calculating Kendall's τ coefficient between the ordering of texts pairs by the value of the *containment* metric and the expert similarity value, where Concordant is the number of matched pairs, Discordant – the number of mismatched pairs

На практике эксперт может вручную проверить не все, а ограниченное количество пар текстов. Поэтому, возвращаясь к формулировкам τ-расстояния Кендалла, данным в начале главы, определим его так: ранговое расстояние Кендалла (при сравнении с упорядоче-

[5] Рассматривает не 100, а 92 пары текстов, поскольку первые 8 пар текстов обладают полным сходством, см. рис. 1.

нием эксперта) позволяет оценить, насколько близко к началу списка будут находиться интересующие нас объекты – в нашем случае это пары наиболее похожих документов.

Таким образом, любая из трех рассмотренных метрик подходит для поиска почти похожих текстов, но τ-расстояние Кендалла показало несколько лучший результат для метрики *resemblance*.

Заключение

Рассмотрена актуальная задача поиска почти похожих текстов. Разработан алгоритм поиска почти похожих текстов на основе подсчета числа общих биграмм.

Проведен эксперимент на 1180 текстах на собственно карельском наречии из Открытого корпуса вепсского и карельского языков. Были найдены тексты с высокой степенью сходства. Из 100 найденных программой пар наиболее похожих текстов эксперт подтвердил около половины случаев сходства.

С помощью рангового расстояния Кендалла подсчитано, какая из трех рассмотренных метрик сходства текстов упорядочивает пары похожих текстов наиболее близко к экспертному.

Надеемся, что разработанная программа и в дальнейшем будет использоваться в корпусе текстов ВепКар.

Выражаем благодарность старшему научному сотруднику ИЯЛИ КарНЦ РАН, кандидату филологических наук Ирине Петровне Новак за выполнение экспертной оценки найденных пар текстов на карельском языке.

Литература

1. *Быков Ф. Ю.* Почти похожие тексты в ВепКар 2023 // Figshare. doi: 10.6084/m9.figshare.22134422.v1

2. *Зеленков Ю. Г., Сегалович И. В.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 9-й Всероссийской научной конференции, RCDL′2007. Переславль-Залесский, 2007. С. 166–174.

3. *Крюкова А. В.* Определение семантической близости текстов с использованием инструмента DKPro Similarity // Компьютерная лингвистика и вычислительные онтологии. Вып. 1. СПб: ИТМО, 2017. С. 87–97. doi: 10.17586/2541-9781-2017-1-87-97

4. *Abdi H.* The Kendall rank correlation coefficient // Encyclopedia of Measurement and Statistics. CA. 2007. P. 508–510.

5. *Abebe S. L., Haiduc S., Marcus A., Tonella P., Antoniol G.* Analyzing the evolution of the source code vocabulary // 13th European Conference on Software Maintenance and Reengineering. 2009. P. 189–198. doi: 10.1109/CSMR.2009.61

6. *Baroni M., Kilgarriff A.* Large linguistically-processed web corpora for multiple languages // EACL′06: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics. Trento, Italy, 2006. P. 87–90. doi: 10.3115/1608974.1608976

7. *Bernstein Y., Zobel J.* A scalable system for identifying co-derivative documents // String Processing and Information Retrieval Symp. Springer, 2004. P. 55–67.

8. *Broder A. Z.* On the resemblance and containment of documents // Proceedings of the Compression and Complexity of Sequences. June 1977. P. 21–29. doi: 10.1109/SEQUEN.1997.666900

9. *Erjavec T., Ljubešić N., Logar N.* The slWaC corpus of the Slovene Web // Informatica. 2015. Vol. 39, no. 1. P. 35–42.

10. *Ferraresi A., Zanchetta E., Baroni M., Bernardini S.* Introducing and evaluating ukWaC, a very large web-derived corpus of English // Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google. 2008. P. 47–54.

11. *Lapata M.* Automatic evaluation of information ordering: Kendall's tau // Computat. Linguist. 2006. Vol. 32, iss. 4. P. 471–484.

12. *Manning C. D., Raghavan P. Schutze H.* An introduction to information retrieval. Cambridge: Cambridge Univ. Press, 2009. URL: https://nlp.stanford.edu/IR-book (дата обращения: 27.05.2023).

13. *Marcus A., Poshyvanyk D.* The conceptual cohesion of classes // 21st IEEE International Conference on Software Maintenance. 2005. P. 133–142.

14. *Pomikalek J.* Removing boilerplate and duplicate content from Web Corpora: PhD thesis. Masaryk University, Faculty of Informatics. Brno, 2011.

15. *Potthast M., Stein B.* New issues in near-duplicate detection // Data Analysis, Machine Learning and Applications. Springer, 2008. P. 601–609.

16. *Spoustová J., Spousta M.* A High-quality Web corpus of Czech // Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC′12). 2012. P. 311–315.

17. *Xie X., Poshyvanyk D., Marcus A.* Support for static concept location with sv3D // 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis. 2005. P. 1–6.

References

1. *Bykov F. Yu.* Near-duplicate texts in VepKar 2023. *Figshare*. doi: 10.6084/m9.figshare.22134422.v1 (In Russ.)

2. *Zelenkov Yu. G., Segalovich I. V.* Comparative analysis of near-duplicate detection methods of Web documents. *Digital libraries: advanced methods and technologies, digital collections, RCDL 2007.* Pereslavl-Zalessky; 2007. P. 166–174. (In Russ.)

3. *Kriukova A. V.* Computing semantic similarity of russian texts by means of DKPro Similarity tool. *Computational Linguistics and Intellectual Technologies: Proceedings of the Annual conference 'Dialogue'.* Iss. 1. St. Petersburg; 2017. P. 87–97. doi: 10.17586/2541-9781-2017-1-87-97 (In Russ.)

4. *Abdi H.* The Kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA.* 2007. P. 508–510.

5. *Abebe S. L., Haiduc S., Marcus A., Tonella P., Antoniol G.* Analyzing the evolution of the source code vocabulary. *13th European Conference on Software Maintenance and Reengineering.* 2009. P. 189–198. doi: 10.1109/CSMR.2009.61

6. *Baroni M., Kilgarriff A.* Large linguistically-processed web corpora for multiple languages. *In EACL'06: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics.* Trento, Italy; 2006. P. 87–90. doi: 10.3115/1608974.1608976

7. *Bernstein Y., Zobel J.* A scalable system for identifying co-derivative documents. *String Processing and Information Retrieval Symp.* Springer; 2004. P. 55–67.

8. *Broder A. Z.* On the resemblance and containment of documents. *Proceedings of the Compression and Complexity of Sequences.* June 1997. P. 21–29. doi: 10.1109/SEQUEN.1997.666900

9. *Erjavec T., Ljubešić N., Logar N.* The slWaC corpus of the Slovene Web. *Informatica.* 2015;39(1):35–42.

10. *Ferraresi A., Zanchetta E., Baroni M., Bernardini S.* Introducing and evaluating ukWaC, a very large web-derived corpus of English. *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google.* 2008. P. 47–54.

11. *Lapata M.* Automatic evaluation of information ordering: Kendall's tau. *Computat. Linguist.* 2006;32(4):471–484.

12. *Manning C. D., Raghavan P. Schutze H.* An introduction to information retrieval. Cambridge: Cambridge Univ. Press; 2009. URL: https://nlp.stanford.edu/IR-book (accessed: 27.05.2023).

13. *Marcus A., Poshyvanyk D.* The conceptual cohesion of classes. *21st IEEE International Conference on Software Maintenance.* 2005. P. 133–142.

14. *Pomikalek J.* Removing boilerplate and duplicate content from Web Corpora: PhD thesis. Masaryk University, Faculty of Informatics. Brno; 2011.

15. *Potthast M., Stein B.* New issues in near-duplicate detection. *Data Analysis, Machine Learning and Applications.* Springer; 2008. P. 601–609.

16. *Spoustová J., Spousta M.* A High-quality Web Corpus of Czech. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC).* 2012. P. 311–315.

17. *Xie X., Poshyvanyk D., Marcus A.* Support for static concept location with sv3D. *3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis.* 2005. P. 1–6.

Поступила в редакцию / received: 30.04.2023; принята к публикации / accepted: 29.05.2023.
Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interest.

СВЕДЕНИЯ ОБ АВТОРАХ:

Быков Федор Юрьевич
студент

e-mail: saimur2@gmail.com

Крижановский Андрей Анатольевич
канд. техн. наук, ведущий научный сотрудник

e-mail: andew.krizhanovsky@gmail.com

CONTRIBUTORS:

Bykov, Fedor
Student

Krizhanovsky, Andrew
Cand. Sci. (Tech.), Leading Researcher