Received 01 November 2025, accepted 01 January 2026, date of publication 31 March 2026
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.25.1.4500

Variations of the CorDeGen+ Method for the Languages of Northern European Countries

YAKIV YUSYN
Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Corresponding author: Yakiv Yusyn (e-mail: yusyn@pzks.fpm.kpi.ua).

ABSTRACT This study is devoted to the problem of generating text corpora for their use during the development and testing of natural language processing information systems. The CorDeGen and CorDeGen+ methods are among the approaches that address this problem. However, as shown in this paper, the application of these methods to the development and testing of information systems for processing texts in "regional" languages (less widely spoken than English) has not yet been considered, despite its challenges. In this study, the languages of Northern Europe are considered as such "regional" languages, and the issue of removing part of the terms (if they coincide with the stop words of these languages) from the generated corpora during preprocessing is solved. To address this issue, the paper proposes seven new language variations of the CorDeGen+ method, specifically for Lithuanian, Danish, Swedish, Norwegian, Northern Sami, Lule Sami, and Icelandic languages. Latvian, Estonian, Finnish, and Southern Sami languages are also considered in this study, and the results show that the use of the CorDeGen+$^{(0\text{-}9)}$ variation, already described in the literature, is sufficient for them. The experimental verification of the effect of removing part of the terms showed that the use of the proposed language variations and CorDeGen+$^{(0\text{-}9)}$ variation prevents the removal of 20–43% of all terms from the generated corpus during preprocessing.

KEYWORDS Text Corpora; Corpora Generation; Software Engineering; Software Testing; Northern European.

I. INTRODUCTION

Today, in many fields of professional activity, information systems are actively used to solve various problems related to natural language processing, primarily texts. From a software engineering perspective, the development and testing of such systems are often associated with different challenges that are not encountered in the development and testing of other types of information systems.

One of the challenges in developing and testing information systems for natural language processing is the need for a sufficient number of diverse text data corpora that these systems must process. Even during manual testing, and when applying the simplest automated testing methodologies (such as oracle-based or table-based tests) at various levels (unit, integration, end-to-end), it is desirable not to limit testing to only a few pre-prepared corpora with predefined results. If there is a need to apply more advanced automated testing methodologies (property-based testing, metamorphic testing, intramorphic testing, etc.), then hundreds and even thousands of corpora are needed, and the only solution is to generate them on the fly. Nevertheless, so far, little attention has been paid to the issue of developing methods for generating text data corpora adapted for their use during the development and testing of information systems, despite the practical need for such methods.

Also, an additional complexity level of this problem arises if the information system is designed to process text data in different languages, not just English or other common languages. These can be different "regional" languages spoken primarily in one country or region. For example, such languages include Scandinavian languages and other languages of Northern Europe. According to available data, approximately 20 million people speak Scandinavian languages [1]. Although most of these speakers also speak English, the availability and development of information systems for their native languages are also important, including for the preservation of these languages. In this case, additional complexity arises for two main reasons. First, corpus generation methods for these languages are less common and less studied. Second, developers of such systems may not be native speakers of these languages, which is often the case in outsourced and outstaffed teams.

That is why the improvement of existing and the development of new methods of generating text data corpora for their use during the development and testing of information systems for text processing in natural "regional" languages is relevant.

II. STATE-OF-THE-ART

A large number of existing studies devoted to the construction and/or generation of text data corpora focus only on the one-time execution of this process. The properties of the methods and algorithms used in such papers for constructing and/or generating the corpus make them hardly applicable for their use as sources of text data corpora during the development and testing of information systems.

Most of the studies are devoted to the construction/generation of corpora based on large natural data, which are transformed and processed in a certain way.

In paper [2], the authors described the method of corpus generation based on historical news texts. But, since these news texts are presented in the form of images (scan copies), the proposed method includes an optical character recognition (OCR) step to convert images into text. This factor significantly reduces the applicability of this method in the development and testing of information systems: images require a lot of storage space, and the OCR step is slow enough to be used in conjunction with automated testing methodologies. Also, the quality of the OCR step depends on the language of the input data, and the quality for "regional" languages may be lower than for more common ones.

Paper [3] presented two methods for generating a large parallel corpus of texts for correcting grammatical errors. Both methods use natural texts obtained from Wikipedia as input data. This already limits the suitability of these methods in the development and testing of information systems, as they depend on the online source, and offline copies of Wikipedia require a significant amount of storage space. At the same time, the first of the presented methods uses the revision history of Wikipedia pages, which further increases the storage requirements. The second presented method, although it does not require page revision history, uses machine translation, which can also be of poor quality in the case of "regional" languages.

The method presented in [4] exhibits similar limitations because this method uses posts and comments from Facebook as input data. The corpus constructed in this paper consists of texts written primarily in the Malagasy language, which fully meets the definition of a "regional" language used in this paper. However, dependence on the online source of texts, as in study [3], restricts the use of the presented method in the development and testing of natural language processing information systems.

Similarly to the previous paper, work [5] presented a corpus for another "regional" language, namely the Kashmiri language. The constructed parallel corpus contains about thirty thousand pairs of sentences written in the English and Kashmiri languages. The method presented in this paper, which was used to construct this corpus, also utilized natural text data in the original language. This may impose restrictions on the use of this method in the development and testing of information systems.

There are many other papers dedicated to corpus construction and/or generation, similar to the above-described papers, including [6-9]. They share similar drawbacks in the context of the development and testing of information systems, like the already described papers.

Today, the methods of constructing and/or generating corpora that do not require a large amount of input natural data to obtain a corpus have also been described.

Paper [10] presented a rule-based method for obtaining a corpus of texts written in a "regional" Tunisian dialect of Standard Arabic. This set of rules can be applied to texts of any size, and a similar approach can be extended to other cases of "regional" dialects. Despite this, this method only solves the problem of a lack of a corpus written in a specific "regional" dialect when input data is available in a "standard" language and cannot be applied to individual "regional" languages.

The method presented in [11] uses machine learning to further generate question-answer corpora. Despite the fact that a neural network requires a significant amount of input data for training, it can later be used to generate corpora from a set of texts of any size. At the same time, several "question-answer" pairs can be obtained from one input text, which will be included in the final corpus. Nevertheless, it is necessary to pay attention to the fact that this method still requires a large amount of natural data – in this case, during the training phase– which can be complicated in the case of "regional" languages. Additionally, this method focuses solely on generating corpora specialized for a single task in natural language processing.

There are also methods for generating corpora of text data, which are specialized for solving software engineering problems during the creation of information systems. These methods consider the specific requirements imposed on them by their use in the development and testing of information systems, such as determinism, the minimum number of input parameters, the predictability of the corpus structure, etc. We can consider such methods to be the "state-of-the-art" that best suits the context of this study.

The CorDeGen method [12] is one of such methods and requires only a single parameter: the number of unique terms in the generated corpus. The number of texts in the corpus is then derived as the fourth root of this number, rounded down to an integer. To obtain a string representation of each of the terms, the CorDeGen method uses a hexadecimal representation of the ordinal index of that term. This allows for the minimum number of input parameters and the ease of generating thousands of different corpora. Next, the received term representation is written to certain texts in calculated quantities, while this procedure is deterministic – for the same input value, the texts will always be the same.

The CorDeGen method is language-independent, so, in theory, it can be used in the development and testing of information systems for text processing in any language, including "regional" ones. However, the method does not consider how the generated corpora will be processed further, so, in practice, various problems may arise with the generated corpora. One such problem is the removal of some terms as stop words during processing.

The CorDeGen+ method [13] addresses this deficiency by introducing a Boolean function $v(x)$ to check each term's string representation for admissibility, where the value 0 means the prohibition of this representation and 1 means that this representation is allowed. For forbidden string representations, their regeneration is performed. The function $v(x)$ for each natural language has its own form, to account for stop words (forbidden representations) specific to each language. Implementations of the CorDeGen+ method with different functions $v(x)$ for different languages are referred to as language variations in [13]. Study [13] described language

variations only for the most common European languages: English, German, French, and Italian. In addition to these variations, a basic variation was also presented, which prohibits only those terms that coincide with decimal digits. This variation, according to [13], can be used for languages based on the non-Latin alphabet, including many "regional" languages. But other "regional" variations were not described in [13].

Table 1 summarizes the main properties of methods from reviewed papers, such as applicability to "regional" languages. A "Yes" answer is considered the best for each property, and a "Partially" answer indicates limited fulfilment of the property or that a "Yes" answer can only be achieved with some additional restrictions.

Table 1. The "state-of-the-art" summary

Paper	Does not require a big volume of natural data	Applicable to "regional" languages	Supports different NLP tasks	Applicable in Soft. Eng. context
[2]	Partially	Partially	Yes	No
[3]	No	Partially	No	No
[4]	No	Yes	Yes	No
[5]	Partially	Yes	Yes	Partially
[10]	Yes	Partially	Yes	Partially
[11]	Partially	Yes	No	Yes
[12]/[13]	Yes	Partially	Yes	Yes

Based on the above description and the summary in Table 1, we can conclude that most of the existing methods of construction and/or generation of corpora are not applicable in the context of the development and testing of information systems for processing texts, especially written in "regional" languages. For the majority of the considered methods, this is due to the need for a large amount of natural data used as input or at the training stage of machine learning models. The best methods are developed to address specific requirements when applied to solve software engineering problems, such as CorDeGen [12] and its modification CorDeGen+ [13], which is oriented to consider the features of different languages. However, this modified method also lacks described variations for "regional" languages.

All this suggests that it is advisable to conduct a study on the development of new variations of existing methods for generating text data corpora that will work in conjunction with "regional" oriented information systems for text processing.

III. MATERIALS AND METHODS

A. OBJECTS AND HYPOTHESIS

The main goal of this study is to improve the processes of development and testing of information systems for processing texts in "regional" languages by developing variations of the existing methods of generating text data corpora for these languages.

The objects of this study are:

- the process of generating corpora of text data using the CorDeGen method;
- the process of preprocessing generated corpora with natural language processing tasks.

The main simplifications in this study are:

- consideration of only one group of "regional" languages (languages of the Northern European region);
- consideration of only one text corpora generation method (CorDeGen+).

The main hypothesis of this study consists of the following assumptions:

- a significant part of the terms is removed from the corpora generated by the CorDeGen method when considered as being written in Northern European languages, due to their coincidence with stop words;
- it is possible to avoid this negative impact by developing variations of the existing CorDeGen+ method for these languages.

B. GEOGRAPHIC CLASSIFICATION

To date, there is no single generally accepted classification of European countries into regions.

The United Nations M49 coding classification [14] classifies the following countries as Northern European countries: Denmark, Estonia, Finland, Iceland, Ireland, Latvia, Lithuania, Norway, Sweden, and the United Kingdom (together with dependent territories).

The World Factbook [15] identifies the largest number of regions in Europe: 7. This source includes the following countries in the northern region: Denmark, Iceland, Finland, Norway, and Sweden (together with dependent territories).

This study uses the official classification of the European Union [16], which includes the following countries in Northern Europe: Estonia, Latvia, Lithuania, Denmark, Finland, Iceland, Norway, and Sweden.

C. STOP WORDS COLLECTIONS

The official languages of the countries specified in the III.B subsection are Latvian, Lithuanian, Estonian, Finnish, Danish, Swedish, Norwegian, the Sami language group, and Icelandic.

This study uses the "Stopwords ISO" collection [17] as a main source of stop words for the above-mentioned languages. The "Stopwords ISO" collection contains stop word lists for the following languages: Latvian [18], Lithuanian [19], Estonian [20], Finnish [21], Danish [22], Swedish [23], and Norwegian [24].

The "Stop word Sami" project [25] is used as a source of stop words for Sami languages, namely, Northern Sami, Southern Sami, and Lule Sami.

The stop words list, extracted from the Icelandic Gigaword Corpus [26], is used as a source of stop words for the Icelandic language. All sublists are used except abbreviations and typos/OCR errors.

Since the CorDeGen and CorDeGen+ methods use the hexadecimal representation of term indices as terms, they can only contain digits 0–9 and Latin letters a–f. Therefore, when developing CorDeGen+ variations for selected Northern European languages, only stop words composed entirely of Latin letters a–f need to be considered for filtering. Other stop words from these lists may not appear in the generated corpus and can be ignored by these variations.

D. HARDWARE AND SOFTWARE

All experiments conducted in this study were run on a physical machine whose characteristics are listed in Table 2.

Table 2. Physical machine characteristics

Characteristic	Value
CPU	Intel Core i7-9750H 2.6 GHz (6 physical / 12 logical cores)
RAM	16 GB (2667 MHz)
OS	Windows 10 (22H2)

The .NET 8 [27] and C# 12 [28] are used for the software

implementation of all developed variations of the CorDeGen+ method. The developed software was run using the .NET 8.0.6 runtime.

In this study, unit tests are used to check the correctness of the developed implementations of language variations of the CorDeGen+ method. Given the relatively simple logic of all variations, oracle-based testing and table-based testing (as an extension of oracle-based tests) methodologies are sufficient for correctness testing. To implement these methodologies, the xUnit library [29, 30] was used, which provides the corresponding [Fact] and [Theory] attributes.

IV. RESULTS

A. IMPACT OF STOP WORDS REMOVAL

The impact of removing stop words was investigated using the developed software. For each Northern European language, terms that coincided with stop words were removed from the corpus generated by the basic CorDeGen method. The number of terms removed from the corpus was recorded separately for each language. This procedure was performed for corpus sizes from 256 to 14956 unique terms, in steps of 100.

Fig. 1 shows a graph constructed for the obtained data, where:

- the abscissa axis indicates all sizes of the corpus for which the removal of terms was measured;
- the ordinate axis shows the percentage ratio of the number of removed terms to the total number of terms in the corpus (the axis is shown in the range of values from 20% to 45%);
- the minimum and maximum values among all Northern European languages were used as data series.

Table 3 shows a slice of the obtained data for each Northern European language (identified by its ISO 639-1 [31] or ISO 639-3 [32] code in the former absence) with a step of 800.

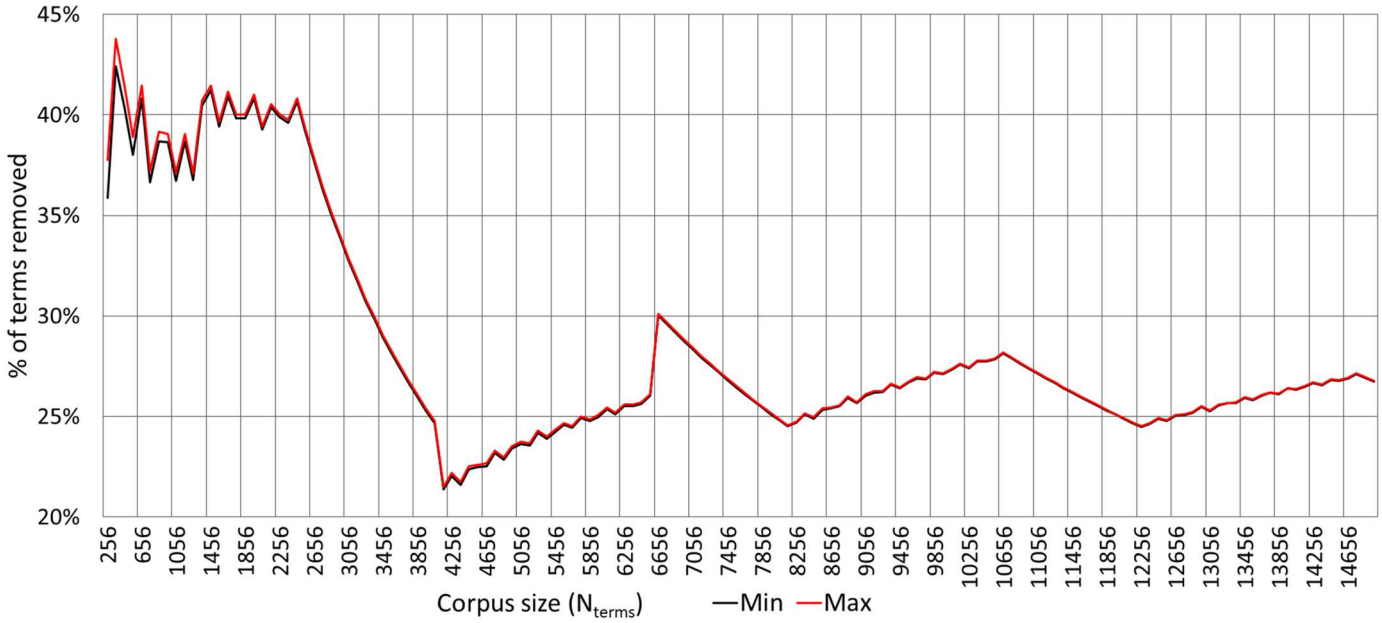

Figure 1. Minimum & maximum percentage of terms removed

Table 3. Slice of the obtained data

Nterms	LV	LT	ET	FI	DA	SV	NO	SE	SMA	SMJ	IS
256	35.88%	36.35%	35.88%	35.88%	37.75%	36.82%	36.82%	36.35%	35.88%	37.29%	37.75%
1056	36.72%	36.72%	36.72%	36.72%	37.07%	36.98%	36.94%	36.81%	36.72%	37.11%	36.98%
1856	39.83%	39.90%	39.83%	39.83%	40.01%	39.89%	39.89%	39.84%	39.83%	39.94%	39.98%
2656	37.70%	37.72%	37.70%	37.70%	37.85%	37.77%	37.78%	37.76%	37.70%	37.79%	37.84%
3456	28.97%	28.98%	28.97%	28.97%	29.08%	29.02%	29.03%	29.01%	28.97%	29.04%	29.08%
4256	22.05%	22.09%	22.05%	22.05%	22.18%	22.13%	22.11%	22.09%	22.05%	22.15%	22.18%
5056	23.63%	23.66%	23.63%	23.63%	23.74%	23.69%	23.67%	23.66%	23.63%	23.70%	23.73%
5856	24.77%	24.80%	24.77%	24.77%	24.87%	24.83%	24.81%	24.80%	24.77%	24.84%	24.86%
6656	30.03%	30.04%	30.03%	30.03%	30.09%	30.07%	30.06%	30.06%	30.03%	30.08%	30.10%
7456	26.81%	26.82%	26.81%	26.81%	26.86%	26.85%	26.84%	26.83%	26.81%	26.85%	26.87%
8256	24.69%	24.70%	24.69%	24.69%	24.73%	24.72%	24.72%	24.71%	24.69%	24.73%	24.75%
9056	26.05%	26.05%	26.05%	26.05%	26.09%	26.08%	26.07%	26.06%	26.05%	26.08%	26.10%
9856	27.17%	27.18%	27.17%	27.17%	27.21%	27.20%	27.19%	27.19%	27.17%	27.20%	27.22%
10656	28.16%	28.16%	28.16%	28.16%	28.20%	28.17%	28.18%	28.16%	28.16%	28.19%	28.19%
11456	26.19%	26.19%	26.19%	26.19%	26.23%	26.20%	26.21%	26.20%	26.19%	26.22%	26.23%
12256	24.48%	24.48%	24.48%	24.48%	24.51%	24.49%	24.50%	24.49%	24.48%	24.51%	24.51%
13056	25.28%	25.28%	25.28%	25.28%	25.31%	25.29%	25.30%	25.28%	25.28%	25.30%	25.31%
13856	26.13%	26.13%	26.13%	26.13%	26.16%	26.14%	26.15%	26.13%	26.13%	26.15%	26.16%
14656	26.88%	26.89%	26.88%	26.88%	26.92%	26.89%	26.90%	26.89%	26.88%	26.90%	26.92%

B. DEVELOPED VARIATIONS OF THE CORDEGEN+ METHOD

The CorDeGen+$^{(LT)}$ variation for the Lithuanian language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [19] stop word list. This list contains only one term which consists of only a–f letters and which can be generated by the CorDeGen method: be. So, the function *v(x)* of the CorDeGen+$^{(LT)}$ variation can be given as follows:

- If the hexadecimal representation *x* is "be", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(DA)}$ variation for the Danish language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [22] stop word list. This list contains four terms that can be generated by the CorDeGen method (consist of only a–f letters): ad, af, da, and de. So, the function *v(x)* of the CorDeGen+$^{(DA)}$ variation can be given as follows:

- If the length of the hexadecimal representation *x* is 2 and it is "ad", "af", "da", or "de", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(SV)}$ variation for the Swedish language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [23] stop word list. This list contains two terms that consist of only a–f letters and that can be generated by the CorDeGen method: de and e. So, the function *v(x)* of the CorDeGen+$^{(SV)}$ variation can be given as follows:

- If the hexadecimal representation *x* is "de" or "e", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(NO)}$ variation for the Norwegian language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [24] stop word list. This list contains two terms that can be generated by the CorDeGen method (consist of only a–f letters): da and de. So, the function *v(x)* of the CorDeGen+$^{(NO)}$ variation can be given as follows:

- If the hexadecimal representation *x* is "da" or "de", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(SE)}$ variation for the Northern Sami language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [25] stop word list. This list of 25 terms contains only one term which consists of only a–f letters and which can be generated by the CorDeGen method: de. So, the function *v(x)* of the CorDeGen+$^{(SE)}$ variation can be given as follows:

- If the hexadecimal representation *x* is "de", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(SMJ)}$ variation for the Lule Sami language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [25] stop word list. This list of 100 terms contains three terms that can be generated by the CorDeGen method (consist of only a–f letters): e, da, and de. So, the function *v(x)* of the CorDeGen+$^{(SMJ)}$ variation can be given as follows:

- If the length of the hexadecimal representation *x* is 2 and it is "e", "da", or "de", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

The CorDeGen+$^{(IS)}$ variation for the Icelandic language is developed based on the CorDeGen+$^{(0-9)}$ variation with the inclusion of selected terms from the [26] stop word list. This list contains eight terms that consist of only a–f letters and that can be generated by the CorDeGen method: aa, af, ea, ed, aaa, abba, cafa, and daff. Due to the relatively large number of stop words, the function *v(x)* can be represented in different ways (including decimal numbers), with different ways and degrees of optimization of the check. The simplest option is as follows:

- If the length of the hexadecimal representation *x* is 2 and it is "aa", "af", "ea", or "ed", then it is forbidden.
- If the length of the hexadecimal representation *x* is 4 and it is "abba", "cafa", or "daff", then it is forbidden.
- If the hexadecimal representation *x* is "aaa", then it is forbidden.
- If the hexadecimal representation *x* contains only decimal digits, then it is forbidden.
- Otherwise, it is allowed.

Latvian, Estonian, Finnish, and Southern Sami do not require separate CorDeGen+ variations, as an analysis of their stop word lists (list [18] – 161 stop words, list [20] – 35 stop words, list [21] – 847 stop words, list [25] – 99 stop words) shows that none of these words can be generated by the basic CorDeGen method. For these languages, the existing CorDeGen+$^{(0-9)}$ variation described in [13] is sufficient, since in this case only decimal digits should be forbidden during generation.

V. DISCUSSION OF RESULTS

The obtained results of measuring the number of terms removed as stop words during the preprocessing of the generated corpus as written in a Northern European language confirm the significant amount of terms affected by this process. The maximum value of the percentage of terms removed is achieved when the number of unique terms is 350, reaching 43.77% of all terms (see Fig. 1). This value begins to decrease sharply as the corpus size increases only after the mark of about 2600 unique terms, before that it does not fall below 37% of terms (see Fig. 1, Table 3). The value then fluctuates in the range of 25–30% with a decreasing trend; however, as the extrapolation results show, it does not exceed the 20% mark for any corpus sizes that make practical sense (up to 30000 unique terms; see Fig. 1, Table 3). At the same time, these data are similar for all Northern European languages: on small corpora, the difference between the values for different languages reaches 1.87%, then rapidly decreases to the values of 0.03–0.04% (see Table 3). It should be noted that the maximum value of the percentage of terms removed is

almost always observed for the Icelandic language, which is associated with the largest list of stop words.

Thus, the first assumption of the main hypothesis of this study can be considered confirmed.

The removal of such a large part of corpus terms significantly affects the results of testing information systems for natural language processing for the Northern European region, due to the distortion of the expected corpus structure (distribution of terms between texts). For example, the clustering results of such a corpus will be significantly different from those of the original undistorted corpus, as was already shown for some other European languages with comparable stop word amounts in [13]. In such a case, it makes it much more difficult (to the point of being impractical or impossible) to use any testing methodologies (oracle-based, property-based, metamorphic, etc.) because expected results, properties, metamorphic relations, etc. must be adjusted to account for the changed corpus structure. This also applies to other text processing tasks, such as classification, sentiment analysis, and data mining.

The developed seven new variations of the CorDeGen+ method for Northern European languages are based on the existing variation CorDeGen+$^{(0\text{-}9)}$, and additionally cover from one (CorDeGen+$^{(LT)}$, CorDeGen+$^{(SE)}$) to eight (CorDeGen+$^{(IS)}$) specific stop words. For the other four Northern European languages under consideration, it is sufficient to use the CorDeGen+$^{(0\text{-}9)}$ variation. The use of these variations allows us to reduce the part of the terms that will be removed as stop words during preprocessing to 0% and thus avoid a negative impact on the expected results because the structure of the corpus will remain the same and coincide with the corpus generated by the basic CorDeGen method. This result (0% removed terms) is independent of the corpus size and holds for any input value of the number of unique terms.

Thus, the second assumption of the main study hypothesis can also be considered confirmed.

There are no publicly available lists of stop words for other official and unofficial languages of the Northern European region (for example, the other living languages of the Sami group, namely Inari Sami, Skolt Sami, Kildin Sami, Ter Sami, Pite Sami, and Ume Sami). So, it is possible to assume that any information system for processing texts in such languages will not contain them either. Therefore, for such languages, it is also possible to use the existing variation CorDeGen+$^{(0\text{-}9)}$ to avoid removing terms that coincide with decimal numbers.

VI. CONCLUSIONS

This study shows the need to improve the CorDeGen method for its application during the development and testing of information systems for processing texts in "regional" languages, namely the official languages of Northern European countries.

The result of this research is the development of seven new variations of the CorDeGen+ method for the following official languages of the Northern European countries (according to the classification of the European Union): Lithuanian, Danish, Swedish, Norwegian, Northern Sami, Lule Sami, and Icelandic. Also, for the other four languages (Latvian, Estonian, Finnish, and Southern Sami), the suitability of the existing variation CorDeGen+$^{(0\text{-}9)}$ is shown, which is the basis for all other existing variations and variations developed within the framework of this study.

Using the developed language variations of the CorDeGen+ method avoids removing terms similar to stop words from the generated corpora during preprocessing, reducing the proportion of removed terms from 25–44% (depending on the corpus size) to 0%. This makes it possible to avoid distortion of the expected results of the processing of the generated corpus, in particular, during the development and testing of information systems for processing texts in Northern European languages.

These results validate the practical applicability of the developed variations and confirm the relevance of further research in a similar direction for other "regional" languages that are based on the Latin alphabet.

References

[1] A. Holmberg and C. Platzack, *The Scandinavian languages*, in: G. Cinque, R. S. Kayne (Eds.), The Comparative Syntax Handbook, 2012, pp. 420-458.

[2] K. Tanaka, C. Chu and T. Kajiwara, "Corpus Construction for Historical Newspapers: A Case Study on Public Meeting Corpus Construction Using OCR Error Correction," *SN Computer Science*, vol. 3, 2022. https://doi.org/10.1007/s42979-022-01393-6.

[3] J. Lichtarge, "Corpora generation for grammatical error correction," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 3291-3301. https://doi.org/10.18653/v1/N19-1333.

[4] F. Rakotomalala, A. R. Hajalalaina and M. V. Ravonimanantsoa Ndaohialy, "FLICs (Facebook Language Informal Corpus): a novel dataset for informal language," *International Journal of Data Science and Analytics*, vol. 18, pp. 393-403, 2024. https://doi.org/10.1007/s41060-023-00460-2.

[5] S. M. U. Qumar, M. Azim and S. M. K. Quadri, "Addressing the data gap: building a parallel corpus for Kashmiri language," *International Journal of Information Technology*, 2024. https://doi.org/10.1007/s41870-024-01979-8.

[6] K. Meden, T. Erjavec and A. Pančur, "Slovenian parliamentary corpus siParl," *Language Resources and Evaluation*, 2024. https://doi.org/10.1007/s10579-024-09746-8.

[7] Š. Arhar Holdt and I. Kosem, "Šolar, the developmental corpus of Slovene," *Language Resources and Evaluation*, 2024. https://doi.org/10.1007/s10579-024-09758-4.

[8] D. Vitório, E. Souza and L. Martins, "Building a relevance feedback corpus for legal information retrieval in the real-case scenario of the Brazilian Chamber of Deputies," *Language Resources and Evaluation*, 2024. https://doi.org/10.1007/s10579-024-09767-3.

[9] G. Recski, E. Iklódi and B. Lellmann, "BRISE-plandok: a German legal corpus of building regulations," *Language Resources and Evaluation*, 2024. https://doi.org/10.1007/s10579-024-09747-7.

[10] R. Boujelbane, M. Ellouze Khemekhem and L. Belguith, "Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora," *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 419-428. https://aclanthology.org/I13-1048.

[11] C. Alberti, D. Andor, E. Pitler, J. Devlin and M. Collins, "Synthetic QA Corpora Generation with Roundtrip Consistency," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6168-6173. https://doi.org/10.18653/v1/P19-1620.

[12] Y. Yusyn and T. Zabolotnia, "Text data corpora generation on the basis of the deterministic method," *KPI Science News*, vol. 2021, no. 3, pp. 38-45, 2021. http://scinews.kpi.ua/article/view/240780. (in Ukrainian).

[13] Y. Yusyn and N. Rybachok, "Improvement of the deterministic method of the text data corpora generation," *Herald of Khmelnytskyi National University. Technical sciences*, vol. 333, no. 2, p. 437–445, 2024. https://doi.org/10.31891/2307-5732-2024-333-2-69.

[14] United Nations Statistics Division, *Standard Country or Area Codes for Statistical Use (M49)*, 1999.

[15] Central Intelligence Agency, *The World Factbook 2021*, Washington, DC, 2021.

[16] Publications Office of the European Union, *Northern Europe*, 2024, [Online]. Available at: https://op.europa.eu/s/zXfh.

[17] D. Gene, *stopwords-iso*, 2024, [Online]. Available at: https://github.com/stopwords-iso.

[18] D. Gene, *stopwords-iso/stopwords-lv*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-lv.

[19] D. Gene, *stopwords-iso/stopwords-lt*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-lt.
[20] D. Gene, *stopwords-iso/stopwords-et*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-et.
[21] D. Gene, *stopwords-iso/stopwords-fi*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-fi.
[22] D. Gene, *stopwords-iso/stopwords-da*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-da.
[23] D. Gene, *stopwords-iso/stopwords-sv*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-sv.
[24] D. Gene, *stopwords-iso/stopwords-no*, 2016, [Online]. Available at: https://github.com/stopwords-iso/stopwords-no.
[25] E. Klem, *eklem/stopword-sami*, 2020, [Online]. Available at: https://github.com/eklem/stopword-sami.
[26] S. Friðriksdóttir, *rmh_filters*, 2021, [Online]. Available at: https://github.com/steinunnfridriks/rmh_filters.
[27] Microsoft, *What's new in .NET 8*, 2023, [Online]. Available at: https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-8/overview.
[28] Microsoft, *What's new in C# 12*, 2023, [Online]. Available at: https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-12.
[29] .NET Foundation and contributors, *Home > xUnit.net*, 2019, [Online]. Available at: https://xunit.net.
[30] .NET Foundation and contributors, *xunit/xunit at 2.6.2*, 2023, [Online]. Available at: https://github.com/xunit/xunit/tree/2.6.2.
[31] International Organization for Standardization, *Codes for the representation of names of languages—Part 1: Alpha-2 code (ISO Standard No. 639-1:2002)*, 2002.
[32] International Organization for Standardization, *Codes for the representation of names of languages – Part 3: Alpha-3 code for comprehensive coverage of languages (ISO Standard No. 639-3:2007)*, 2007.

YAKIV YUSYN, ***PhD in Software Engineering, Assistant of the Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (Kyiv, Ukraine). Research interests: software testing, natural language processing, software quality assurance.***