ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

INFORMATION TECHNOLOGY

UDC 681.518:658.519 **DOI: 10.20998/2079-0023.2021.02.10**

V. Y. SOKOL, V. O. KRYKUN, M. O. BILOVA, I. D. PEREPELYTSYA, V. V. PUSTOVAROV

TOPIC SEGMENTATION METHODS COMPARISON ON COMPUTER SCIENCE TEXTS

The demand for the creation of information systems that simplifies and accelerates work has greatly increased in the context of the rapid informatization of society and all its branches. It provokes the emergence of more and more companies involved in the development of software products and information systems in general. In order to ensure the systematization, processing and use of this knowledge, knowledge management systems are used. One of the main tasks of IT companies is continuous training of personnel. This requires export of the content from the company's knowledge management system to the learning management system. The main goal of the research is to choose an algorithm that allows solving the problem of marking up the text of articles close to those used in knowledge management systems of IT companies. To achieve this goal, it is necessary to compare various topic segmentation methods on a dataset with a computer science texts. Inspec is one such dataset used for keyword extraction and in this research it has been adapted to the structure of the datasets used for the topic segmentation problem. The TextTiling and TextSeg methods were used for comparison on some well-known data science metrics and specific metrics that relate to the topic segmentation problem. A new generalized metric was also introduced to compare the results for the topic segmentation problem. All software implementations of the algorithms were written in Python programming language and represent a set of interrelated functions. Results were obtained showing the advantages of the Text Seg method in comparison with TextTiling when compared using classical data science metrics and special metrics developed for the topic segmentation task. From all the metrics, including the introduced one it can be concluded that the TextSeg algorithm performs better than the TextTiling algorithm on the adapted Inspec test data set.

Keywords: topic segmentation, TextTiling, TextSeg, Inspec, IT Companies, computer science texts.

В. Є. СОКОЛ, В. О. КРИКУН, М. О. БІЛОВА, І. Д. ПЕРЕПЕЛИЦЯ, В. В. ПУСТОВАРОВ

ПОРІВНЯННЯ МЕТОДІВ СЕГМЕНТАЦІЇ ТЕМ ЗА ТЕКСТАМИ З КОМП'ЮТЕРНИХ НАУК

Попит на створення інформаційних систем, що спрощують і прискорюють роботу, значно зріс в умовах стрімкої інформатизації суспільства та всіх сфер діяльності. Це пов'язано з появою все більшої кількості компаній, що займаються розробкою програмних продуктів та інформаційних систем в цілому. З метою забезпечення систематизації, обробки та використання цих знань використовуються системи управління знаннями. Одним з головних завдань ІТ-компаній є постійне навчання персоналу. Для цього потрібно експортувати контент із системи управління знаннями компанії в систему управління навчанням. Основною метою дослідження є вибір алгоритму, який дозволяє вирішити задачу розмітки тексту статей, близьких до тих, що використовуються в системах управління знаннями ІТ-компаній. Для досягнення цієї мети необхідно порівняти різні методи сегментації тем на наборі даних з текстами з комп'ютерних наук. Inspec є одним із таких наборів даних, які використовуються для виділення ключових слів, і у цьому дослідженні він був адаптований до структури наборів даних, які використовуються для проблеми сегментації тем. Методи TextTiling і TextSeg були використані для порівняння деяких добре відомих показників науки про дані та конкретних показників, які стосуються проблеми сегментації тем. Також була введена нова узагальнена метрика для порівняння результатів для задачі сегментації тем. Усі програмні реалізації алгоритмів написані мовою програмування Python і представляють собою набір взаємопов'язаних функцій. Отримано результати, що демонструють переваги методу Text Seg у порівнянні з TextTiling з використанням класичних метрик науки про дані та спеціальних метрик, розроблених для завдання сегментації тем. З усіх метрик, включаючи введену, можна зробити висновок, що алгоритм TextSeg працює краще, ніж алгоритм TextTiling на адаптованому наборі тестових даних Inspec.

Ключові слова: сегментація тем, TextTiling, TextSeg, Inspec, ІТ-компанії, тексти з комп'ютерних наук.

В. Е. СОКОЛ, В. А. КРИКУН, М. А. БЕЛОВА, И. Д. ПЕРЕПЕЛИЦА, В. В. ПУСТОВАРОВ

СРАВНЕНИЕ МЕТОДОВ СЕГМЕНТАЦИИ ТЕМ НА ТЕКСТАХ ПО КОМПЬЮТЕРНЫМ НАУКАМ

Спрос на создание информационных систем, упрощающих и ускоряющих работу, значительно возрос в условиях быстрой информатизации общества и всех его сфер деятельности. Это способствует появлению все большего числа компаний, занимающихся разработкой программных продуктов и информационных систем в целом. Для обеспечения систематизации, обработки и использования этих знаний используются системы управления знаниями. Одна из основных задач ІТ-компаний - непрерывное обучение персонала. Это требует экспорта контента из системы управления знаниями компании в систему управления обучением. Основная цель исследования - выбрать алгоритм, позволяющий решить задачу разметки текста статей, близких к тем, которые используются в системах управления знаниями ІТ-компаний. Для достижения этой цели необходимо сравнить различные методы тематической сегментации в наборе данных с текстами по компьютернім наукам. Inspec - один из таких наборов данных, используемых для извлечения ключевых слов, и в данном исследовании он был адаптирован к структуре наборов данных, используемых для тематической сегментации. Методы TextTiling и TextSeg использовались для сравнения некоторых хорошо известных показателей науки о данных и конкретных показателей, которые относятся к проблеме сегментации темы. Также была введена новая обобщенная метрика для сравнения результатов для задачи сегментации тем. Все

программные реализации алгоритмов написаны на языке программирования Python и представляют собой набор взаимосвязанных функций. Были получены результаты, показывающие преимущества метода TextSeg по сравнению с TextTiling при сравнении с использованием классических метрик науки о данных и специальных метрик, разработанных для задачи тематической сегментации. По всем показателям, включая предложенный, можно сделать вывод, что алгоритм TextSeg работает лучше, чем алгоритм TextTiling на адаптированном наборе тестовых данных Inspec.

Ключевые слова: тематическая сегментация, TextTiling, TextSeg, Inspec, ИТ-компании, тексты по компьютерным наукам.

Introduction. In the context of the rapid informatization of society and all its branches, both daily and professional activities, the demand for the creation of information systems that simplifies and accelerates work has greatly increased. This need provokes the emergence of more and more companies involved in the development of software products and information systems in general. Also, at the same time, there is a development of technologies, working tools in the whole market and the accumulation of practical experience of individual IT companies. In order to ensure the systematization, processing and use of this knowledge, knowledge management systems are used.

One of the main tasks of IT companies is continuous training of personnel to improve their qualifications and ensure greater work efficiency. For this purpose, content from the company's knowledge management system must be exported to the learning management system. To export this content, it must be marked up, which means segmented into different thematic areas with a specific set of keywords. In order to segment the text into thematic sections, it is necessary to solve the problem of topic segmentation, for which some well-known methods can be used. These methods have worked well when testing, mainly on news texts. But the main topics, knowledge of which is accumulating in IT companies, is computer science.

The purpose of this work is to compare the effectiveness of some well-known topic segmentation methods on a dataset on computer science topics, which will help in the future to implement an appropriate component for a knowledge management system when preparing content for export to a learning management system.

Topic segmentation methods classification. The task of topic segmentation is to select from the text those parts that will describe only certain topics that are different in nature (the task of linear segmentation) or those parts that will be different in the degree of description detailing of a particular topic, in other words, to highlight some subtopics (hierarchical segmentation task) [1].

Topic segmentation methods classification is given on fig. 1.

Some methods use lexical cohesion or similarity metric that characterizes the similarity degree of some parts of the text, while looking for areas that are characterized by the least similarity and are perceived as segment boundaries or vice versa, group areas that are characterized by the greatest similarity using clustering [1]. Examples of these methods are:

- TextTiling (Hearst 1997) [2];
- LCSeg (Galley et al. 2005) that uses lexical chains and is applied mostly to dialogue data [3];
- some supervised classification approaches (Georgescul et al 2006) [4];
- Dot-Plotting (Reynar 1994) [5] that is the most famous one from clustering approaches.

Methods that use generative models involve the creation of a specific model that characterizes the text generated as a set of topics, which in turn generate the original vocabulary characteristic of this topic. Based on this, if it is possible to distinguish topics from the existing vocabulary in the text, you can determine the boundaries

of these existing generated topics in the same way [1]. Examples of these approaches are:

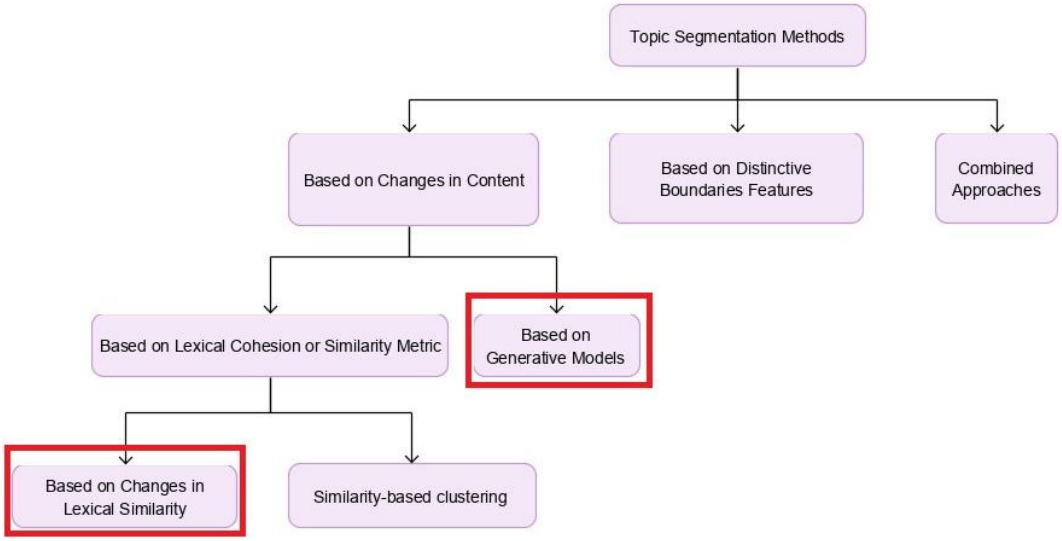

Fig. 1. Topic segmentation methods classification

- Hidden Markov Models methods (Mulbregt et al. 1999 [6]; Yamron et al. 1998 [7]);
- Latent Concept Modeling (Blei and Moreno 2001) [8], namely the Probabilistic Latent Semantic Indexing (pLSI) and Latent Dirichlet Allocation methods;
- Compact Language Modeling methods, such as TextSeg (Utiyama and Isahara 2001) [9] and its modification with LDA by Eisenstein and Barzilay (2008) [10].

Evaluation metrics. Many classification problems use a standard approach based on the calculation of precision, recall, and F-score metrics. This approach can be applied to the problem of topic segmentation, but it does not consider the degree of approximation of the correct answer from the actual segment boundaries. Therefore, in order to assess the quality of classification problems, several other approaches have been proposed [1].

The first measure was P_k, which indicates the probability of segmentation error, so the value of this measure can be obtained in the range from 0 to 1 [1, 11]. To calculate the value of P_k, the concept of a window the size of k sentences is used, which moves along the entire text. The indicative function $\delta_s(\mathrm{i,j})$ is 1 if the sentences i and j belong to the same segment, and 0 - otherwise. To determine whether there is an error between the correct segmentation R and the predicted segmentation H, the XOR operator is used, which is equal to 1 in the case of discrepancy of binary operands. Thus, the value of P_k is calculated by the following formula, where N is the number of sentences, and k is the length of the window [1, 11]:

$$P_k = \frac{\sum_{i=1}^{N-k} \delta_H(i, i+k) \oplus \delta_R(i, i+k)}{(N-k)}. \qquad (1)$$

The value of P_k has significantly improved the quality of evaluation compared to classical metrics, but there are possible situations when either false negative boundaries or false positive ones go unnoticed [1, 12]. To solve this problem, a measure WD was proposed, in which $b_H(i,j)$ determines the number of boundaries between the sentences i and j, which was provided by the algorithm. Based on this value, WD can be calculated as follows [1, 12]:

$$WD = \frac{\sum_{i=1}^{N-k}[|b(i, i+k) - b_R(i, i+k)| > 0]}{(N-k)}. \qquad (2)$$

Methods. TextTiling. One of the topic segmentation algorithms is TextTiling, which involves dividing the text into paragraphs depending on their thematic characteristics. To characterize the thematic structure, features based on the metrics of lexical co-occurrence patterns are used. The algorithm itself consists of three main stages [2]:

- tokenization (and appropriate pre-processing of the text);
- lexical score determination;
- segment boundaries identification.

The purpose of the first stage (tokenization stage) is to bring the original data set to a single format, which includes pre-processing and data filtering. Also important points of the tokenization stage are bringing all words to one register, filtering stop words (which are frequent in the language and do not have any specific thematic characteristics of the text), as well as bringing words to the original morphological form (lemmatization). At the end of the pre-processing, pseudo-sentences containing *w* words are formed in order to bring the sizes of sentences to the same values. The obtained pseudo-sentences are called token-sequences in terms of the given algorithm [2].

In the second stage of the algorithm, lexical similarity metrics are determined for each gap between the token sequences, i.e. between some blocks of text before and after the gap. There are several approaches to determining this value. In the method based on the comparison of blocks, the general similarity between the lexical characteristics of adjacent blocks is calculated. The length of the block is marked as *k* and is the number of sentences that are compared with each other and characterizes the approximate size of the topic segment. If the gap between certain token sequences is denoted as *i*, then the value *score(i)* is assigned to it. The *score(i)* value characterizes how similar the blocks are from the token sequence *i-k* to *i* and from the token-sequence *i+1* to *i+k+1*. The corresponding blocks are denoted as *b1* (*bl = {token-sequence (i-k), ..., token-sequence (i)}*) and *b2* (*b2 = {token-sequence (i+1), ..., token-sequence (i+k+1)}*). Based on this, the *score(i)* value (from 0 to 1) is calculated by the following formula, where t includes all words that were processed at the tokenization stage, except all the stop words, and $w_{t,b}$ denotes the value from the table (in this case the number of occurrences) for the word *t* in block *b* [2]:

$$score(i) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_t w_{t,b_2}^2}}. \qquad (3)$$

The last third stage of the algorithm is to determine the boundaries between the segments. For each interval, a depth score is determined, i.e. it determines how evident are the signs of topic change between the two sides to the left and right of this gap. This depth score is calculated as the sum of the differences between the left and right similarity values between the largest vertex on each side of the gap (left and right, respectively) to the value in that gap. So, the deeper valleys on the diagram get higher values of this depth score. The depth scores are sort and boundaries are determined - the higher the value, the more likely the segment boundary is in this gap [2].

Methods. TextSeg. TextSeg is one of the examples of generative methods of topic segmentation, the essence of which is the assumption that the text is generated based on a certain sequence of topics, which in turn have their own models of language, i.e. the probabilities of meeting words. In this case, having such models, segmentation is performed in such a way as to maximize the likelihood, calculated from the data from which these language models were formed. This approach does not use data to train the model, building language models directly from

the data for which segmentation is performed [1, 9].

Assume that there is some text composed of n words (W=$w_1 w_2 \dots w_n$), while the desired segmentation is expressed as S=$s_1 s_2 \dots s_m$, where m is the number of segments. Suppose that n_i is the number of words in the segment S_i and w_j^i defines the j-word in the segment S_i, then $W_i = w_1^i w_2^i \dots w_{n_i}^i$. Assume that $f_i(w_j^i)$ denotes the number of words in W_i the same as w_j^i; k – is the number of unique words in W.

To find the most likely segmentation, the cost of segmentation C(S) has to be minimized. The formula for calculating C(S) value is expressed as [9]:

$$C(S) = \sum_{i=1}^{m} c\left(w_1^i w_2^i \dots w_{n_i}^i | n, k\right) = \\ = \sum_{i=1}^{m} \sum_{j=1}^{n_i} log \frac{n_i + k}{f_i(w_j^i) + 1} + logn\,. \quad (4)$$

It is possible to use words or individual sentences as structural units for algorithm. Assume that the sentences are chosen as structural units, then g_i defines the interval between several adjacent sentences i and i+1. In this case, it is possible to determine the graph G = <V, E> (V is the set of vertices, E is the set of edges), where:

$$V = \{g_i | 0 \le i \le n\} \\ E = \{e_{ij} | 0 \le i < j \le n\} \quad (5)$$

The edge e_{ij} begins in g_i and ends in g_j, respectively.

The algorithm can be divided into 2 stages [9]:

1) calculate the value of $c_{ij} = c(w_{i+1} w_{i+2} \dots w_j | n, k)$ for all the corresponding edges e_{ij} where 0 <= i <j <= n;

2) find the least cost path between vertices g_0 and g_n.

To find the least cost path, any algorithm that solves a given problem can be used. For example, an approach based on dynamic programming, one of which is the Dijkstra algorithm can be applied [9].

Method adaptation and implementations for testing. In order to compare the algorithms according to the given metrics, it is necessary to have their software implementations, in this case for the TextTiling and TextSeg methods. The algorithm of the TextTiling method is shown on fig. 2 and for the TextSeg method on fig. 3. Models are presented in the UML activity diagram notation.

The peculiarity of the TextSeg algorithm adaptation for the problem is the process of creating a matrix of estimates and its size. The original algorithm involves the input of text and the creation of a matrix of estimates that has the size of the number of words in the text. According to this original version, the boundaries of the segments are defined, which do not have to be at the end of the sentence, because they can also be in the middle of the sentences.

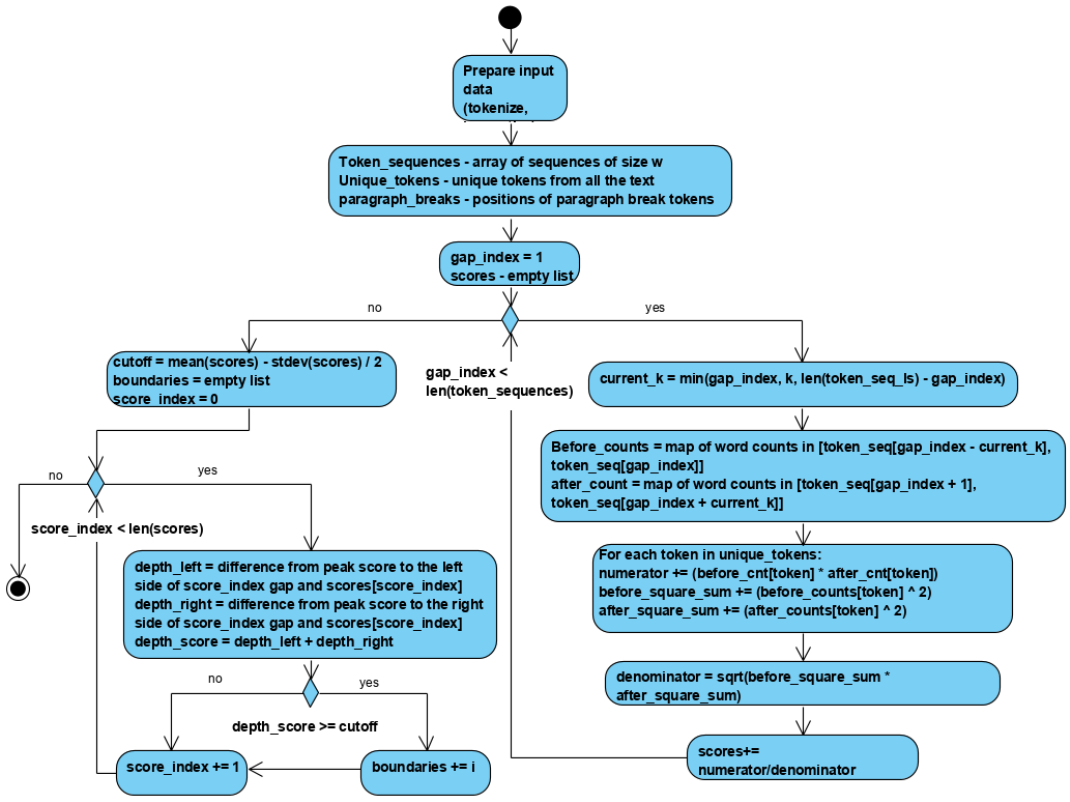

Fig. 2. TextTiling algorithm (algorithm)

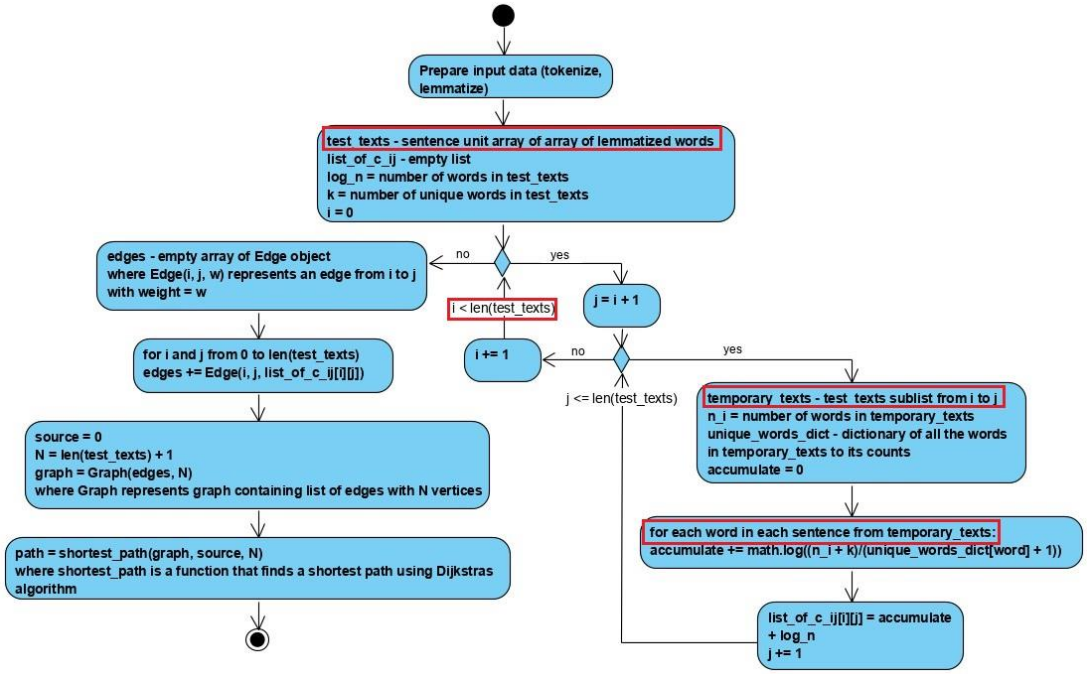

Fig. 3. TextSeg algorithm (algorithm)

Thus, in the given problem solution segment boundaries should be exclusively within the limits of the end of the previous sentence. The last available index in the rows and columns of the estimates matrix will be equal to the number of sentences fed to the input of the topic segmentation algorithm. Element c_{ij} of the estimates matrix corresponds to the segment estimate that begins before the sentence *i* beginning and ends after the sentence *j*. In the original algorithm, the process of pre-segmentation into sentences is absent, and that is why the number of words between the elements of the matrix *i* and *j* in the original algorithm is always equal to one.

It should be noted that in fig. 3 some elements are highlighted to describe points that show how this algorithm was adapted.

All software implementations of the algorithms are written in Python and represent a set of interrelated functions. In the future, the program code can be supplemented with logic that will allow deploying the component for topic text segmentation as a restful web service for its interconnection with the component of the knowledge management system.

Results. Inspec Dataset Adaptation. To test the effectiveness of these algorithms, we need to generate test data, for which the correct answers must be marked. For these purposes, in the field of topic segmentation, a large number of test data sets are used on various topics, including news broadcasts. However, for the purposes of this research, it is necessary to use data that are close to those used for organizations in the field of information technology. There are several such test data sets on the subject of computer science, which were created for the task of forming a set of keywords and phrases. For example, Inspec consists of 2,000 different abstracts of articles on computer science, taken from scientific sources, and related keywords and phrases,

Accordingly, this test data set Inspec was adapted to the problem of topic segmentation of the text. This uses an approach similar to that used by Choi in his work and many others who took the same test data set or adapted it, based on the same data combination principle to check the quality of text segmentation according to their thematic content. The data set used in Choi's work consists of artificially generated documents based on documents from the Brown corpus. This approach involves random selection of a document from the Brown corpus, then the first 3–11 sentences are taken from this document and these sentence are perceived as a segment. A combination of 10 such segments creates a single text in the Choi dataset [13].

The final data set for testing algorithms consists of fifty texts, each of which is ten segments long, each of the segments is one of the abstracts included in the Inspec. Accordingly, the boundaries of the segments are considered as a transition between one text from Inspec to another.

Two approaches to the formation of text units were used for testing. The first one is that individual sentences are the input units of the algorithm. The second one is to combine the individual sentences of each segment into paragraphs with a maximum length of three sentences. For example, if the segment consists of 7 sentences, then the first unit will be 3 sentences long, the second one will be also 3 sentences long, and the third one will be 1 sentence long.

Results. Algorithm efficiency comparison. To determine the efficiency of the algorithms, 5 metrics were used, namely precision, recall, F-score, P_k, WD. Accordingly, lower values of P_k and WD indicate better results of topic segmentation. A comparison of these

metrics for algorithms is given in table 1 for the first test data approach (without combining sentences in paragraphs) and in table 2 for the second one (with combining some sentences of the segment in paragraphs).

Table 1 – Metric values comparison for the first test data approach

Algorithm	Precision	Recall	F-score	P_k	WD
TextTiling	0.2676	0.34	0.2995	0.3462	0.3973
TextSeg	0.8526	0.74	0.7923	0.1082	0.1110

Table 2 – Metric values comparison for the second test data approach

Algorithm	Precision	Recall	F-score	P_k	WD
TextTiling	0.5662	0.6178	0.5909	0.3054	0.2848
TextSeg	0.9417	0.8133	0.8728	0.0866	0.0866

It should be noted that in table 1 and table 2 the results of TextTiling are given with the parameters w = 30, k = 5, and conservative measure for cutoff (HC). For better visualization of the results, the algorithms are compared using histograms for each metric (fig. 4–8).

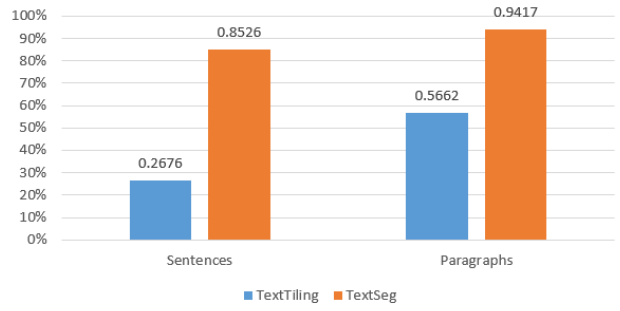

Fig. 4. Precision results comparison

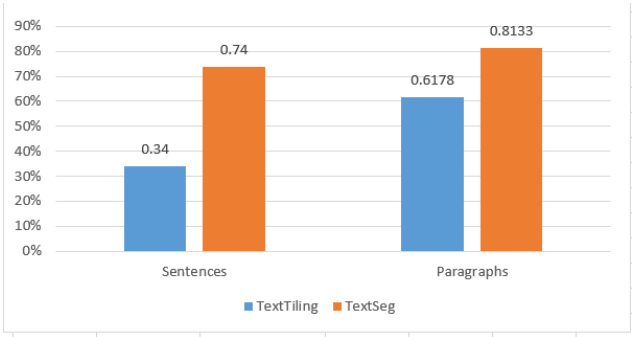

Fig. 5. Recall results comparison

In order to summarize the results, it was decided to introduce a new metric that considers each of the above, namely the F-metric in combination with the error probabilities of the classifier P_k and WD. If this metric is called T_{gen}, then its calculation can be expressed in the form given in the formula:

$$T_{gen} = \frac{\left(\frac{1}{3P_k{}'} + \frac{1}{3WD'} + \frac{F_\beta{}'}{3}\right)}{100} \quad (6)$$

where $P_k{}'$ – P_k value in percentage;

WD′ – WD value in percentage;

$F_\beta{}'$ – F_β (F-score) value in percentage.

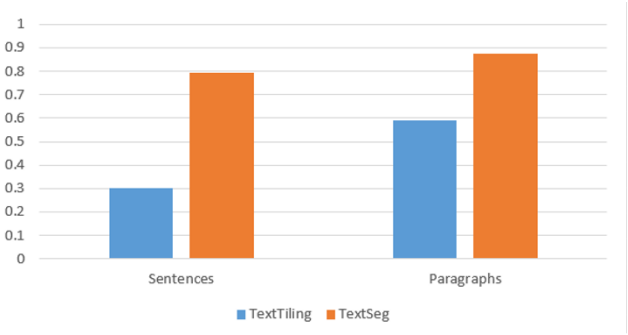

Fig. 6. F-score results comparison

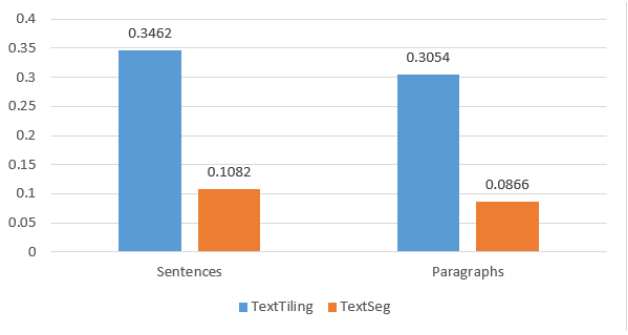

Fig. 7. Segmentation error probability (P_k) results comparison

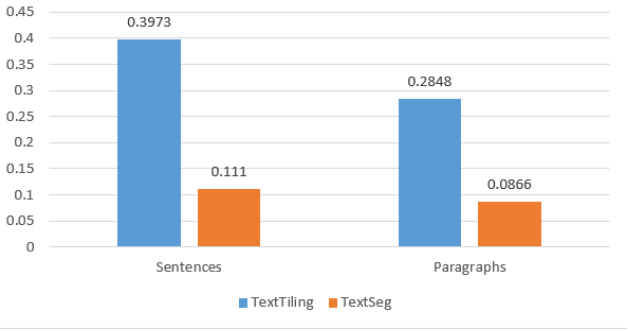

Fig. 8. Windows diff (WD) results comparison

Each of the metrics, which is generalized in T_{gen}, was given the same weighting factor. In this case, the error probability value was presented in reverse due to the fact that larger values of P_k and WD describe a worse algorithm, in contrast to the F_β value.

A comparison of this entered metric T_{gen} for algorithms is given in table 3 for two different test data approaches.

Table 3 – T_{gen} values comparison for different test data approaches

Algorithm	First test data approach (sentences level)	Second test data approach (paragraphs level)
TextTiling	0.1	0.1972
TextSeg	0.2647	0.2917

To better illustrate the difference in the indicator T_{gen}, the results of the algorithms are shown in fig. 9.

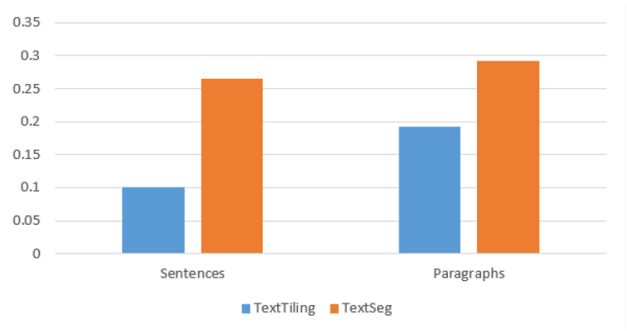

Fig. 9. T_{gen} results comparison

Therefore, from all the above metrics, including the introduced one it can be concluded that the TextSeg algorithm performs better than the TextTiling algorithm on the adapted Inspec test data set.

Conclusions. This paper includes the application of existing and well-known topic segmentation methods on computer science texts. To compare the performance of the algorithms, the Inspec dataset was adapted to a structure that is widely used in the topic segmentation problem. Results were obtained showing the advantages of the Text Seg method in comparison with TextTiling when compared using classical data science metrics and special metrics developed for the topic segmentation task.

References

1. Purver M. Topic Segmentation. *Spoken Language Understanding*. John Wiley & Sons, Ltd, Chichester, UK, 2011, P. 291–317.
2. Hearst M. A. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics.* 1997. № 23 (1). P. 33–64.
3. Galley M., McKeown K., Fosler-Lussier E., Jing H. Discourse segmentation of multi-party conversation. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003. P. 562–569.
4. Georgescul M, Clark A and Armstrong S. Word distributions for thematic segmentation in a support vector machine approach. *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLLX)*. New York City, New York, 2006. P. 101–108.
5. Reynar J. An automatic method of finding topic boundaries. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL).* Association for Computational Linguistics, LasCruces, NM. 1994. P. 331–333.
6. Mulbregt P. V., Carp I., Gillick L., Lowe S., Yamron J. Segmentation of automatically transcribed broadcast news text. *Proceedings of the DARPA Broadcast News Workshop*. Morgan Kaufmann. 1999. P. 77–80.
7. Yamron J., Carp I., Gillick L., Lowe S., van Mulbregt P. A hidden Markov model approach to text segmentation and event tracking. *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*. 1998. P. 333–336.
8. Blei D., Moreno P. Topic segmentation with an aspect hidden Markov model. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2001. P. 343–348.
9. Utiyama M., Isahara H. A Statistical Model for Domain-Independent Text Segmentation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.* 2001. P. 499–506.
10. *Eisenstein J., Barzilay R. Bayesian unsupervised topic segmentation Proceedings of the 2008 Conferenceon Empirical Methods in Natural Language Processing, Association for Computational Linguistics,Honolulu, Hawaii. 2008. P. 334–343.*
11. Beeferman D, Berger A., Lafferty JD. Statistical models for text segmentation. *Machine Learning*. 1999. № 34(1–3). P. 177–210.
12. Pevzner L and Hearst M. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*. 2002. № 28 (1). P. 19–36.
13. Choi F. Advances in Domain Independent Linear Text Segmentation *Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. P. 26–33.

References (transliterated)

1. Purver M. Topic Segmentation. *Spoken Language Understanding*. John Wiley & Sons, Ltd, Chichester, UK, 2011, pp. 291–317.
2. Hearst M. A. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics.* 1997. no 23 (1). pp. 33–64.
3. Galley M., McKeown K., Fosler-Lussier E., Jing H. Discourse segmentation of multi-party conversation. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003. pp. 562–569.
4. Georgescul M, Clark A and Armstrong S. Word distributions for thematic segmentation in a support vector machine approach. *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLLX)*. New York City, New York, 2006. pp. 101–108.
5. Reynar J. An automatic method of finding topic boundaries. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL).* Association for Computational Linguistics, LasCruces, NM. 1994. pp. 331–333.
6. Mulbregt P. V., Carp I., Gillick L., Lowe S., Yamron J. Segmentation of automatically transcribed broadcast news text. *Proceedings of the DARPA Broadcast News Workshop*. Morgan Kaufmann. 1999. pp. 77–80.
7. Yamron J., Carp I., Gillick L., Lowe S., van Mulbregt P. A hidden Markov model approach to text segmentation and event tracking. *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*. 1998. pp. 333–336.
8. Blei D., Moreno P. Topic segmentation with an aspect hidden Markov model. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2001. pp. 343–348.
9. Utiyama M., Isahara H. A Statistical Model for Domain-Independent Text Segmentation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.* 2001. pp. 499–506.
10. *Eisenstein J., Barzilay R. Bayesian unsupervised topic segmentation Proceedings of the 2008 Conferenceon Empirical Methods in Natural Language Processing, Association for Computational Linguistics,Honolulu, Hawaii. 2008. pp. 334–343.*
11. Beeferman D, Berger A., Lafferty JD. Statistical models for text segmentation. *Machine Learning*. 1999. no 34(1–3). pp. 177–210.
12. Pevzner L and Hearst M. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*. 2002. no 28 (1). pp. 19–36.
13. Choi F. Advances in Domain Independent Linear Text Segmentation *Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. pp. 26–33.

Received 13.10.2021

Відомості про авторів / Сведения об авторах / About the Authors

Сокол Володимир Євгенович – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інформаційних технологій управління; м. Харків, Україна; ORCID: https://orcid.org/0000-0002-4689-3356; e-mail: vlad.sokol@gmail.com

Крикун Віталій Олександрович – Національний технічний університет «Харківський політехнічний інститут», студент; м. Харків, Україна; ORCID: https://orcid.org/0000-0003-2576-1001; e-mail: vetall1999real@gmail.com

Білова Марія Олексіївна – кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інформаційних технологій управління; м. Харків, Україна; ORCID: https://orcid.org/0000-0001-7002-4698; e-mail: missalchem@gmail.com

Перепелиця Іван Дмитрович – кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інформаційних технологій управління; м. Харків, Україна; ORCID: https://orcid.org/0000-0001-7683-8780; e-mail: IvanPerepelytsya@gmail.com

Пустоваров Володимир Володимирович – кандидат технічних наук, начальник групи, Харківське представництво генерального Замовника - Державного космічного агентства України; м. Харків, Україна; ORCID: http://orcid.org/0000-0003-3944-5771; e-mail: Harkov11978@gmail.com

Сокол Владимир Евгеньевич – кандидат технических наук, доцент, Национальный технический университет «Харьковский политехнический институт», доцент кафедры программной инженерии и информационных технологий управления; г. Харьков, Украина; ORCID: https://orcid.org/0000-0002-4689-3356; e-mail: vlad.sokol@gmail.com

Крикун Виталий Александрович – Национальный технический университет «Харьковский политехнический институт», студент; г. Харьков, Украина; ORCID: https://orcid.org/0000-0003-2576-1001; e-mail: vetall1999real@gmail.com

Белова Мария Алексеевна – кандидат технических наук, Национальный технический университет «Харьковский политехнический институт», доцент кафедры программной инженерии и информационных технологий управления; г. Харьков, Украина; ORCID: https://orcid.org/0000-0001-7002-4698; e-mail: missalchem@gmail.com

Перепелица Иван Дмитриевич – кандидат технических наук, Национальный технический университет «Харьковский политехнический институт», доцент кафедры программной инженерии и информационных технологий управления; г. Харьков, Украина; ORCID: https://orcid.org/0000-0001-7683-8780; e-mail: IvanPerepelytsya@gmail.com

Пустоваров Владимир Владимирович – кандидат технических наук, начальник группы, Харьковское представительство генерального Заказчика – Государственного космического агенства Украины; г. Харьков, Украина; ORCID: http://orcid.org/0000-0003-3944-5771; e-mail: Harkov11978@gmail.com

Sokol Volodymyr Yevhenovych – PhD, Associate Professor, National Technical University «Kharkov Polytechnic Institute», Associate Professor of the Department of Software Engineering and Management Information Technologies; Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0002-4689-3356; e-mail: vlad.sokol@gmail.com

Krykun Vitalii Oleksandrovich – National Technical University «Kharkov Polytechnic Institute», student; Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0003-2576-1001; e-mail: vetall1999real@gmail.com

Bilova Mariia Oleksiivna – PhD, National Technical University «Kharkov Polytechnic Institute», Associate Professor of the Department of Software Engineering and Management Information Technologies; Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0001-7002-4698; e-mail: missalchem@gmail.com

Perepelytsya Ivan Dmytrovich – PhD, National Technical University «Kharkov Polytechnic Institute», Associate Professor of the Department of Software Engineering and Management Information Technologies; Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0001-7683-8780; e-mail: IvanPerepelytsya@gmail.com

Pustovarov Volodymyr Volodymyrovich – PhD, group leader, Kharkiv office of the General Customer - State Space Agency of Ukraine.; Kharkiv, Ukraine; ORCID: http://orcid.org/0000-0003-3944-5771; e-mail: Harkov11978@gmail.com