Evaluation of the Coherence of Polish Texts Using Neural Network Models

Sergii Telenyk *,†, Sergiy Pogorilyy 2,† and Artem Kramov 2,†

1 Department of Theoretical Electrical Engineering and Computer Science, Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland
2 Computer Engineering Department, Taras Shevchenko National University of Kyiv, 60 Volodymyrska Street, 01033 Kyiv, Ukraine; sdp77@i.ua (S.P.); artemkramovphd@knu.ua (A.K.)

* Correspondence: stelenyk@pk.edu.pl
† These authors contributed equally to this work.

Abstract: Coherence evaluation of texts falls into a category of natural language processing tasks. The evaluation of texts’ coherence implies the estimation of their semantic and logical integrity; such a feature of a text can be utilized during the solving of multidisciplinary tasks (SEO analysis, medicine area, detection of fake texts, etc.). In this paper, different state-of-the-art coherence evaluation methods based on machine learning models have been analyzed. The investigation of the effectiveness of different methods for the coherence estimation of Polish texts has been performed. The impact of text’s features on the output coherence value has been analyzed using different approaches of a semantic similarity graph. Two neural networks based on LSTM layers and a pre-trained BERT model correspondingly have been designed and trained for the coherence estimation of input texts. The results obtained may indicate that both lexical and semantic components should be taken into account during the coherence evaluation of Polish documents; moreover, it is advisable to analyze corresponding documents in a sentence-by-sentence manner taking into account word order. According to the retrieved accuracy of the proposed neural networks, it can be concluded that suggested models may be used in order to solve typical coherence estimation tasks for a Polish corpus.

Keywords: natural language processing; coherence evaluation; BERT model; LSTM-based neural network; Polish language

1. Introduction

The natural language processing (NLP) area incorporates different tasks that are connected with the automatic analysis of text information by utilizing the means of computer linguistics and machine learning: text generation, information extraction, speech analysis, etc. According to the growth of data arrays within the media environment, the relevance of these issues can be explained by the necessity of the processing of data arrays that should be considered as Big Data. The automatic analysis of such data volumes without human interaction allows performing actual tasks (e.g., the search for web pages, detection of fake news, voice assistants) due to a sufficient accuracy. It should be mentioned that some of these tasks fall into the category of AI-complete issues, i.e., issues that’s computational complexity is equal to the solving of the main task of Artificial Intelligence: the creation of a system that is as smart as a human. The modeling...
of textual coherence, namely, distinguishing coherent documents from incoherent ones [1], refers to this kind of task.

The term “coherence” implies the informative and semantic type of connectivity [2]; the markers of such connectivity are the semantic consistency of lexical units, repeats, synonyms, antonyms, different phrases, etc. The coherence of a text can be considered as the set of procedures that provide its cognitive integrity. Such procedures involve logical connections between cause and effect, condition and result. Moreover, the coherence provides the consistency of text data with background knowledge. Thus, the coherent document is easier to read and understand than incoherent ones. Let us consider the corresponding examples. Figure 1 demonstrates the fragment of coherent and incoherent texts.

(1) Alice has a younger brother Bob.
(2) In the morning Alice helps him get dressed.
(3) Then she takes Bob to kindergarten.

(1) Alice has a younger brother Bob.
(2) She is in the second grade.
(3) Then she takes Bob to kindergarten.

Figure 1. The example of coherent and incoherent documents.

The sentences of the coherent document are grouped around a common topic, namely, a short story about a daily routine. It seems that all sentences of the incoherent document share the same topic too. Moreover, the pairs of sentences (1–2) and (2–3) contain common coreferent objects “Alice” and “she” (corefent objects imply entities that have the same referent [3]) that provide a reader with the logical connection between the different parts of the text. However, the semantic meaning of the sentence (2) does not correspond to the story about a daily routine: it describes the main person itself. Thus, the coherence of a document cannot be achieved only by lexical means. In order to model textual coherence, it is necessary to reconcile the parts of the document according to background knowledge. Such a feature of coherent documents determines the usage of this characteristic of a text in the following tasks:

- Text generation (e.g., creation of responses for voice assistants).
- Text summarization (for instance, formulation of abstracts).
- Detection of schizophrenia symptoms (deviation from the topic of conversation) [4].

The automatic estimation of the coherence of a document requires deep analysis of semantic connectivity between the parts of a text and its structural connectivity simultaneously. In contrast to constructed and formal languages, natural languages cannot be fully interpreted as the pre-defined set of rules. Each person utilizes the means of a natural language to express his thoughts in his own way; therefore, neither the structure of a sentence nor its semantic meaning can be predicted by a certain algorithm. Thus, different machine learning techniques are widely used to reveal such logical connections within a text. According to the growth of computational power, the state-of-the-art methods utilize deep learning models (e.g., neural networks) in order to estimate the coherence of a text based on the corresponding dataset.

It should be mentioned that current approaches for coherence estimation were proposed and verified for English-language texts. However, the investigation of the coherence evaluation for the Polish corpus is still at the initial stage. Despite the belonging of both English and Polish languages to inflected languages, different approaches for the expression of syntactic relationships within a sentence are utilized. In the case of the English language (analytical language), the word order and helper words are used to build a sentence. In contrast, the addition of morphemes to a root word is widely applied in the Polish language (synthetic language) in order to build phrases and other discourse units. Like with other Slavic languages, there is a higher morpheme-to-word ratio and a lower percentage of aux-
iliary words than for analytical languages. Thus, it is advisable to verify the effectiveness of different approaches of the coherence estimation for the Polish-language corpus.

The purposes of the article are the following:

- analysis of the state-of-the-art methods of the coherence evaluation of English texts;
- investigation of the impact of lexical and semantic components on the coherence estimation of a Polish-language text;
- experimental verification of the effectiveness of different methods based on neural networks for the solving of typical coherence evaluation tasks on the Polish-language corpus.

2. Materials and Methods

2.1. Related Work

One of the first approaches with the usage of machine learning techniques called Entity Grid was proposed in the paper [5]. It was suggested to investigate transitions between the syntactic roles of entities (subject, object, no role) within adjacent sentences. Such an approach was based on the centering theory [6]. It was supposed that key entities should have a small ratio of absent roles within a text while other entities should not have any role in most cases. Firstly, the syntactic analysis of the sentence \( s \) is performed; a certain role is assigned to each entity \( e \). Then the entity grid is formed: each row corresponds to a certain sentence; each column denotes a unique entity. The cell of the grid \( c_{ij} \) contains the syntactic role of an entity \( e_j \) in a sentence \( s_i \). The feature vectors are formed according to the probability of role changes for the entities of adjacent sentences within a document \( D \):

\[
\Phi(D) = (p_1(D), p_2(D), \ldots, p_m(D)),
\]

where \( m \) is the count of allowed role transitions, \( p_i(D), i = 1...m \) denotes the probability of the local role transition between adjacent sentences according to the pre-built entity grid. The formed dataset is passed to the input of a binary classifier (support vector machines) to train this model. The output of the model interprets a prediction whether the input text is coherent or not. This idea of the investigation of the syntactic role changes was used in the further connected researches [7,8]. In contrast to the Entity Grid approach, these methods are based on the representation of a text as a graph structure. Such an approach allows taking into account long-distance relations between sentences. Moreover, the graph-based representation of a text can be used in order to reveal weak connections between the parts of the document with the further improvement of the coherence estimation value of the text.

In the paper [9], it was suggested to analyze the impact of the transition of the discourse roles of entities on the coherence estimation process. According to the Rhetorical Structure Theory [10], a coherent text can be represented as a tree structure, namely, a discourse tree (constituent tree). After the building of the discourse tree (see Figure 2), pre-order of this tree from bottom to top is performed. Each entity is assigned to multiple roles according to predefined rules. Then the mentioned entity grid is built where each cell \( c_{ij} \) contains the discourse roles of an entity \( e_j \) in a sentence \( s_i \). The formation of feature vectors is performed as it was shown in Equation (1). However, in contrast to the Entity Grid, it was suggested to take into account several roles simultaneously.
The common disadvantages of the considered methods are the following:

- Entity detection mechanism is required. Moreover, it is necessary to perform additional coreference resolution analysis in order to represent the same referents as a single object (for instance, “Microsoft” and “Microsoft Corporation”).
- The detection of such roles cannot be applied to all languages. For instance, the building of a constituent tree was proposed for the English language. However, the usage of this structure for synthetic languages (like the Polish language) is complicated due to the features of sentence building.
- Other text features like cohesive component, semantic relatedness are not taken into account. Moreover, all non-entity words are neglected despite the impact of such tokens on the representation of the semantic meaning of a text.

The semantic distributed representation of the parts of a text using pre-trained models [11–14] resulted in the creation of methods based on neural networks. Such
distributed text representation allows the processing of data by input layers with the further forwarding of a signal through other layers. In the paper [15], the recurrent and recursive neural networks were designed in order to estimate the coherence of a text. The usage of recurrent layers (simple RNN, LSTM, etc.) can be explained by the processing of input data in a recursive manner. The availability of reverse connections from the previous time step helps to simulate the processing of a text according to a reading process: word by word. The recurrent layer allows the processing of input data of unfixed length that results in the usage of this neural network in the area of NLP. As for the recursive neural network, it implies the representation of an input sentence via a binary constituent tree. An example of such a tree is shown in Figure 3. The signal forwarding is performed in a “bottom-to-top” manner.

![Figure 3. The binary constituent tree that represents a sentence “Marry was very hungry”.

The usage of either recurrent or recursive layers allows performing the vector representation of each input sequence. Then the sequence of dense (linear) layers can be applied in order to classify whether the input sequence of words is coherent or not. The idea of the usage of LSTM layers was followed in further researches [16–18]. The principal difference between all mentioned works consists of the different vector representations of a whole document. In the paper [16], it was suggested to pass all sentence vectors through an additional LSTM layer that should represent the entire text. As for the paper [17], the similarity between hidden states of the first LSTM layer (the layer that represents each sentence) was investigated. Then a convolutional operation was applied in order to reveal the main semantic components of sentences. In the paper [18], a similar approach was suggested: both LSTM and convolutional layers are applied. However, the LSTM layer is used in order to analyze the consistency of adjacent sentences while the CNN layers perform the processing of all sentences simultaneously. Thus, both local and global consistencies of text parts are taken into account.

To sum up, it is advisable to utilize an LSTM-based neural network in order to estimate the coherence of English texts. Moreover, an additional convolution operation can be applied in order to extract main components from sentences. In order to investigate the coherence estimation process for the Polish-language corpus, it is suggested to consider the different models based on both neural networks and graph-based structures.

2.2. Coherence Estimation Models for The Polish Language

2.2.1. Semantic Similarity Graph

Despite the effectiveness of the usage of neural networks to solve typical tasks, it is advisable to analyze the impact of different text features on the coherence evaluation process at first. Thus, it is suggested to consider the graph-based method of the coherence evaluation, namely, a semantic similarity graph [19]. Let us represent a text \( T \) as the set of sentences \( T = \{s_1, s_2, \ldots, s_N\} \) (\( N \) is a number of sentences); each sentence \( s_i, i = 1, 2, \ldots, N \) is represented as the set of words \( \{w_{i1}, w_{i2}, \ldots, w_{im}\} \). The next step consists in the formalization of each sentence in a semantic space. It is necessary to
perform sentence representation as a vector using a pre-trained semantic embedding model. In order to represent each word in a vector form, it is suggested to use ELMo model. Such a choice can be explained by the possibility of the architecture of this model to take into account the context of word usage while generating an output vector for it. The vector representation of each word is not unique and depends on surrounding words. Thus, each sentence $s_i$ incorporates a set of vectors $\{w_{i1}, w_{i2}, \ldots, w_{im}\}$. Then, the averaged value of word vectors represents the sentence $s_i$:

$$s_i = \frac{1}{m} \sum_{k=1}^{m} w_k^i$$

(2)

According to the formed set of sentences, a semantic similarity graph $G(V, E)$ is built where $V$ is the set of vertices, $E$ denotes the set of edges. Each vertex $v_i \in V$ corresponds to a certain sentence, while a weighted directed edge $e_{ij} \in E$ denotes the connection between sentences $s_i$ and $s_j$. There are three different approaches for the setting of edges $e_{ij} \in E$: preceding adjacent vertex (PAV), single similar vertex (SSV), and multiple similar vertex (MSV).

In the PAV approach, the connection is established just between adjacent sentences according to the order of sentences within a text. As a reader processes a text sentence by sentence, he or she can understand a current document’s fragment basing on the previous text’s parts. While designing a semantic similarity graph, such a cognitive process can be formalized using directed edges from the previous vertices (sentences) to the next ones. Firstly, an attempt to set an edge $e_{ij}$ from the current vertex $v_i$ to the previous one $v_j$ is performed. If the value of the corresponding weight is non-zero, then the edge is set and the next vertex is chosen as current. In another case, the attempt to set an edge is performed for vertices $v_i$ and $v_{j-1}$. The weight of the edge $e_{ij}$ denotes a semantic similarity between sentences $s_i$ and $s_j$; it is calculated by the following equation:

$$\text{weight}(e_{ij}) = \alpha \text{uot}(s_i, s_j) + (1 - \alpha) \cos(s_i, s_j),$$

(3)

where uot is the ratio of a number of common tokens of sentences $s_i$ and $s_j$ to a number of unique words; $\cos(s_i, s_j)$ is a cosine distance between corresponding sentence vectors; $\alpha \in [0, 1]$ is a regulative parameter that allows taking into account the impact of different components on the accuracy of the method. In order to apply the uot function, a lemmatization operation is performed for the words of the sentences $s_i$ and $s_j$.

While building the graph $G(V, E)$ with the usage of the SSV approach, it is possible to set edges between vertices regardless of their order. For each vertex $v_i$, the search for the most similar vertex $v_j, i \neq j$ is performed. The most similar vertex for $v_i$ is chosen according to the highest weight of an edge $e_{ij}$ among all possible connections. Then the corresponding edge $e_{ij}$ is set. It should be mentioned that the outdegree value of each vertex is equal to one. The weight of an edge $e_{ij}$ is calculated in the following way:

$$\text{weight}(e_{ij}) = \cos(s_i, s_j)$$

(4)

The denominator $|i - j|$ helps to take into account the distance between sentences. It is expected that adjacent sentences should have stronger connections than long-distance parts of a text.

According to the MSV approach, the search for multiple vertices $v_j, i \neq j$ is performed for each vertex $v_i$. An edge $e_{ij}$ is set between vertices $v_i$ and $v_j$ when the corresponding weight is greater than a pre-defined threshold value $\theta$. The weight of edges is calculated as was shown in the Equation (4). Thus, the MSV approach allows taking into account the connection between all sentences according to the distance between them. The examples of graphs built by PAV, SSV, and MSV approaches are shown in Figure 4. The coherence of an input text $T$ is calculated as the average value of all edge weights of the built graph $G$. 
2.2.2. LSTM-Based Coherence Estimation Model

According to the performed analysis of state-of-the-art coherence estimation models, LSTM neural networks are commonly used in order to represent either sentences or whole texts. Moreover, consecutive convolution operations are applied for the purpose of revealing main semantic components within input data. It should be mentioned that all considered models were tested on English corpora. As the Polish language belongs to another family group, it is advisable to verify if the usage of LSTM layers is suitable for the processing of Polish texts at all while evaluating output coherence. The key question consists in the order of words’ processing, namely, whether it should be taken into account for the coherence evaluation of Polish corpora. Thus, it is suggested to design an LSTM-based coherence estimation model without additional CNN layers in order to estimate the expediency of the analysis of sentences in a word-by-word manner. Let us consider the main components of a suggested coherence estimation model.

The input of the model is represented as a text \( T = \{s_1, s_2, \ldots, s_N\} \), the output value interprets the estimation of the coherence of the text. Let us divide the input text \( T \) into a set of cliques, i.e., ordered groups of sentences where each group incorporates \( L \) sentences. Such a set is created by the sliding of a “window” (width of “window” is equal to \( L \)) through the text with an ordinary step. The cardinality of the created set of cliques is equal to \( N - L - 1 \). According to the experimental search for the optimal value of \( L \) that was provided in the paper [20], it is advisable to set the following width of the “window”: \( L = 3 \). Such a search was performed by taking different values of \( L \) during the training and inference of a neural network. The further increase of this parameter allows taking into account more sentences simultaneously; however, it does not provide a significant accuracy improvement while solving the sentence ordering task. It should be mentioned that the corresponding experimental selection of \( L \) was performed according to the analysis of an English corpus. However, this feature should not have a severe dependency on a language: despite the different structure of the sentences of English and Polish languages, the way to group sentences around the key idea of a text should remain the same for all persons. The size of a “window” may depend on the style of a text (e.g., the long-distance sentences of a step-by-step manual can be more connected than the sentences of some

![Figure 4. The examples of the semantic similarity graphs built by different approaches.](image)
fashion that describes nature, environment, etc.). As both English [20] and Polish corpora consist of news reports, it was decided to set the same value \( L = 3 \) during the designing and experimental verification of the model. The text \( T \) is represented in the following way:

\[
T = \{ \langle s_1, s_2, s_3 \rangle, \langle s_2, s_3, s_4 \rangle, \ldots , \langle s_{N-2}, s_{N-1}, s_N \rangle \} \tag{5}
\]

The coherence of the entire text \( T \) is calculated by the equation:

\[
Coh(T) = \prod_{c \in T} p(y_c = 1), \tag{6}
\]

where \( p(y_c = 1) \) is the probability of the coherence of an input clique \( c \in T \). Thus, the key task of the model implies the estimation of the probability of coherence for all text’s cliques. This task is performed using an LSTM-based neural network. Figure 5 shows the main components of the designed model.

![Figure 5](image)

**Figure 5.** The main components of the coherence estimation model of input cliques based on an LSTM neural network.

Firstly, all input sentences are passed through a preprocessing block. The purpose of the preprocessing block is to convert an input sentence \( s_i, i = 1, 2, 3 \) to a sequence of word vectors \( \{w_1^i, w_2^i, w_3^i, \ldots \} \). Despite the ability to remove stop-words from input texts, it is proposed to use all tokens for the vector representation of a sentence using a pre-trained ELMo embedding. It is expected that such an approach will help recognize the context of the words for the appropriate vector representation of tokens in a semantic space. After the applying of the embedding model, a padding operation is performed. The purpose of this operation consists of the addition of extra masking symbols (zeros) to each input sequence in order to set their common fixed length. Such an approach allows the usage of mini-batch mode during a training process. The length of each sequence is set up to
40 words. According to the research of the American Press Institute [21], it is expected that longer sentences are hard to perceive during reading. Such sentences are truncated to the chosen length.

Then, all sequences of vectors are passed through a sentence model in order to form a sentence embedding representation. The sentence model is shared, i.e., the same layers of the neural network are applied for each input sequence. The sentence model incorporates two layers: masking and LSTM. The purpose of the masking layer consists of the filtering of non-valuable parameters (zeros) from an input sequence. Then, the LSTM layer performs word-by-word processing. According to the classical LSTM with a forget gate [22], the signal flow is described by the following equations:

\[
f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t = o_t \odot \sigma_o(c_t),
\]

where \(x_t\) denotes an input signal (a word vector \(w_t\)); \(h_t\) represents the final output vector of an LSTM unit; \(W, b\) are free parameters; \(\odot\) sign denotes an element-wise product.

Thus, the result of the applying of the sentence model to the input sequences is represented as corresponding three vectors \(h_{s1}, h_{s2}, h_{s3}\). Then the concatenation layer is applied in order to form the common vector representation of the input clique:

\[
h_c = [h_{s1}, h_{s2}, h_{s3}]
\]

The next dense layers should be considered as a binary classifier. The last dense layer contains a single neuron with a sigmoid activation function. The output of this layer interprets the probability of the coherence of the input group of sentences:

\[
q_c = f(W_{sen}^T h_c + b_{sen}) \\
p(y_c = 1) = \text{sigmoid}(U^T q_c + b)
\]

2.2.3. Using of the BERT Model

While considering the representation of multi-language texts, it is advisable to pay attention to the Transformer architecture [23]. This architecture is widely used in different NLP tasks, especially, for machine translation purposes. The key feature of the Transformer-based approach consists of an attention mechanism that implies the revealing of connections between different words of a sentence. Figure 6 demonstrates the example of such attention from each left token to another right. The darker color of lines indicates higher attention weight between tokens. Tokens [CLS] and [SEP] denote the start and the end of sentences correspondingly. The sets of such attention mechanisms (so-called “heads”) allow the effective encoding of input sequence with further decoding of it for the representation of translated sequence without the loss of information about word connections. The availability of such connection between tokens, regardless of word order, can be important for synthetic languages like the Polish language, where the order of words is not so valuable in comparison with analytical languages. Thus, it is suggested to consider the impact of such attention mechanisms on the coherence evaluation process of Polish texts.

The classical Transformer model incorporates two consecutive parts: an encoder and a decoder. According to the coherence estimation process, it is necessary to perform the vector representation of input data for further binary classification. Thus, the encoder part of a Transformer-based model should be used. In order to perform such representation, the Bidirectional Encoder Representations from Transformers(BERT) model can be utilized. The training of this Transformer-based model provides for bidirectional signal flow. It means that the processing of a text incorporates both left-to-right and right-to-left signal directions. The results obtained showed that such an approach allows better recognition of
context in comparison with a single-direction mode. In the case of the analysis of synthetic languages, it can help to reveal more dependencies between tokens regardless of word order. As the BERT model is designed to generate a language model, only the encoder part of the Transformer architecture is trained. Thus, the pre-trained BERT model can be utilized during the coherence evaluation process of a Polish-language corpus.

Let us formulate the coherence estimation task according to the previously considered LSTM-based model. Thus, the representation of an input text and its coherence calculation are described by Equations (5) and (6). The key task consists in the prediction of the coherence of an input clique (a group of sentences). Figure 7 shows the main components of the coherence estimation model based on the pre-trained BERT layers. Firstly, input sentences are passed through a preprocessing block. In contrast to the previously considered model, all sentences are processed as a whole block. The preprocessing implies the tokenization of the input text block using a pre-defined BERT tokenizer. Thus, the output of the block is represented as a set of numbers \( \{ t_1, t_2, \ldots \} \) where each number corresponds to an input token according to the vocabulary \( V \) of the tokenizer. Moreover, as the Polish language has different grammatical cases, the cased version of both the tokenizer and the BERT model are suggested to be utilized.

The next component is the pre-trained BERT model itself. In order to verify how the representation of the input text by the pre-trained BERT model can fit coherence requirements, it is proposed to “freeze” all parameters of this component during a training process. Such an approach can help verify whether this neural network architecture is suitable for the coherence evaluation task at all. The BERT model incorporates the following parts:

- Embedding that processes the sequences of input token numbers and attention masks in order to perform the dense representation.
- Encoder that consists of 11 attention-based components.
- Pooler that represents the final vector representation of the input text.

The output of the pooler layer returns the required vector \( h_c \). The further signal flow through a binary classifier is described in the Equations (9).
Figure 7. The main components of the BERT-based coherence estimation model for an input clique.

3. Experiments and Results

3.1. Models Preparation and Training

Let us consider which metrics can represent the effectiveness of the coherence estimation models. The effectiveness of these models is usually verified on a sentence ordering task. This task implies the ability of the model to recognize the coherent text (original version) and incoherent one (the text where all sentences of this text are permuted). The model calculates the coherence of each version. The recognition is considered successful if the coherence of the original version is higher than the corresponding value of the modified version. The output metric is calculated by the equation:

$$\text{SentOrd}(\text{Corpus}) = \frac{\sum_{T \in \text{Corpus}} \text{IsRecognized}(T)}{|\text{Corpus}|},$$  \hspace{1cm} (10)

where $\text{Corpus}$ is a set of texts, $\text{IsRecognized}(T)$ is a function that returns either 1 or 0 in the case of successful/ unsuccessful recognition correspondingly. In addition, another task can be additionally utilized in order to verify the effectiveness of coherence estimation models. The insertion task implies the ability of the model to detect the correct position of a sentence in a text. Firstly, the coherence of the original version of a document is calculated. Then a certain sentence is extracted from the text with further insertion in all possible positions except a correct one. The recognition is considered successful if the coherence of the original text’s version is greater than the coherence of retrieved versions. The output metric $\text{InsTask}(\text{Corpus})$ is calculated by Equation (10). It should be mentioned that the insertion task is harder to resolve because it is necessary to understand the whole idea of a text in order to insert a sentence in the right position. A similar task is used in exams like TOEFL or IELTS in order to examine the knowledge of the English language. In order to approximate the insertion task to the mentioned test, it was decided to extract the most informative sentence, namely, the longest sentence due to the count of words.

In order to generate training and test datasets, the ChronoPress text corpus [24] was utilized. The pre-trained ELMo embedding model for the Polish language was taken from
the repository [25]. This ELMo model was trained on Wiki dumps and texts from common web-crawl. The dimension of each word vector is equal to 1024. The lemmatization of Polish texts for the semantic similarity graph building was performed by the Morfeusz 2 inflectional analyzer and generator [26]. The pre-trained BERT tokenizer and model were taken from the Polbert project [27]. The training dataset was generated from the cliques of original texts (coherence examples) and from the cliques of the permutated version of texts (incoherent examples). The binary cross-entropy function was chosen as an objective minimizing function for both neural networks. The Adam optimizer was utilized for the parameter update process. In order to prevent overfitting, a sentence ordering metric was calculated at the end of each epoch.

All software was written using the programming language Python 3.6. The Tensorflow 2 framework was utilized to design and train the LSTM-based neural network. The training of the BERT-based model was implemented with the usage of PyTorch packages. All calculations were performed on Google Colab Platform using a GPU hardware accelerator. The corresponding software is presented as a Google Colab notebook [28] that contains examples how to load and use trained models in order to estimate the coherence of Polish documents.

3.2. Results

After the preparation of the considered models, the accuracy of the semantic similarity graph (SSG), the LSTM-based model (LSTM), and the BERT-based model (BERT) was calculated according to the solving of the sentence ordering and the insertion task. The results obtained are shown in Table 1.

Table 1. Accuracies of the solving of different tasks for coherence estimation models on a Polish corpus.

| Model                        | Parameter Value | Sentence Ordering | Insertion Task |
|------------------------------|-----------------|-------------------|----------------|
| *SSG (PAV), parameter α*    | 0.0             | 0.786             | 0.105          |
|                              | 0.1             | 0.788             | 0.109          |
|                              | 0.2             | 0.794             | 0.107          |
|                              | 0.3             | 0.792             | 0.131          |
|                              | 0.4             | 0.790             | 0.133          |
|                              | 0.5             | 0.796             | 0.131          |
|                              | 0.6             | **0.807**         | **0.138**      |
|                              | 0.7             | 0.799             | 0.131          |
|                              | 0.8             | 0.786             | 0.116          |
|                              | 0.9             | 0.757             | 0.112          |
|                              | 1.0             | 0.740             | 0.098          |
| *SSG (SSV)*                  |                 | **0.760**         | **0.083**      |
| *SSG (MSV), parameter θ*    | 0.0             | **0.759**         | **0.107**      |
|                              | 0.1             | 0.718             | 0.092          |
|                              | 0.2             | 0.696             | 0.077          |
|                              | 0.3             | 0.716             | 0.072          |
|                              | 0.4             | 0.659             | 0.105          |
|                              | 0.5             | 0.731             | 0.105          |
|                              | 0.6             | 0.665             | 0.094          |
|                              | 0.7             | 0.584             | 0.077          |
|                              | 0.8             | 0.525             | 0.055          |
|                              | 0.9             | 0.055             | 0.002          |
| *LSTM*                       |                 | **0.875**         | **0.168**      |
| *BERT*                       |                 | **0.835**         | **0.143**      |

Firstly, let us analyze the obtained results for the different approaches of the semantic similarity graph. In the case of the PAV approach, the increase of both metrics is tracked
during the increase of the regulative parameter \( \alpha \) till the reach of the peak with \( \alpha = 0.6 \). The further increase of the \( \alpha \) parameter decreases corresponding metrics. Such a value \( \alpha = 0.6 \) may indicate the necessity to take into account both lexical and semantic components of a text simultaneously while evaluating the coherence of a document. Moreover, the highest metric values for the semantic similarity graph method were retrieved using the PAV approach that underlines the expediency of the tracking of the connection between sentences sequentially. Such a conclusion can be drawn from the metrics of the SSV and MSV approaches, too. Despite the change of the regulative parameter \( \theta \), all corresponding metrics of the MSV approach are lower than the corresponding values of the SSV approach. Thus, it is advisable to analyze a text in a sentence-by-sentence manner according to their positions while estimating the coherence of a Polish corpus.

The metrics for other models based on neural networks outperform corresponding values of the semantic similarity graph approach. The highest values of the LSTM-based model for both tasks may indicate the possibility to apply different recurrent neural networks in order to estimate the coherence of Polish texts. Moreover, the word order can be taken into account during the coherence evaluation of Polish documents despite the relevance of the Polish language to the category of synthetic languages. As for the BERT-based model, the obtained metrics may indicate the ability of such architecture to resolve typical tasks even with the usage of externally pre-trained parameters. The increase of the corresponding values for this approach can be achieved by the fine-tuning of all parameters of the BERT model according to the coherence estimation task.

4. Conclusions

According to the analysis of state-of-the-art methods of the coherence evaluation and results obtained, the following conclusions can be drawn:

- It is advisable to use both LSTM layers and convolutional operations during the designing of a neural network for the coherence evaluation of texts. The LSTM cells allow the performing of the vector representation of either sentences or entire texts according to items position. Meanwhile, the applying of convolutional operation helps reveal main semantic components from an input text that may represent the topic of the input sequence.
- The highest metrics among the different semantic similarity graph approaches were obtained with the PAV approach. Thus, it is advisable to analyze a text in a sentence-by-sentence manner according to its order within a text for a Polish corpus.
- The peak of the accuracy of the PAV approach with the regulative parameter \( \alpha = 0.6 \) while solving typical tasks may indicate that both lexical and semantic components of a text can be taken into account during the estimation of the coherence of Polish texts.
- The highest values of metrics among all models were retrieved for the LSTM-based neural network. Thus, despite the relevance of the Polish language to the category of synthetic languages, word order can also be taken into account during the coherence evaluation of a text corpus.
- The results obtained for the BERT-based model with fixed BERT parameters may indicate that such Transformer-based architecture can be used in order to represent different Polish text parts for the further coherence classification layers. The accuracy of this model can be increased by the fine-tuning of all parameters according to the training of the model for the estimation of the coherence of texts of the Polish language.

Author Contributions: Conceptualization, S.T., S.P., and A.K.; methodology, S.T., S.P., and A.K.; software, A.K.; validation, A.K.; formal analysis, S.T., S.P., and A.K.; writing—original draft preparation, A.K.; writing—review and editing, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: In this section, please add the Institutional Review Board Statement and approval number for studies involving humans or animals.
Informed Consent Statement: Any research article describing a study involving humans should contain this statement.

Data Availability Statement: Software code to utilize trained models is represented as a Google Colab Notebook: https://colab.research.google.com/drive/1OMbYKmy9FYtKMRTE-byQrvqXmooH8r?usp=sharing. It provides a user with instructions on how to design models, load their weights, and run them in an inference mode in order to estimate the coherence of Polish texts.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pishdad, L.; Fancellu, F.; Zhang, R.; Fazly, A. How coherent are neural models of coherence? In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, December 2020; pp. 6126–6138. doi:10.18653/v1/2020.coling-main.539.
2. Xiong, H.; He, Z.; Wu, H.; Wang, H. Modeling coherence for discourse neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7338–7345.
3. Pogorilyy, S.; Kramov, A. Coreference Resolution Method Using a Convolutional Neural Network. In Proceedings of the 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 18–20 December 2019; pp. 397–401, doi:10.1109/ATIT49449.2019.9030596.
4. Kramov, A. Evaluating text coherence based on the graph of the consistency of phrases to identify symptoms of schizophrenia. *Data Rec. Storage Process.* 2020, 22, 62–71, doi:10.35681/1560-9189.2020.11.207786.
5. Barzilay, R.; Lapata, M. Modeling Local Coherence: An Entity-Based Approach. *Comput. Linguist.* 2008, 34, 1–34, doi:10.1162/coli.2008.34.1.1.
6. Walker, M.A.; Joshi, A.K.; Prince, E.F. *Centering Theory in Discourse*; Claredon: London, UK, 1998.
7. Guinaudeau, C.; Strube, M. Graph-based Local Coherence Modeling. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; Association for Computational Linguistics: Sofia, Bulgaria, 2013; pp. 93–103.
8. Zhang, M.; Feng, V.W.; Qin, B.; Hirst, G.; Liu, T.; Huang, J. Encoding World Knowledge in the Evaluation of Local Coherence. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; Association for Computational Linguistics: Denver, CO, USA, 2015; pp. 1087–1096, doi:10.3115/v1/N15-1115.
9. Feng, V.W.; Lin, Z.; Hirst, G. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; Dublin City University and Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 940–949.
10. Mann, W.C.; Thompson, S.A. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text Interdiscip. J.* 1988, 8, 243–281, doi:10.1515/text.1.1988.8.3.243.
11. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on International Conference on Machine Learning—Volume 32, JMLR.org, ICML’14, Beijing, China, 22-24 June 2014; pp. II-1188–II-1196.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186, doi:10.18653/v1/N19-1423.
13. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 2227–2237, doi:10.18653/v1/N18-1202.
14. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543, doi:10.3115/v1/D14-1162.
15. Li, J.; Hovy, E. A Model of Coherence Based on Distributed Sentence Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 2039–2048, doi:10.3115/v1/D14-1218.
16. Lai, A.; Tetreault, J. Discourse Coherence in the Wild: A Dataset, Evaluation and Methods. In Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, 12–14 June 2018; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 214–223, doi:10.18653/v1/W18-5023.
17. Mesgar, M.; Strube, M. A Neural Local Coherence Model for Text Quality Assessment. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4328–4339, doi:10.18653/v1/D18-1464.
18. Moon, H.C.; Mohiuddin, T.; Joty, S.; Xu, C. A Unified Neural Coherence Model. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 2262–2272, doi:10.18653/v1/D19-1231.

19. Pogorilyy, S.D.; Kramov, A.A. Assessment of Text Coherence by Constructing the Graph of Semantic, Lexical, and Grammatical Consistency of Phrases of Sentences. *Cybern. Syst. Anal.* 2020, 56, 893–899, doi:10.1007/s10559-020-00309-7.

20. Cui, B.; Li, Y.; Zhang, Y.; Zhang, Z. Text Coherence Analysis Based on Deep Neural Network. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; Association for Computing Machinery: New York, NY, USA, 2017; CIKM ’17, pp. 2027–2030, doi:10.1145/3132847.3133047.

21. How to Make Your Copy More Readable: Make Sentences Shorter—PRsay. Available online: http://comprehension.prsa.org/?p=217 (accessed on 11 March 2021).

22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, 9, 1735–1780, doi:10.1162/neco.1997.9.8.1735.

23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, U.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; NIPS’17, pp. 6000–6010.

24. Pawlowski, A. *ChronoPress—Chronological Corpus of Polish Press Texts (1945–1962)*; CLARIN2017 Book of Abstracts; 2017.

25. Fares, M.; Kutuzov, A.; Oepen, S.; Velldal, E. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden, 22–24 May 2017; Association for Computational Linguistics: Gothenburg, Sweden, 2017; pp. 271–276.

26. Woźniński, M. Morfeusz Reloaded. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14), Reykjavík, Iceland, 26–31 May 2014; European Language Resources Association (ELRA): Reykjavík, Iceland, 2014; pp. 1106–1111.

27. Polbert—Polish BERT. Available online: https://huggingface.co/dkleczek/bert-base-polish-uncased-v1 (accessed on 11 March 2021).

28. Polish Coherence Models—Google Colab Notebook. Available online: https://colab.research.google.com/drive/1OMbYKmy9fYVtKMRTE-byQrivqXmooH8r?usp=sharing (accessed on 11 March 2021).