Automated bot Detection Based on Coherence Metric

Oleksandr Marchenko and Mariam Isoieva

Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine

Abstract

This paper describes a model of bot detection based on a coherence metric. Much attention is being paid to natural language generation technologies since the middle of the previous century, many companies invest in research in this direction. The quality of automatically generated texts becomes better. Nowadays, it is hard to distinguish machine-generated texts from those written by humans. While this can be used for helping people in automating even creative tasks, there is also a downside of such technologies being widely available, e. g. easier spread of fake news and propaganda. That is why it is also important to develop efficient bot detection methods for having misinformation protection tools. Here an attempt to create such a model is made. One of the main distinguishing features of high-quality texts is coherence. This is also what automatically generated texts, especially long ones, often lack in. A classifier with a set of features based on a coherence metric and syntactic characteristics has been built. The method can be extended and used for different languages.

Keywords [1]

Bot detection, natural language generation, coherence

1. Introduction

The human mind is in the process of constant evolution, new intellectual needs arise. Automatically generated texts can be used for good purposes: for education, entertainment, for easier and faster knowledge gathering and summarization. But there is also a dark side to these technologies. They are used for sharing propaganda and fake news, conducting illegal political campaigns, committing financial crime through automatic credit applications generation, for misleading people. And such texts are being spread everywhere faster and faster. Posting of such texts in social networks and web resources can be automated. That is why bot detection technologies are needed. While much attention is paid to the development of natural language generation systems, the detection of machine-generated texts is out of the spotlight. Are there more potential dangers than everyone sees?

With the rapid development of natural language generation technologies, it is hard to distinguish automatically created texts from those written by humans. While many existing methods for bot detection rely on features related to supporting information, such as settings of the analyzed accounts in social networks or their activity, less attention is paid to such analysis based on pure text characteristics. The main goal of this work is to develop a model for the identification of automatically generated text based on its coherence.

Coherence is one of the main characteristics of high-quality text, which is easy to perceive and understand. This concept is multi-faceted. Coherence depends on both semantic and syntactic features, and automatically generated texts, especially long ones, lack coherence. There can be logical breaks or topic switches inside the text, which make the text inconsistent. That is why it is reasonable to use a coherence metric, e.g. [1], as a distinguishing feature for bot detection.

2. Models of natural language generation

Natural language generation (NLG) is considered one of the most important milestones for artificial intelligence. Solving the task of automatic text creation requires using powerful algorithms.

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: omarchenko@univ.kiev.ua (A. 1); isoyevamaryam@gmail.com (A. 2)
ORCID: 0000-0002-5408-5279 (A. 1)

It is also valuable to understand how text is being generated by the models used today to build an efficient bot detection system. Some of the modern methods of NLG and their classification are described in this section.

Different criteria for the classification of NLG tasks and methods related to the generation of natural language text can be defined.

Some systems are used to generate texts only on specific topics, and others can be used to generate texts on arbitrary topics. For generating texts related to a predetermined topic, it is possible to expand the dataset and to train the model on thematically related data, to use tools such as specialized dictionaries, knowledge bases. The construction of "universal" text generators requires finding ways to obtain new data at the user's request or the availability of significant amounts of data that can be used for generation. Thematic diversity is necessary for systems of general dialogue, summarization, question answering.

In addition, universal systems must preserve the consistency of style for the specific text generated and for all texts produced by the system. This aspect is crucial for algorithms based on the extraction of sentences from different corpora. In addition to syntactic integrity and semantic unity, the stylistic homogeneity of the text is also an important indicator of quality.

By the methods of text creation, models can also be classified into extractive and abstractive. Considering the example of automatic summarization, compiling an abstract from sentences of the original text is called extractive. Abstractive summarization means creating a new text without directly using original sentences as parts of the summary formed.

For natural language generation systems, some input data is usually required as a basis for the future text. Such data may be structured, e. g. numbers, maps of fields and corresponding values, graphs, or unstructured, such as human-written text. This can be a corpus of texts, on which the model is trained, or which is used to find and convert the necessary fragments (for example, when sentences from the corpus are used). For question answering systems, the text corresponding to the request is generated based on the questions formulated by a person. In addition, the input of the natural language text generation system can contain numerical data: characteristics of the observed phenomenon or process, indicators for generating reports, etc. A natural language text generation system may also have a subsystem of semantic analysis of the input text or a set of texts for creating abstract representations of the future text content. Such formal representations can be built based on certain data and be set as a basis for text formation.

The length of the text generated by the system can be controlled. For example, the natural description of the image may consist of one sentence or phrase, the user can specify the number of sentences in the automatically generated summary in advance: generated short descriptions of news articles are usually one or several sentences long and automatically generated articles on a particular topic can be much longer. The length of the texts produced by a deep learning model often depends on what data it has been trained on. The capability to generate long texts can be determined by the complexity of the model. Many issues arise when it comes to the generation of long texts. For example, recurrent neural networks can have self-repetitions when the same fragments of text are repeated several times. It is also hard to achieve coherence and ease of perception for human readers. These may require some additional mechanisms for the control of semantics of the text being generated. There are interactive systems for generating natural language texts, for which a user can control some stages of generation, make intermediate decisions, and "static" ones, for which human control is possible only at the level of the basis, the request. Different problems require consideration of their peculiarities for their efficient solving.While methods of text generation depend on the tasks which they solve, there is a common basis for such algorithms.

Reiter and Dale summarize the experience of computational linguists and formulate the main steps of automatic generation of natural language text: definition of content, discourse planning, sentence aggregation, lexicalization, generation of key concepts, linguistic implementation [2]. In the first stage, the information that should be specified in the generated text is defined. The form of representation and the method of its formation depend on the generation problem formulation. At this stage, the concepts that will be described in the text and the connections between them that need to be highlighted are defined. The second stage is "planning" of the discourse, i.e., determining the basic structure of the future text, organizing the semantic structures identified at the first stage. During the next phase, the sentences representations are formed and combined into larger structures. At this stage the basis of coherence of the future text is set: the concepts, which are connected, are grouped to form some structures. Lexicalization is the choice of those words and phrases that reflect the essence of the underlying meaning and are correct to use. Most of the first text generation systems used prepared phrases suitable for the related domain or topic. Lexical diversity is a sign of high-quality text. Increasingly complex generation methods are being developed to achieve it, researchers try to find methods for automatic expansion of thesauri and ontologies. The next step is the definition of the main entities, the concepts to be discussed, for example, certain proper names that need to be mentioned in the text [2]. For cases when the basis of generation is natural language text, there are problems of selection of the named entities and the choice of language pointers. These steps are necessary for further coordination of the elements of the generated text, selection of syntactic structures, etc. Linguistic implementation is one of the most important stages, as it is the process of generating the text itself. Grammar rules of a particular language should be followed. The main focus is on this step nowadays, some of the previous ones can be omitted.

Figure 1 summarizes the mentioned stages of natural language texts generation.

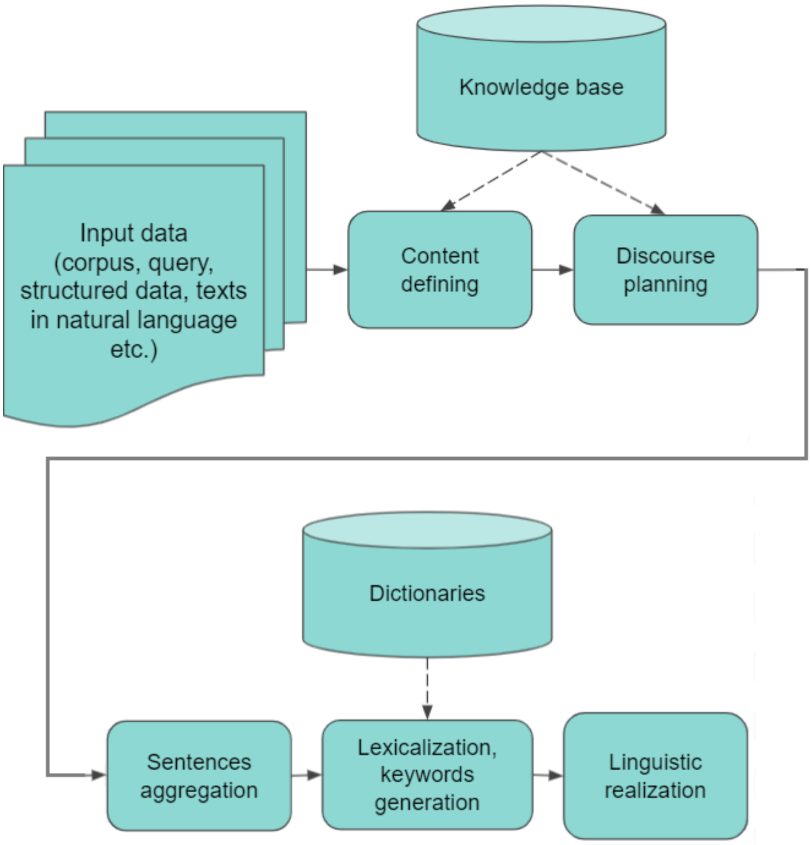

Knowledge base
Input data (corpus, query, structured data, texts in natural language etc.)
Content defining
Discourse planning
Dictionaries
Sentences aggregation
Lexicalization, keywords generation
Linguistic realization

Figure 1: Common steps for natural language generation, based on [2]

Methods of generating natural language text, based on the use of deep learning models are the most popular and widely used today. But they have some limitations. Most models, such as basic recurrent neural networks, GRU (Gated recurrent unit), LSTM (Long Short Term Memory) and their modifications, can be used to generate short texts, the length of which is one or more sentences. More complex models, such as the GPT family, are used to generate longer texts. But the size of the trained model is a significant barrier for using such methods in practice, as there is a need for significant computing resources, access to which is often limited and expensive.

In addition to that, since neural networks remain black boxes, the improvement of such models is complex and is usually based on empirical studies of modified versions of the base model. There is also no guarantee of the quality of the generated text, especially if it is generated based on data that is new to the model, for example, based on texts on topics not present in the training sample.

Attempts to explain the process of data processing and generation by neural networks are a separate area of research of deep learning methods.

Deep learning methods can be used to generate locally coherent short texts (one or several sentences long), but the generation of longer sequences is still challenging. Generated text can be inconsistent, hard to read, describe one phenomenon or fact several times or have inappropriate semantic breaks. Recurrent neural networks are often used to generate text. Various modifications of the basic architecture are developed to improve the generated text, in particular, to achieve a certain level of coherence and semantic unity.

Kiddon et al. proposed to extend the basic model of generation, in particular, to use a “list” of what should be described in the text, to ensure the coherence of the text generated by the recurrent neural network [3]. The developers of the pointer generator network architecture also use information about the text that has already been generated and parts of the original text that have been analyzed [3]. This model is used for the correct representation of factual information for automatic summarization.

Generative adversarial network (GAN) is a popular architecture for generating images and texts. It consists of two parts: generator and discriminator. The discriminator recognizes machine-generated images or texts and becomes more resistant to noise in its input data, and the generator learns to “fake” the distribution of real data.

With sufficient amounts of training data, GANs generate impressively realistic images of people, their faces, animals, etc. Apart from that, this architecture is used to generate texts. The discriminator “learns” to recognize machine-generated text, and the generator trains to form texts, which are similar to the human-written ones.

Generation of long texts has been studied by Sang Cho et al.; researchers propose to use GAN [4]. They emphasize the importance of local and global coherence. Namely, two discriminators have been used: one to assess cohesion, the other to evaluate coherence. Researchers make the basic assumption that global coherence depends on the organization of sentences and focus on assessing coherence at the sentence level of individual paragraphs, not just adjacent sentences. Evaluation of cohesion within sentences is based on the technique of the Deep Structured Semantic Model, which was originally used to determine semantic similarity [4]. Previous attempts to generate text using GAN have used language models as discriminators, which do not always adequately address aspects of text coherence.

Most generation methods are based on the use of some raw or pre-processed data: individual sentences of the corpus, structures that describe the plot line (for generating stories), formal graph structures. Depending on the presentation format of such information and the method of generation, there are different ways to reduce the amount of data searched and speed up their analysis.

Even for heavy and powerful systems, such as GPT-2 or GPT-3, it is hard to model the long-term dependencies for bigger texts, preserve the consistency, create the correct structure of the text. The main indicator of such a problem is the low level of coherence, when different parts are not logically united, connections between them are inappropriate or absent, the whole text is hard to read. Measuring coherence is a hard task even for humans but establishing some basis for evaluation is required for further improvement of NLG and bot detection models.

3. Aspects of coherence

For the solution of any problem, which requires generation of natural language text, coherence of the text is necessary for human perception of the result. Definitions of coherence differ, but most of them concentrate on the semantic aspect. According to the Cambridge Dictionary, “coherent” means clear, consistent, characterized by the relatedness of its parts [5]. Linguists study the coherence of the text as a phenomenon that finds its manifestation through the use of certain linguistic means: ellipses, usage of synonyms, conjunctions, linguistic references and other connections. The study of coherence is important for the development of NLG systems, as well as for bot detection models.

The coherence of any text depends on the semantic relatedness of the main concepts, phenomena, ideas mentioned, the syntactic consistency of the components of individual sentences and their unity within a group of sentences.

Depending on the level of consideration, concepts of local and global coherence of the text are defined. Local coherence exists at the level of individual sentences, connections between parts of neighboring sentences, it relates to semantic transitions between successive sentences. This property is important for the creation of high-quality natural language text. Local coherence is necessary for global coherence, which is the logical unity, the integrity of the whole text.

Sevbo [6] have studied how the main idea is transferred and developed from sentence to sentence. The researcher focuses on the realization of coherence using repetition of meaningful words in consecutive sentences. The algorithm proposed in the work [6] is based on determining the syntactic structure of sentences of the input text, construction of "phrase trees" that describes it. In addition to the repetition of words, it is necessary to take into account anaphora, coreferences.

A method of evaluation of the coherence level, developed by Lapata and Barzilay, is based on the model of discourse representation [7]. Researchers propose to map each text to a matrix of entities that reflects the distribution of elements of discourse in sentences. The rows of this matrix correspond to the sentences, and the columns represent the elements of the discourse. The corresponding elements of the matrix indicate the presence or absence of an element of discourse in the sequence of sentences and their syntactic role. Special notations are used for different grammatical categories (subjects, objects, and others). If the element for the corresponding column is absent in the sentence, a special mark '-' is put [7]. Coreferences for the relevant elements are also considered. Dependency parsers are used to determine the grammatical role of each word, in particular, the authors propose to use parsers developed by Lin [8] and Briscoe [9]. The quality of coherence modelling depends significantly on how accurate the coreference resolution systems and dependency parsers are. Researchers emphasize that the discourse representation model plays an important role. This method is based on previous studies of the coherence of the natural language texts, in which the main focus is on the study of the influence of grammatical connections between its elements on the local coherence of the text. The authors also make an assumption that the distribution of elements of discourse for coherent texts follows some pattern. They use the achievements from the theory of centering, for which the main aspect of coherence is the number and nature of semantic transitions and changes in "focus", i.e., the main object mentioned [7]. Based on these assumptions, it can be argued that densely filled matrices correspond to coherent texts, while the sparseness of the matrix may indicate a weak semantic connection between the sentences of the text under consideration. This study of coherence focuses on the definition of typical patterns of coherent texts structure in the context of the described model of text representation.

The authors of the model have proposed to form numerical vector representations of texts based on the analysis of probabilities of syntactic role changes in consecutive sentences for each entity. They can be used for the classification of texts and their subsequent study, in particular, for evaluating the coherence. The approach described above is also used to model coherence in text generation. In particular, it is possible to use sentences for ranking to form the most coherent text from them or to restore the original order of sentences. For example, McIntyre has used this technique to generate stories [10]. In his work [10], the researcher has used a dataset of fairy tales by the writer Andrew Lang. This choice of training data is associated with one of the tasks considered by the scientist, namely the generation of stories and fairy tales that could be used for teaching children.

An important step in the study of textual coherence as a phenomenon and the study of methods for its definition has been made by Grosz, Joshi, and Weinstein [11]. They have developed the theory of centering, which has been used in the aforementioned works. Many further developments are based on the properties of coherence identified by these researchers, and the expansion of the constructed theory, in particular, the model described above. The name "centering" is inspired by the basic statement of this theory. According to Grosz et al. [11], some of the entities mentioned in the text are central. The speaker's choice of constructions, linguistic references is determined by the central entities, their features. Therefore, the coherence of discourse depends on the relatedness, compatibility of the properties of the central entity and those related [11].

A technique proposed by Iida and Tokunaga is also related to the theory of centering [12]. The algorithm uses the output of the subsystem for coreference resolution. The authors propose to use the developed metric as one of the features to build a model based on the matrix of entities and demonstrate the improvement of the basic model, taking into account the existing anaphors, the entities described in the text, and the relationships between them.

Since many methods of evaluating and modelling the coherence of the text require some data on which the model is trained or the generation algorithm is based, there is an issue of making these methods independent of a specific topic or subject. Apart from that, the techniques discussed above focus on certain aspects of coherence. There is a need to improve these algorithms by integrating the underlying paradigms.

Researchers Li and Jurafsky have considered methods of such improvement and proposed a method for evaluating coherence and generating coherent text [13]. The proposed discriminative model is based on deep learning methods. A set of consecutive sentences is an input. The central sentence is considered together with the sentences surrounding it. Vector representation of each sentence is obtained as an output of the LSTM network. The vectors corresponding to the sentences are concatenated. Another neural network, the upper layer of which has a sigmoid activation function, is used as a classifier to determine the probability that the input set of sentences is coherent. The generative model is proposed in two variants: a modification of the basic sequence-to-sequence model and a model based on the hidden Markov model and the LDA (Latent Dirichlet Allocation). The training sample includes articles from Wikipedia.

Basile et al. have defined a metric of connection between individual frames [14]. Some results of this study were used in the development of the method [1], which has been applied in this work.

In addition to logical and semantic coherence, there is also an important aspect of thematic homogeneity and coherence. There are studies on this issue, which are based on the use of LDA [15]. Blei et al. define this generative probability model as a three-level hierarchical Bayesian model. LDA is used for classification tasks, modelling of text topics, etc. The developers of this model have tried to take into account the limitations that approaches based on TF-IDF (term frequency – inverse document frequency) and LSI (Latent Semantic Indexing) have [15].

The definition of coherence may also depend on the task for which the text is generated. For example, for assessing the quality of dialogue systems, the specifics of this task must be considered. Coherence metrics for dialogue should reflect these features and assess the general characteristics of a logically constructed text (integrity within a single generated response, etc.).

4. Existing methods of bot detection

Researchers from OpenAI in their report “Release Strategies and the Social Impacts of Language Models” [16] classify the existing models of identification of automatically generated text into:

- those that are based on classical machine learning methods and learn from samples of pairs of generated and corresponding real texts, written by humans (for example, when a part of the human-written text is used as input for the generation model);
- zero-shot classifiers that use pre-trained generative models, such as GPT family or GROVER, and are applied to texts generated by the same or a similar model and allow to determine the probability with which this fragment could be generated by this particular model. The model is not trained additionally. An example of such a classification system is GLTR [17];
- classifiers based on the model fine-tuning, for which the language model is trained, “learns” to recognize itself in different configurations, with different values of hyperparameters [16].

A semi-automatic method of verifying that the text is machine-generated was proposed by the developers of the GLTR system [17]. GLTR is a tool designed to help a person determine whether the text has been generated automatically. The results of the experiments show that the accuracy of determining the machine-generated text by humans using GLTR increases from 54% to 72% [17]. When generating text sequentially word by word, the most common techniques for choosing the next word from the most likely options are max, k-max sampling, beam search. The probabilities of each word depend on the left context. GLTR visualizes outliers and artifacts that may indicate automatic text generation. BERT and GPT-2 are considered as models, the outputs of which can potentially get to the GLTR input [17]. Some further developments are based on this idea of checking the consistency of the distribution. In particular, similar models have been developed for automatic identification of machine-generated text for RoBERTa, XLNet and other models.

Bao et al. use their own method of determining the coherence of natural language text to detect machine-generated spam [18]. The training set for their experiment was formed from the news articles in Chinese and English. The machine-generated part of the dataset consists of texts that were obtained by automatic translation, summarization of texts, permutations of random words and sentences in the original texts [18]. To model coherence, the researchers use pre-trained Bi-GRU. The vectors of the internal states that were obtained after each of the “passes” through the sequence are used as features for classification. They are fed to a convolutional neural network, the last layer of which is for binary classification [18]. The task of identifying machine-generated text can be considered as one that is reduced to the task of authorship attribution. For example, consider the task of identifying a set of accounts of “bots”, from which the text generated by one model is published in social networks. Thus, if the text was generated by a model whose parameters did not change during generation, we can assume that the use of algorithms for determining authorship is appropriate.

Uchendu et al. consider the task of machine-generated text identification from the standpoint of determining the authorship of the text [19]. The authors of the work “Authorship Attribution for Neural Text Generation” consider the following tasks:

- if two texts are specified, determine whether they have been generated by one model
- establish whether the given text was written by a person or generated automatically
- for a given text and a set of generation methods, determine which one has been used [19].

The authors conduct experiments with texts that have been generated using models CTRL, GPT-2, GROVER, XLM, XLNET, PPLM, FAIR and others. According to their results, models GPT-2, GROVER, FAIR are the most difficult to identify and distinguish the text produced by them from the text written by people [19]. To solve the three problems, researchers have developed several common basic architectures. The first option is to represent each word as a vector of 300 elements, summing these vectors and applying a layer with softmax activation. The second model consists of a GRU layer, to the output of which softmax is also applied. The third variant consists of a sequence of convolutional layers, there are also variants with parallel convolutional layers and a combination of RNN and CNN layers [19]. Each of these models is adapted to solve the above three problems.Zhong et al. emphasize that most methods of identifying machine-generated text do not include mechanisms for analyzing the actual structure of the text, which is a determining factor in distinguishing between generated and man-made texts. Researchers propose a graph model in which the text is presented in the format of a graph of entities [20]. A graph neural network is used to create a graph representation. Then such representations of sentences are composed into a single representation of the whole text. In addition, the relationships between adjacent consecutive sentences are modelled.

Tay et al. studied the “artifacts” that the texts generated using different methods have and the influence of factors such as decoding method, model size, input length. They consider a task of identifying automatically generated text. The authors set a goal of better understanding of fundamental properties of neural models designed to generate text [21]. The study of artifacts that are present in the machine-generated text is a new and important area of research. The main conclusion of the work, which was proved experimentally, is that there are such artifacts and that different simulation variants can be identified by using only the text that was generated [21]. This suggests that text generators may be more sensitive to different modelling options than previously thought. The results of this work allow applying the classical methods of classification, specifying the artifacts discovered by the method developed by them as features. In addition, researchers conclude that such artifacts usually relate more to the lexical features of the text than to the syntactic structure.

The aforementioned works of Tay et al. [21] and Bao et al. [18], the article by Uchendu et al. [19] are consistent, based on similar assumptions. Bakhtin et al. investigated energy-based models and the opportunities of their application to the problem of text generation [22]. They worked on developing a method for improving the quality of such models, automating their self-tuning in the generation process. To generalize their own method, they considered the problem of identifying machine-generated text and the influence of configurations of such models on the complexity of determining that the text was machine-generated. Classical machine learning methods, which are effective for solving classification problems: support vector machine, logistic regression, naive Bayesian classifier, etc., can also be used for the bot detection task. GPT-2 developers show that the baseline model of logistic regression classification and TFIDF vectorization shows high results for smaller versions of the GPT-2 model [16]. This baseline is quite difficult to surpass, but it is worth noting that this approach does not take into account the quality of the text. The advantage of applying the method of evaluating coherence, which was proposed in [1], to the problem of identifying automatically generated text is in the method’s adaptability. It evaluates the general characteristics of the text, this method is generalizable. Coherence is a key indicator of the quality of the text, its semantic correctness, ease of human perception, reflects the correctness of the main idea, the integrity and consistency of presentation, the applicability of the text to solve human problems.

5. Available datasets

One of the most well-known and effective models of natural language text generation is the GPT family. These models, especially GPT-3, allow one to generate texts that are very similar to those written by humans. That is why the methods for identifying that the text was generated by one of these models are especially interesting and important. To train such a model, it is important to have access to datasets of the generated texts, and, preferably, the human-written texts, on which such models have been trained. OpenAI, a company, which develops GPT, has provided access to a set of documents of the WebText dataset, on which GPT-2 has been trained, and for each such text, versions of the text generated by the GPT-2 model with different settings are given [16]. This dataset has been used in this work. There is also a dataset, for which researchers Solaiman et al. finetuned the GPT-2 model on the Amazon Reviews dataset [16]. The initial goal was to "teach" GPT-2 to generate more natural reviews and comments. The difficulty with this dataset is that such comments are short in length, often lacking significant context. Most of the models of bot detection analyzed in the previous section show the worst results on short texts.

Another example of a dataset for training models designed to solve the problem of identifying automatically generated text is a combination of a RealNews dataset and a text set of texts generated by the GROVER model. This model in the original study is trained on the RealNews dataset, which consists of news articles. The model developers provide the part of this dataset that was not used in training separately and examples of the texts produced by the model with certain configurations [23].

Researchers Uchendu et al. have compiled a dataset of human-written news articles on politics [19]. Parts of these articles are used as input for eight language models, including GPT-2, GROVER, XLM, XLNet, PPLM, FAIR.Apart from that, a dataset of Twitter messages generated by bots using different language models: those based on Markov chains, LSTM, basic RNN, GPT-2 and others are available [24]. This dataset is hard to use due to the small length of texts. It is important that these texts are real outputs of social network bots, so the study of such data can give a better understanding of the identification of automatically generated texts, and analysis of the effectiveness of already trained models on this dataset can show the applicability of developed methods.

6. Method of bot detection based on the coherence metric

The task of bot detection is formulated in the following way.

Let T be the considered input text, $T = \{s_i\}, i = 1..N$, where N is the number of sentences in the input text. By having only T and no additional supporting data, such as publishing resource, settings of the account, from which the text has been published, etc., determine which class the text belongs to: automatically generated or human-written texts.

A metric of coherence presented in [1] is used for defining a set of features for classification. This metric is defined as

$$Coherence(T) = \frac{\sum_{i=1}^{n-1} FRel(s_i, s_{i+1})}{n-1},$$

where T is the considered text;

s_i is the i^{th} sentence of the text.

This metric is built based on $FRel$ metric, which is defined as follows:

$$FRel(S_1, S_2) = \alpha \times FRelPred(S_1, S_2) + (1 - \alpha) \times FRelArgs(S_1, S_2)$$

where α is a balancing coefficient, according to Basile et al. [14], the optimal value of α is 0,5.

The value of this metric depends on the values of its two components. The numeric representation of relatedness of predicates is calculated as

$$FRelPred(S_1, S_2) = log_2 \frac{|Cp_1p_2|}{|Cp_1||Cp_2|},$$

where c_{p_1} and c_{p_2} are subsets of sentences from the corpus that have common predicate p_1 from the first sentence S_1 and p_2 with the second sentence S_2, respectively. Cp_1p_2 is a subset of the adjacent sentences in the corpus, where p_1 and p_2 are the main verbs-predicates of the first and the second sentences respectively [1].

The second component is relatedness between predicate arguments

$$FRelArgs(S_1, S_2) = \frac{1}{2}\left(\frac{1}{|\mathrm{arg}s_1|}\sum_{\mathrm{N}_i \in \mathrm{arg}s_1} \max_{\mathrm{N}_j \in \mathrm{arg}s_2} wpsim(\mathrm{N}_i, \mathrm{N}_j) + \frac{1}{|\mathrm{arg}s_2|}\sum_{\mathrm{N}_i \in \mathrm{arg}s_2} \max_{\mathrm{N}_j \in \mathrm{arg}s_1} wpsim(\mathrm{N}_i, \mathrm{N}_j)\right),$$

where $wpsim$ is the Wu-Palmer similarity, $\mathrm{arg}s_1$ and $\mathrm{arg}s_2$ are the sets of noun arguments of predicates p_1 and p_2, respectively.

WebText dataset together with outputs of GPT-2, which has been trained on the WebText data, are used as samples for training and a separate subset is used as testing data. Both datasets have been released by OpenAI and are publicly available [16]. Researchers from OpenAI also mention the difference in amounts of different parts of speech between texts scraped from the web and those generated by different models. After our experiments and hyperparameters tuning, the best results with the defined features have been achieved by XGBoost model. The considered features include maximum, minimum, and average values of the coherence metric [1] for all the pairs of consecutive pairs of sentences in a text, number of nouns, adjectives, adpositions, and verbs. The choice of features is explained by the following. The average value of the coherence metric is an indicator of global connectivity, the overall consistency of all parts of the text.The low minimum value of the metric may indicate the presence of a semantic gap in the text. This may indicate a change of a subtopic in the text, an abrupt and inappropriate change of the topic. Low value also shows the inconsistency of different parts of the text. The high maximum value shows the quality of the text, the consistency of its components and ease of perception. The distribution of nouns, adjectives, verbs and prepositions is considered in comparison for automatically generated and human-written texts (as for the used datasets WebText and GPT-2 output) in **Error! Reference source not found. Error! Reference source not found.**. A feature importance chart is given below in Figure 3. XGBoost model [25] has been used. Feature importance for this algorithm depends on how often a feature has been chosen as the one to make a split on. XGBoost is based on gradient boosting of decision trees. TheFigure 4 shows an example of a tree that is built as one of the estimators in the process of the algorithm execution (the reduced version of the tree is given for illustration). The model is flexible and allows one to adjust it to a specific task because it is possible to set hyperparameters, such as learning rate and number of trees, to set restrictions on the criterion of branching and the maximum depth of trees, to tune other parameters of optimization The model described above has been implemented using Python (version 3.6).

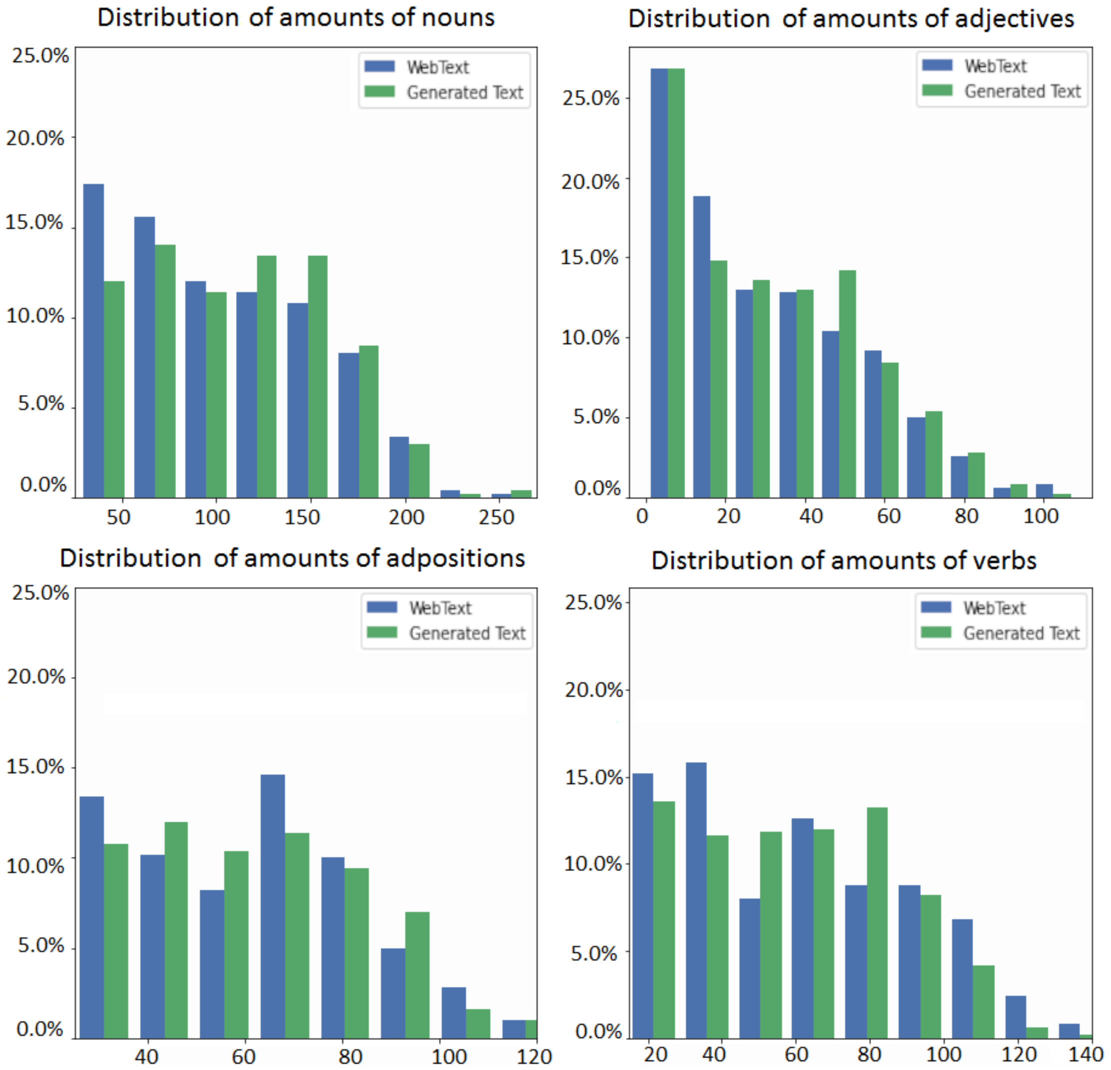
Table 1
Results

Metric	Result
Accuracy	0.7
F1 score	0.67
Recall	0.613
Precision	0.74

WordNet has been used along with Stanford parser, which is implemented as a part of Stanza library [26]. ROCStories corpus [27] has been used as a basic dataset for training the coherence metric. The ROCStrories dataset consists of five-sentence stories. Each of the stories describes trivial situations in people's lives that occur daily, the texts contain common words, simple sentences are used. This dataset has been proposed for use in evaluating systems for natural language understanding and for comparing systems in the Story Cloze Test [27]. The model achieves the following results.

7. Possible improvements and discussion

The described model for bot identification can be extended, new characteristics of the text can be added as features for classification. An advantage of the coherence metric is an opportunity to use it for texts written in different languages. An ontology, a syntax parser, and a training corpus of plain texts are needed. It is worth mentioning, that all of these tools are available for Ukrainian, namely UkrWordNet [28], POS taggers and tools for syntax analysis, different corpora are available and are being developed [29]. This way, it is possible to develop the described system for Ukrainian by retraining the existing model. The developed model of bot detection shows that this metric [1] is easily adaptable and can be used for different tasks which require analyzing semantic properties of text, its quality. The study of identification methods can be the basis for improving the methods of natural language texts generation.


Distribution of amounts of nouns

Distribution of amounts of adjectives

Distribution of amounts of adpositions

Distribution of amounts of verbs

Figure 2: Distribution of nouns, adjectives, verbs and prepositions for human-written and automatically generated texts

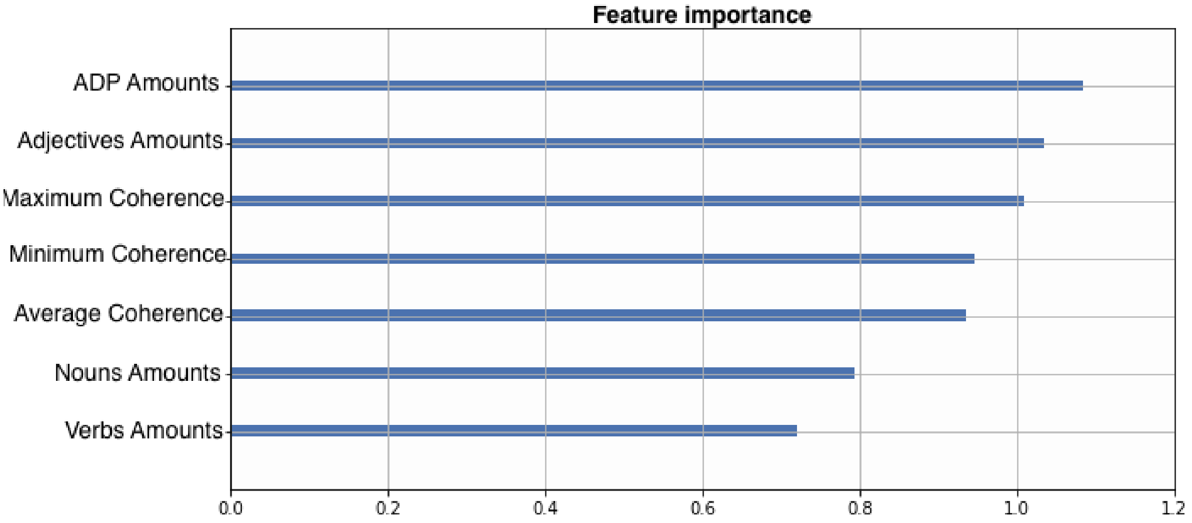
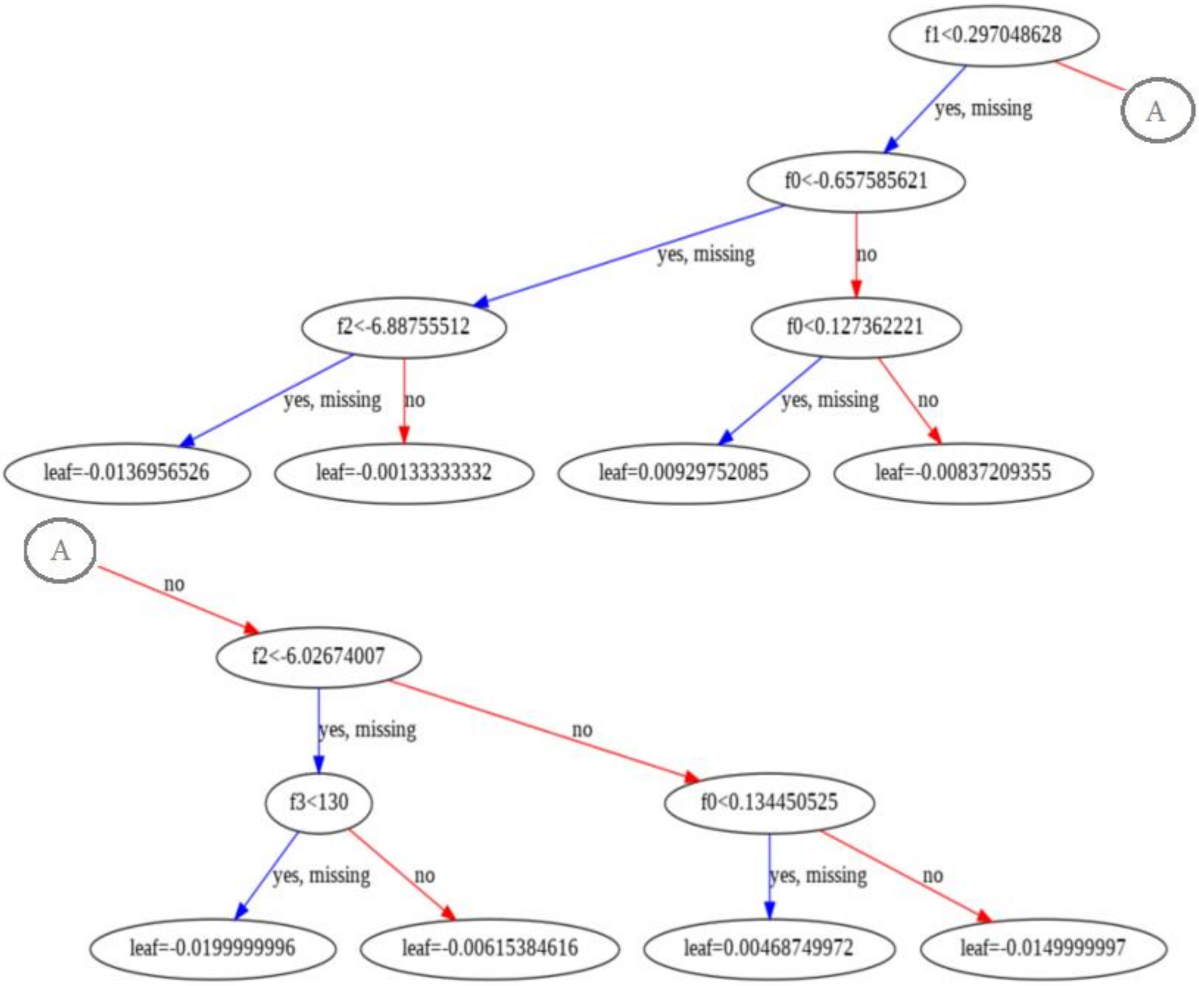

Feature importance

Figure 3: Feature importance based on gain characteristic



.**Figure 4**: An example of a tree that is built as one of the estimators

By understanding what makes a text similar to one written by a human, the best quality of automatic natural text generation is achieved. The process of text generation by humans is complex, it is not fully understood yet, the same is true about the process of perceiving texts. Different researchers agree that coherence is an important aspect in the context of ease of perception. That is why it is required to study coherence first and to include some mechanisms of maximizing coherence into the generation architectures for the development of efficient natural language generation systems.

8. References

[1] O. O. Marchenko, O. S. Radyvonenko, T. S. Ignatova, et al., Improving Text Generation Through Introducing Coherence Metrics, Cybernetics and Systems Analysis 56 (2020) 13–21. doi:10.1007/s10559-020-00216-x.

[2] E. Reiter, R. Dale, Building applied natural language generation systems, Natural Language Engineering 3 (1997) 57 – 87. doi:10.1017/S1351324997001502.

[3] C. Kiddon, L. Zettlemoyer, Y. Choi, Globally Coherent Text Generation with Neural Checklist Models, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.

[4] W. S. Cho, P. Zhang, Y. Zhang, X. Li, M. Galley, C. Brockett, M. Wang, J. Gao, Towards Coherent and Cohesive Long-form Text Generation, in: Proceedings of the First Workshop on Narrative Understanding, 2019, pp. 1–11. doi: 10.18653/v1/W19-2401

[5] Coherent, Cambridge dictionary. URL: https://dictionary.cambridge.org/dictionary/english/coherent

[6] I. Sevbo, On the study of the structure of the coherent text, in: Linguistic research on the general and Slavic typology, Science, 1966.

[7] M. Lapata, R. Barzilay, Automatic Evaluation of Text Coherence: Models and Representations, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, 2005.

[8] D. Lin, LaTaT: Language and text analysis tools, in: Proceedings of the 1st International Conference on Human Language Technology Research, 2001, pp. 222–227.

[9] T. Briscoe, J. Carroll, Robust accurate statistical annotation of general text, in: Proceedings of the 3rd International Conference on Language Resources and Evaluation, 2002, pp. 1499–1504.

[10] N. McIntyre, Learning to Tell Tales: Automatic Story Generation from Corpora, Ph.D. thesis, University of Edinburgh, 2011.

[11] B. Grosz, A. Joshi, S. Weinstein, Centering: A framework for modeling the local coherence of discourse, Computational Linguistics 21 (1995) 203–225.

[12] R. Iida, T. Tokunaga, Metric for Evaluating Discourse Coherence based on Coreference Resolution, in: Proceedings of COLING, 2012, pp. 483–494.

[13] J. Li, D. Jurafsky, Neural net models of open-domain discourse coherence, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2017, pp. 198–209.

[14] V. Basile, R. L. Condori, E. Cabrio, Measuring Frame Instance Relatedness, in: Proceedings of the 7th Joint Conference on Lexical and Computational Semantics, 2018, pp. 245–254. doi: 10.18653/v1/S18-2029

[15] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993-1022.

[16] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, J. Wang, Release Strategies and the Social Impacts of Language Models, 2019. URL: https://arxiv.org/abs/1908.09203.

[17] S. Gehrmann, H. Strobelt, A. Rush, GLTR: Statistical Detection and Visualization of Generated Text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 111–116.

[18] M. Bao, J. Li, J. Zhang, H. Peng, X. Liu, Learning Semantic Coherence for Machine Generated Spam Text Detection, in: Proceedings of the International Joint Conference on Neural Networks, 2019. doi:10.1109/IJCNN.2019.8852340.

[19] A.Uchendu, T. Le, K. Shu, D. Lee, Authorship Attribution for Neural Text Generation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.

[20] W. Zhong, D. Tang, Z. Xu, R. Wang, N. Duan, M. Zhou, J. Wang, J. Yin, Neural Deepfake Detection with Factual Structure of Text, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 2461–2470. doi: 10.18653/v1/2020.emnlp-main.193.

[21] Y. Tay, D. Bahri, C. Zheng, C. Brunk, D. Metzler, A. Tomkins, Reverse Engineering Configurations of Neural Text Generation Models, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 275–279. doi: 10.18653/v1/2020.acl-main.25.

[22] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, A. Szlam, Real or fake? Learning to discriminate machine from human generated text, 2019. URL: https://arxiv.org/pdf/1906.03351.pdf.

[23] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, Advances in Neural Information Processing Systems 32 (2019) 9051–9062.

[24] T. Fagni, F. Falchi, M. Gambini, A. Martella, M. Tesconi, TweepFake: about Detecting Deepfake Tweets, 2021. URL: https://arxiv.org/abs/2008.00036

[25] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[26] M. D. Christopher, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60. doi: 10.3115/v1/P14-5010.

[27] R.Sharma, J. F. Allen, O. Bakhshandeh, N. Mostafazadeh, Tackling the Story Ending Biases in The Story Cloze Test, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 752–757.

[28] A. Anisimov, O. Marchenko, A. Nikonenko, E. Porkhun, V. Taranukha, Ukrainian WordNet: Creation and Filling, in: H. L. Larsen, M. J. Martin-Bautista, M. A. Vila, T. Andreasen, H. Christiansen (eds) Flexible Query Answering Systems. FQAS 2013. Lecture Notes in Computer Science, vol 8132, Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-40769-7_56.

[29] Lang Uk Project, URL: https://lang.org.ua.