TEXT GENERATOR BASED ON LEXIS-SYNTACTIC TEMPLATES

Yodgorov Umidjon,
Senior Lecturer,
Tashkent State University of Uzbek Language and Literature,
yodgorov@navoiy-uni.uz

Abstract. In this paper, we consider a text generator developed using lexis-syntactic templates and compare it with a standard generator based on Markov chains.

Key words: *text generation, natural language processing, Markov chains, LSPL language.*

Абстракт. *В данной работе мы рассматриваем генератор текста, разработанный с использованием лексико-синтаксических шаблонов, и сравниваем его со стандартным генератором на основе цепей Маркова.*

Ключевые слова: *генерация текста, обработка естественного языка, цепи Маркова, язык LSPL.*

Annotatsiya. Ushbu maqolada leksik-sintaksik shablonlardan foydalangan xolda ishlab chiqilgan matn generatori tadqiq etilgan hamda Markov zanjirlariga asoslangan standart matn generatori bilan solishtiriladi.

Kalit so‘zlar: *matn generatori, tabiiy tilni qayta ishlash, Markov zanjirlari, LSPL tili.*

Automatic text generation is one of the most common natural language processing tasks. Text generators are usually used in cases where there are no strict requirements for the quality of the text, and writing it manually takes a lot of time or is an uninteresting task - for example, compiling text summaries of data or writing texts for search engine optimization of web pages.

Among the algorithms for creating texts, two main approaches can be distinguished: methods based on rules allow achieving high quality texts, but require knowledge of the rules of the language and are time-consuming to develop, while methods based on machine learning depend only on training data, but more often they make grammatical and semantic errors in the created texts.

The most common approach to the task of automatic text generation is the use of Markov chains - a probabilistic model that does not depend on the rules of the language and does not require linguistic knowledge. The Markov model refers to machine learning methods.

Rule-based approaches to automatic text generation use lexis-syntactic patterns. This method has been studied in relation to texts in English, the most frequently analyzed in computational linguistics publications. In the case of Uzbek-language texts, lexis-syntactic templates are already used for tasks such as extracting

named entities, but not for creating texts: no works describing the use of templates in Russian text generators have been published at the moment.

In this paper, we consider a text generator developed using lexis-syntactic templates and compare it with a standard generator based on Markov chains.

The texts available on the websites online.navoiy-uni.uz and tsuull.uz were chosen as the material for the text generator.

2. Review of the work on the research topic

Since automatic text generation algorithms in English are widely used in various software systems, there are a significant number of English-language publications describing such algorithms. Here is a brief overview of the articles that is most relevant to the topic.

The most common language-independent approach to automatic text generation is Markov chains. An algorithm based on their application is presented in Sam Zhang [1]. The work carried out by the author includes the implementation of the English text generator in the Python programming language. Various collections of texts in English, including the WordNet corpus, were used as a training sample for the model. To improve the accuracy of the algorithm, the author uses a semantic similarity model in addition to the probabilistic approach.

Another popular machine learning algorithm used to create texts in natural language is the use of conditional random fields. An example of using this approach in the development of an English text generator is presented in Wei Lu et al. [3]. According to the authors, the use of conditional random fields makes it possible to efficiently generate natural language sentences based on their semantic representations. The algorithm proposed in the article consists of two stages. The first stage includes semantic analysis, which allows each sentence to automatically match its semantic representation. At the second stage, the text itself is created based on the obtained semantic representations.

The concept of a lexis-syntactic template as applied to texts in Russian is presented in the work of E.I. Bolshakova and others [2]. The authors give the definition of a lexis-syntactic template as a structural sample of a language construct that reflects its properties, and present their development - the LSPL language, which allows describing text using templates.

3. The purpose and objectives of the work

The purpose of the work is to implement a text generator based on lexis-syntactic patterns and compare it with a standard generator based on Markov chains.

The following tasks can be distinguished in the work on the project:

1. Collect a corpus of texts from online.navoiy-uni.uz and tsuull.uz
2. Develop templates for the main algorithm
3. Implement a template-based feedback generator
4. Implement a feedback generator using Markov chains and train it on the

collected data

5. Compare the results of the two generators

In this project, a rule-based method of automatic creation of texts is used. Lexis-syntactic patterns of texts are set manually and are used by the algorithm to determine the structure of generated sentences. To substitute words into sentences, the algorithm uses a dictionary formed on the basis of a corpus of real texts.

In the modern world, the amount of information is constantly growing, which leads to an increase in demand for automatic means of processing it. Automated text generation software is designed to automate the repetitive and uncreative process of writing supporting text such as summaries, web page indexing by search engines, and so on. However, natural language is difficult to formalize, and algorithms may experience inaccuracies, as automatically generated texts are inferior in quality compared to those created by a person. Research in the field of automatic text generation, aimed at developing new algorithms and improving existing ones, makes it possible to minimize the number of such inaccuracies and to generate texts that fully comply with the rules of the language and do not contain semantic errors.

4. Markov chains

A Markov chain with discrete time is a stochastic system that can be represented as a finite automaton - a set of states and transitions between them. Each of the transitions are assigned a probability value in such a way that for each of the states the sum of the transition probabilities is equal to one. An important property of a Markov chain is the fact that the distribution of transition probabilities from each state depends only on the state itself, but not on those preceding it. Thus, the Markov chain models a random process "without memory".

Markov chains can be used as a probabilistic natural language model. [1] The states of a finite automaton can be either individual words or N-grams - sequences of N words in a row in the text. The application of this model to the tasks of automatic text creation belongs to machine learning methods.

The process of creating text using Markov chains consists of two main stages. At the first stage, the model is trained using a corpus, a collection of texts. The learning process includes the calculation of transition probabilities: for this, the frequency of occurrence of consecutive pairs (or (N+1)-grams) of words in corpus texts is calculated. At the second stage of text creation, the initial state of the constructed finite automaton is selected and then, using a pseudo-random number generator, transitions are made to the next states, each of which adds the next N-gram to the generated text. The transition process is repeated until the desired text length is reached. The accuracy of the algorithm depends on the chosen value of N and on the number of texts in the corpus.

To illustrate that, let's give an example of a Markov chain scheme built on the basis of the input text. A good example is presented in the article [5], where a fragment of the text of song by The Smiths “How Soon Is Now” was used to build

the chain:

"I am the son
and the heir
of a shyness that is criminally vulgar
I am the son and heir
of nothing in particular".

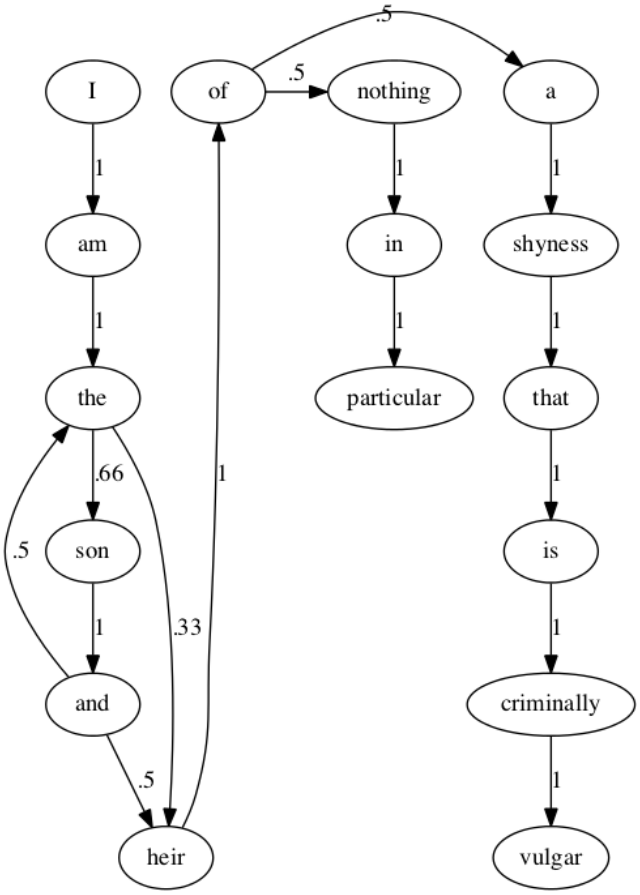

Рис. 1. Пример цепи Маркова [5]

For other languages, the chain is formed in a similar way: it is not the meaning of the words in the text that are important, but their order and frequency.

5. Lexis-syntactic patterns

Lexis-syntactic templates allow you to specify language constructs. A template consists of a logical structure and a semantic description. The use of lexis-syntactic patterns in the task of creating texts in natural language refers to rule-based methods and requires linguistic knowledge. [4, 2] In this paper, the lexis-syntactic template is a sentence scheme consisting of word types. An example of a template that is used in the algorithm presented in this work: a scheme of the form "property

person verb", which corresponds to the fragment of the review "I liked the iron".

The algorithm developed as part of the project is based on a rule-based approach to text processing, combined with the elements of machine learning: sentence templates are built based on manually written rules, and data is used to match them with words in the text creation process. extracted from real texts.

Steps of the algorithm:

0. Lexis-syntactic sentence templates are created manually
1. A list of model characteristics is compiled
2. Verbs and adjectives are extracted from real texts with characteristics
3. Text is formed: words are selected according to templates and using the data obtained in steps 1-2
4. The resulting text is processed in such a way that all neighboring word forms become grammatically consistent with each other

6.Comparison of generators

Generator properties	Template based generator	Generator based on Markov chain
Breaking grammar rules	In a sentence, it is possible only if there are errors in the implementation. With a sufficiently complete analysis of cases of agreement, individual sentences in the output are grammatically correct. But the coordination of proposals among themselves in this version is not provided	Perhaps due to the ambiguity of interpretations of many language constructions
Dependence on the size and variety of input data	Expressed for vocabulary, but not for grammar. The structure of sentences is defined by templates, and vocabulary of the generator is replenished when training on the input data	Expressed for both vocabulary and grammar: the only source for constructing a Markov chain is the input corpus of texts

mismatch of semantic compatibility	Found in early versions of the generator (examples: "long iron", "ceramic instruction"). In the current version, agreement check protects against such situations: phrases encountered in the training corpus entered into the set of valid ones, and when generating text, it is checked whether the current phrase is in this set. A side effect of validation is the fact that it increases dependence on input data and reduces the variety of generated texts.	May occur within a sentence. Example: “The old iron suddenly broke down and I am grateful to him for a sensible offer.”
Variety of language constructs	Low in the current version of the generator. By adding new sentence schemes to the template system, you can increase the variety by bringing texts closer in quality to real reviews. This is one of the possible tasks for further work.	High with a sufficient size of the training sample

7. Conclusion

Based on the results of the work carried out, the following conclusions can be illustrated:

1. Text generator based on lexical and syntactic patterns allows you to get grammatically correct sentences similar to fragments of real texts

2. There is no agreement between individual sentences in the generated texts, therefore, before applying to the creation of voluminous connected texts, the algorithm needs to be improved

3. In the tasks of creating texts for subsequent automatic processing (such as search engine optimization), the proposed algorithm has the same capabilities as the standard generator based on Markov chains

4. The number of grammatical and semantic errors in texts generated using

lexical and syntactic patterns is less than in texts generated using Markov chains

References

1. Sam Zhang. 2009. Natural Language Generation with Markov Chains and Grammar. Computer Systems Lab, 2009-2010
2. Большакова Е. И. и др. Лексико-синтаксические шаблоны в задачах автоматической обработки текстов //Труды межд. конф. Диалог. – 2007. – С. 70-75.
3. Wei Lu, Hwee Tou Ng, Wee Sun Lee. 2009. Natural Language Generation with Tree Conditional Random Fields. Singapore-MIT Alliance. Department of Computer Science, National University of Singapore.
4. Kees Van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real versus Template-Based Natural Language Generation: A False Opposition?. Comput. Linguist. 31, 1 (March 2005), 15-24.
5. How to fake a sophisticated knowledge of wine with Markov Chains. // URL: http://www.onthelambda.com/2014/02/20/how-to-fake-a-sophisticated-knowledge-of-wine-with-markov-chains/