



Dmytro Lande, Leonard Strashnoy

GPT Semantic Networking:

A Dream of the Semantic Web –
The Time is Now

2023

About authors



Dmytro Lande – Professor, Doctor of Science, Head of Department of Information Security Ihor Sikorsky National Technical University of Ukraine. Winner of the Victor Glushkov Prize in Computer Science, and the "Yaroslav Mudry" Prize in Law.



Leonard Strashnoy – Solution Architect for Computer Science Corp, Amazon Inc, and Tata Consulting Services. Certified Amazon AWS Data Architect, Google Certified Digital Cloud Leader.

Dmytro Lande, Leonard Strashnoy

GPT Semantic Networking:

A Dream of the Semantic Web –

The Time is Now

UDC 004.89

Dmytro Lande, Leonard Strashnoy. GPT Semantic Networking: A Dream of the Semantic Web – The Time is Now. – Kyiv: Engineering, 2023. – 168 p. ISBN 978-966-2344-94-3

The book presents research and practical implementations related to natural language processing (NLP) technologies based on the concept of artificial intelligence, generative AI, and the concept of Complex Networks aimed at creating Semantic Networks.

The main principles of NLP, training models on large volumes of text data, new universal and multi-purpose language processing systems are presented. It is shown how the combination of NLP and Semantic Networks technologies opens up new horizons for text analysis, context understanding, the formation of domain models, causal networks, etc. This book presents methods for creating Semantic Networks based on prompt engineering. Practices are presented that will help build semantic networks capable of solving complex problems and making revolutionary changes in the analytical activity.

The publication is intended for those who are going to use large language models for the construction and analysis of semantic networks in order to solve applied problems, in particular, in the field of decision making.

Reviewers:

Waymond Rodgers – Professor, Hull University Business School,
United Kingdom / University of Texas, El Paso, USA

Oleksiy Novikov – Professor, Doctor of Science,
Institute of Physics and Technology
Ihor Sikorsky National Technical University of Ukraine

UDC 004.04

The cover images were generated using the Bing Images Create service (<https://www.bing.com/images/create>) and should symbolize the network structure of the virtual world.

ISBN 978-966-2344-94-3

Copyright © 2023 All rights reserved, Dmytro Lande, Leonard Strashnoy.

Table of contents

PREFACE	6
INTRODUCTION	8
1. ABOUT GPT TECHNOLOGIES	19
LARGE LANGUAGE MODELS.....	19
LLM BRANCHES	27
OVERVIEW OF EXISTING SERVICES BASED ON LLM TECHNOLOGIES	37
<i>ChatGPT</i>	38
<i>Google Bard</i>	39
<i>Character.AI</i>	39
<i>Jasper AI</i>	40
<i>Bing on GPT-4</i>	41
<i>Claude</i>	42
<i>Gptfree.ca</i>	43
2. NETWORKS	49
SEMANTIC NETWORKS	49
COMPLEX NETWORKS	51
<i>Networks settings</i>	53
<i>Global Efficiency Ratio</i>	57
<i>Clustering coefficient</i>	58
<i>Betweenness centrality</i>	59
<i>Modularity</i>	59
<i>Networks phenomena</i>	60
<i>Meanings of "weak ties"</i>	61
<i>Small World</i>	61
<i>The Rich Club Phenomenon</i>	63
<i>Erdős-Rényi random network model</i>	63
<i>Barabashi-Albert Random Network Model</i>	64
<i>Host ranking</i>	65
<i>Algorithm HITS</i>	66

<i>Algorithm PageRank</i>	66
<i>Language Networks</i>	68
SEMANTIC WEB	71

3. NETWORKS ANALYSIS AND VISUALIZATION TOOLS.....77

SOCIAL NETWORK VISUALIZER.....	77
GEPHI	80
<i>General information about Gephi</i>	80
<i>Data Laboratory</i>	82
<i>Overview section</i>	83
<i>Preview section</i>	84
<i>Creating a new graph in Gephi</i>	86
<i>Creating a new graph in Overview mode</i>	86
<i>Creating a new graph in Data Laboratory mode</i>	87
<i>Export graph data from an external file</i>	87
<i>Layout and ranking</i>	89
GRAPHVIZ.....	90
CSV2GRAPH	97

4. BUILDING SEMANTIC NETWORKS WITH CHATGPT101

FORMING A SIMPLE UNDIRECTED NETWORK FORMATION OF NETWORKS OF CHARACTERS.....	103
<i>Forming a simple undirected network</i>	103
<i>Creating a weighted network</i>	104
<i>Creating a directed weighted network</i>	105
<i>Creating a signed network</i>	106
<i>Emulating multiple experts through repeated prompt generation</i> ...	108
FORMATION OF A NETWORK OF CONCEPTS - SUBJECT DOMAIN MODEL	109
<i>Formation of a basic undirected network</i>	109
<i>Emulation of a set of experts by repeatedly generating prompts</i>	111
<i>Hierarchical formation of causal networks</i>	112
FORMATION OF A NETWORK BASED ON SIMPLE HIERARCHICAL ACCESS TO GPT	113
<i>Forming a Network Based on Hierarchical Invocation of Swarm Virtual Experts to ChatGPT</i>	122
<i>Formation of a network based on a generalization of hierarchical querying a swarm of virtual experts to ChatGPT</i>	124

<i>Cause networking techniques</i>	127
<i>Method 1: Repeated execution of the same prompt</i>	134
<i>Method 2: Primary problem decomposition</i>	139
<i>Combining the results</i>	141
BUILDING NETWORKS ON YOUR TEXTS	144
CONCLUSIONS	159
LITERATURE	162

Preface

Dear readers,

we present to your attention the book "ChatGPT Semantic Networking". In this book, we have collected research and practical implementations related to natural language processing (NLP) technologies based on the concept of artificial intelligence and the concept of Complex Networks aimed at creating Semantic Networks.

Part One, "About GPT Technologies," introduces some aspects of how transformer-based language models such as GPT work. We briefly presented the basic principles of NLP and unsupervised model training on large amounts of text data, which allowed us to create universal and multi-purpose language processing systems. A brief overview of tasks and implementations is presented, showing how advanced GPT technologies are transforming language processing. Part two "Networks" is devoted to the modern concepts of Semantic Networks, Complex Networks, and about past successes, problems and relevance of implementing the Semantic Web at a new level. We will talk about the key aspects of such networks, and reveal their connection with NLP. Part three "Networks Analysis and Visualization Tools" will provide you with knowledge of advanced methods for analyzing and visualizing complex networks. You will learn how to create engaging visual representations that will help you better understand the structure and relationships of networks. Part 4, Building Semantic Networks with ChatGPT, shows how the combination of NLP and semantic networks opens up new horizons for text analysis, context understanding, domain modeling, causal networks, and more. This chapter presents methods for creating Semantic Networks based on prompt engineering. Practices are presented that will help build intelligent networks capable of solving

complex problems and making revolutionary changes in the analytical activity.

We sincerely hope that this book will become a source of inspiration and knowledge for anyone interested in cutting-edge NLP, ChatGPT-based machine learning, and the evolution of Semantic Networking. Let it help you see the potential of modern natural language processing technologies and apply modern methods of large linguistic models in analytical work.

Happy reading and inspiration!

Sincerely,

Dmytro Lande,

Leonard Strashnoy,

August 11, 2023

Introduction

This book is dedicated to the task of forming, analyzing and visualizing semantic networks based on the revolutionary technology of large language models.

We will consider the semantic network as a data structure, which is a graph, where the nodes represent entities (concepts, keywords, objects of the real or virtual world), and the edges define the links between these entities. These connections, and edges can be "symmetrical", or non-directional, or they can be directed, expressing some relationships such as "general-private", "cause-effect", etc. Thus, connections indicate relationships between entities. Semantic networks are used to represent knowledge and understand the semantic relationships between different elements.

One type of network is the Semantic Web. On the one hand, it can be seen as a specific web-based implementation of the Semantic Web, and on the other hand, the Semantic Web is a concept that seeks to make information on the Internet more understandable to computers by allowing them to process and link information automatically. In the Semantic Web, information is structured in such a way that relationships can be established between data from different sources, and this contributes to a more intelligent and efficient use of information.

Semantic networks and the Semantic Web play an important role in the development of intelligent applications and search engines, allowing computers to understand and analyze information on web pages more deeply, and enable more efficient use of data in the online environment.

In recent years, the rapid development of large language models has revolutionized natural language processing [Bernard J. Jansen, 2023],

[Wayne Xin Zhao, 2023], [Ce Zhou, 2023]. These powerful models have shown great potential in solving various NLP natural language processing tasks, from natural language understanding (NLU) to generation tasks, even paving the way for artificial general intelligence (AGI). However, the effective and efficient use of these models requires a practical understanding of their capabilities and limitations, as well as the data and challenges associated with NLP [Jingfeng Yang, 2023].

Seemingly completely independent of the Semantic Web, the new GPT (Generative Pre-trained Transformer) technology is designed to solve completely different tasks, for example, text summarization, translation, “creative” generation, and processing of information. At the same time, it is this technology, according to the authors, that can lead to the revival of the Semantic Web at a new level, when it is no longer the voluntary markup of their web pages by users, but the global linguistic processing of the web space that will allow extracting all the necessary data and connections. GPT is a natural language processing technology based on a transformer architecture that learns from a large amount of text data and is capable of generating high quality texts. These models are trained on huge data sets and learn to understand the syntax and semantics of the language, which, in particular, makes them powerful tools for building semantic networks and domain models.

The sensational appearance of ChatGPT in people's daily lives in late 2022-early 2023 is also evidenced by the data of the Google Trends service, which takes into account and processes the dynamics of user requests to the Google search engine (Fig. 1).

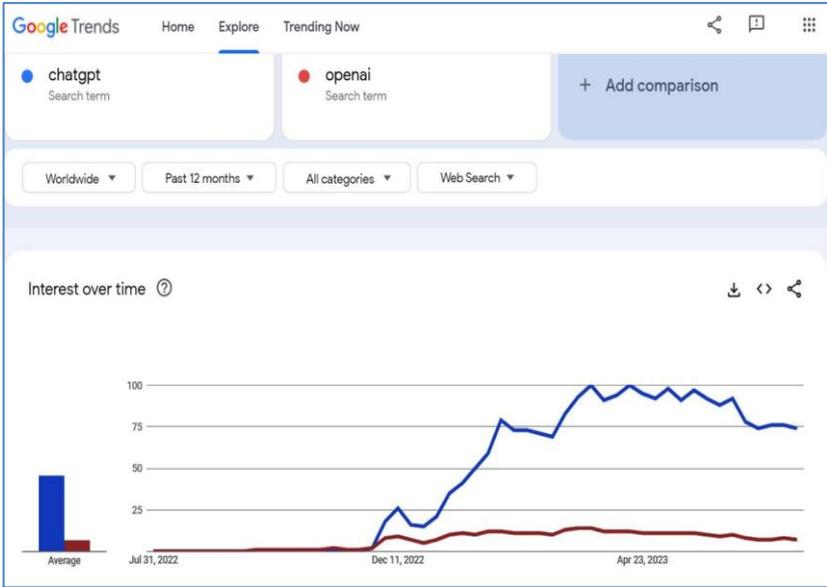


Figure 1. Dynamics of user requests to the Google system containing the words *chatgpt* and *OpenAI* according to the Google Trends service

The ChatGPT model is built on top of the OpenAI GPT-3, GPT-3.5 and GPT-4 family of large language models. The fine tuning of the chatbot was performed using both supervised learning methods and reinforcement learning.

ChatGPT is a project that uses GPT and adds a conversational web interface, memory, and other convenience features to the base model [Aymen El Amri, 2023].

Other notable projects using GPT include, among others:

- GitHub Copilot (using the OpenAI Codex model, a descendant of GPT-3, configured for code generation);
- Copy.ai and Jasper.ai (content generation for marketing purposes);
- Algolia (improving search engine capabilities).

Systems and services like ChatGPT[Stephen Wolfram, 2023] and other modern language models, in just a few covid years, made a revolution in information technology. Although prior to the development of these advanced models, there were other text processing methods such as classical NLP algorithms, regular expressions, and statistical approaches, but they had limited capabilities and could not fully capture the complexities of natural language.

The revolution taking place in the field of large language models is also evidenced by the number of parameters covered by systems created in the past few years, Fig. 2 (<https://cobusgreyling.medium.com/what-are-realistic-gpt-4-size-expectations-73f00c39b832>):

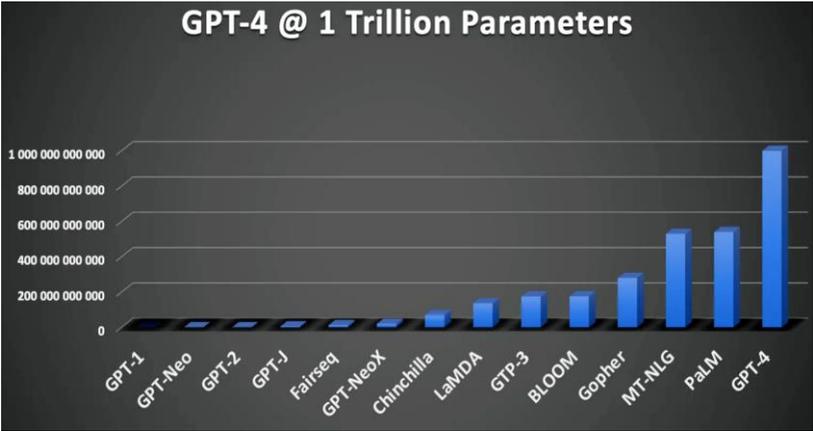


Figure 2. Growth of the parameters of the LLM type system

The volumes of parameters of various systems in tabular form are as follows:

Model	Number of parameters
GPT-1	117 000 000
GPT-Neo	1 300 000 000
GPT-2	1 500 000 000

GPT-J	6 000 000 000
Fairseq	13 000 000 000
GPT-NeoX	20 000 000 000
Chinchilla	70 000 000 000
LaMDA	137 000 000 000
GTP-3	175 000 000 000
BLOOM	176 000 000 000
Gopher	280 000 000 000
MT-NLG	530 000 000 000
PaLM	540 000 000 000
GPT-4	1 000 000 000 000

Currently, we can observe the emergence of such breakthroughs associated with systems such as ChatGPT:

- **Breakthrough in deep learning:** With the advent of deep learning and high computing power, it has become possible to train deep neural networks with billions of parameters. This made it possible to create much more powerful language models capable of processing natural language efficiently.
- **Transformer Architecture:** In 2017, Transformer architecture was introduced, which became the basis for many modern language models, including GPT. The attention mechanism proposed in Transformer made it possible to efficiently process dependencies between words in large text sequences.
- **Pre-training and further training:** Models like ChatGPT are pre-trained on massive amounts of unsupervised data, allowing them to gain a general understanding of the language. These models can then

be further trained on specific data or tasks, making them versatile and applicable in various fields.

- **Availability of data and resources:** The development of the Internet and the availability of large volumes of textual data have become an important factors for the learning and development of language models. Companies and researchers have access to a variety of textual sources, which allows them to train models on huge amounts of data.

Artificial intelligence, in particular systems like ChatGPT, is becoming commonplace now for several reasons:

- **Performance improvement:** Breakthroughs in the field of deep learning and the development of more efficient algorithms have allowed the creation of powerful and fast artificial intelligence models and systems.
- **Big Data:** In recent years, the amount of available data has increased significantly, which has become a key factor for the training and development of AI systems.
- **Infrastructure Availability:** The advent of cloud computing and the availability of powerful computing resources have made the development and application of artificial intelligence more accessible.
- **Integration into various areas of life:** AI is increasingly being integrated into our daily lives through voice assistants, process automation, recommender systems, and context-sensitive, personalized content.

As a result, the improved accessibility of artificial intelligence has made it commonplace in various fields and has improved the user experience in many aspects of life.

Recently, systems and services like ChatGPT and other modern language models have provided a breakthrough in information technology in a number of key aspects:

- **Natural language and human-machine interaction:** Models based on NLP technologies have significantly improved the interaction between humans and computers. They enable more natural communication with machines through voice assistants, chatbots, and speech-based and text-based interfaces, making communication with computers easier and more convenient for users.
- **Automatic translation and multilingualism:** Modern language models, including ChatGPT, have significantly improved the quality of automatic text translation between different languages. This has become especially important for global businesses, linking cultures and enhancing the user experience in different language contexts.
- **Text and content generation:** Systems like ChatGPT are capable of generating readable and coherent text, which has become useful in various areas such as creating content for websites, advertising materials, generating responses to user questions, and even writing articles, reviews, and other textual material.
- **Big Data Processing:** Modern language models can be trained on specific data, which makes them universal and applicable in various industries. This facilitates the development of personalized solutions and the integration of NLP into many applications and systems.
- **New features and creativity:** Models like ChatGPT allow researchers and developers to create new creative NLP applications and research, including music generation, character creation, dialogue simulation, and more.

Specifically, the ChatGPT system developed by OpenAI is one version of the GPT (Generative Pre-trained Transformer) model and has become a significant advance in natural language processing (NLP) technologies. GPT is a deep learning scaling model based on the Transformer architecture and was built with a lot of data and powerful training on large computing resources. The capabilities and applications of ChatGPT, as well as other versions of GPT, have revolutionized NLP technology and practice for several reasons:

- **High performance in language tasks:** The GPT model has millions of parameters, which allows it to train on large amounts of textual data and capture complex language dependencies. This results in improved performance in many language tasks such as machine translation, sentiment analysis, question answering, text generation, and more.
- **Pre-training on a large amount of text data:** Before the GPT model became available for use, it was pre-trained on a huge amount of textual data from the Internet. This pre-training step allowed the model to "understand" language as a reflection of the world at large, making it a versatile and multi-purpose tool.
- **Learning without a teacher:** Since GPT is pre-trained on huge unsupervised data, it is able to generate texts and perform tasks without the need for large and expensive labeled data, making it a cost-effective option for many applications.
- **Automatic translation and text generation:** ChatGPT and other versions of GPT have been successfully used to automatically translate text between languages and generate quality text.
- **Personalized Experience:** ChatGPT can be further trained on specific data or contexts, allowing you to create custom models for

specific applications and industries such as tech support, education, healthcare, and more.

Semantic networks can be integrated with NLP technologies, in particular, with ChatGPT to improve its understanding of textual information and create context-sensitive semantic models.

Here are a few tasks where these two technologies can be integrated:

- **Context Understanding:** Semantic networks help ChatGPT understand the relationships between different concepts in a text. This allows the model to better understand the context and correctly interpret the meanings of words and sentences.
- **Semantic Search:** Semantic networks provide an efficient mechanism for semantic search. The model can use semantic relationships in the network to find more accurate and query-related results.
- **Context-Sensitive Response Generation:** Integrating Semantic Networks with ChatGPT allows the model to generate responses given the context and meaning of previous messages, making the conversational system more natural and informative.
- **Semantic schemas for training:** Semantic networks can be used to create semantic schemas that help structure and represent data for model training, simplifying and speeding up the learning process.

Currently, there are a sufficient number of trained systems such as GPT, which will provide us with a valuable resource for building semantic networks, domain models and ontologies. The formation of semantic networks using GPT technologies can be performed in several stages:

- **Entity extraction:** With the help of pre-trained GPT models like GPT-3.5, and GPT-4, you can efficiently extract entities from text data. GPT-based Named Entity Recognition (NER) systems allow

you to accurately define and classify named entities such as people, organizations, locations, and other key domain concepts.

- **Definition of links:** with GPT models, it is also possible to define semantic relationships between entities and understand the contextual relationships between them. This allows you to identify similar concepts, as well as establish relationships such as "is", "refers to" and others.
- **Building semantic networks:** The collected entities and relationships can be used to build semantic networks. Each entity becomes a node of the graph, and connections become edges, which allows you to visualize the relationship between elements.
- **Ontologies and domain models:** Pre-trained GPT models can be used to create and extend ontologies and domain models. Understanding the semantic relationships between concepts helps to clarify the structure of the ontology, which facilitates the classification and search for information within the subject area.
- **Knowledge base expansion:** Thanks to access to many trained GPT systems, it is possible to supplement the knowledge base of the semantic network with new relationships and entities, which makes it more extensive and informative.
- **Clarification of requests and responses:** Semantic networks based on GPT technologies can be used to process and refine user requests. This allows you to better understand the context of the request and provide more accurate and relevant responses.

It should be noted that although pretrained GPT models provide powerful tools for creating semantic networks and domain models, it is also important to consider the limitations of such systems. For example, the accuracy and completeness of the analysis may be limited by the quality of

the training data and the degree of correspondence of the text to the subject area. Therefore, the integration of GPT technologies with other methods and data quality control are important aspects for the successful construction of semantic networks and domain models.

The integration of semantic networks with ChatGPT allows you to create more intelligent and context-sensitive natural language processing systems, which leads to more accurate and informative results in various NLP tasks.

1. About GPT technologies

Large Language Models

Systems like ChatGPT are commonly referred to as Generative Language Models (GLM), Large Language Models (LLM), or simply Language Models (LM). These terms are widely used in scientific and technical circles to refer to systems capable of generating text based on input and context.

The foundations and principles of systems like ChatGPT are based on advanced AI techniques. These systems use deep learning and generative language models to create human-readable responses to text queries. They train on large amounts of data to understand the context and generate relevant and informative responses. The operating principles of such systems include text comprehension, response generation, and improving the quality of the model through feedback and additional learning. These fundamentals and principles allow systems like ChatGPT to achieve a high degree of automation and efficiency in processing text queries.

In systems like ChatGPT, "text comprehension" refers to the ability of the model to parse and interpret input text to correctly understand its meaning and context. This includes recognizing keywords, phrases, and sentences and understanding the relationships between them.

The GPT is a *generative* text model. The implementation of text understanding in systems like ChatGPT is based on deep learning and the use of large amounts of data. The model is trained on a variety of textual sources in order to learn to recognize and analyze various language constructs and semantic relationships.

This model has gained wide recognition in the field of NLP due to its ability to create high-quality and coherent texts that seem natural to human

perception. The model is able to create new text by predicting its continuation based on the received input data.

GPT is a type of neural network based on the Transformer architecture, which is a form of neural network optimized for sequential data such as texts. Transformer is based on a sequence of self-attention blocks that allow the model to analyze the context and determine the relationships between different words or tokens in the input text. This significantly improves the model's ability to process long texts and understand their meaning.

The self-attention mechanism used in GPT and other NLP models allows the network to pay more attention to important words or phrases in context, resulting in more accurate predictions. This mechanism becomes especially important in text generation, as it allows the model to create sequences of words that are logically related to the previous context and make sense within the entire text.

One of the features of the GPT is its ability to work in "independent learning" or "unsupervised learning" environments. This means that the model does not require well-defined input-output pairs, but can learn the structure of the language from a set of texts without explicit training on specific tasks. This opens up the possibility of using GPT in various NLP tasks without having to retrain each of them.

ChatGPT has been specifically designed as a general-purpose language model, making it a flexible tool for a variety of natural language processing tasks. It can be used to automatically translate texts into multiple languages, create short summaries of large texts, generate responses to user questions, and create descriptions and synthesize texts in other applications.

By default, the GPT model does not store information about previous inputs, it does not have memory for transferring data between successive requests or text blocks. When GPT generates text, it doesn't know ahead of

time what should come next based on previous data. Instead, it generates the next word each time given a probability based on the context that was provided in the previous input. Thus, each word in the text is determined independently of each other, taking into account the previous context, which can sometimes lead to unexpected and not always logical results.

The developers at OpenAI pre-trained the ChatGPT model, but you can create your own model using the fine-tuning procedure on your own datasets.

To achieve text understanding, the model can use methods such as vector representation of words, recurrent neural networks (Recurrent Neural Networks, RNN), transformers and other deep learning architectures. These methods allow the model to capture contextual dependencies and build an internal representation of the text that can be used to generate relevant and informative.

Recurrent Neural Networks are a type of neural network designed to process sequential data such as texts, time series, speech, etc. They have the ability to store information about previous states and use it to process the next elements of the sequence, which makes them especially suitable for tasks where the spatial structure of the data is important.

The main idea of the work of recurrent neural networks is to add feedback (cycles) between nodes in the network, which allows you to transfer information from one time step to another. As an example, imagine a sentence in which each word is processed sequentially, and the context of previous words helps to understand the meaning of the current word.

Main components of recurrent neural networks:

- Recurrent Layer: The main element of an RNN is a recurrent layer consisting of neurons that exchange information with each other within a data sequence. The inputs at each time step are fed

into the input of the recurrent layer, and the outputs at each step are passed to the next time step and used as the hidden state.

- Feedback weights: Recurrent neural networks have weights that allow you to determine what information will be transferred between time steps. These weights are network parameters and are updated during training based on the problem to be solved.
- Activation function: Typically, recurrent layers use an activation function for internal neurons such as hyperbolic tangent (tanh) or recurrent long and short term memory cells (LSTM) and control gates (GRU).

Application of recurrent neural networks:

- Text processing: RNNs are widely used in text processing tasks such as machine translation, sentiment analysis, text generation, and more.
- Time series: RNNs are also applied to time series forecasting such as weather forecasting, financial markets, etc.
- Speech Recognition: In speech recognition tasks, RNNs can be used to analyze audio recordings and determine the meaning of speech.
- Music Generation: RNN can be applied to generate musical compositions or continue melodies.

Recurrent Neural Networks are widely used in generative language models due to their ability to take into account the context and sequence of data. An RNN consists of repeating blocks that allow you to model dependencies between previous and current inputs. In generative language models such as ChatGPT, RNN can be used to generate sequences of words or characters that form natural-sounding responses to text queries. The RNN stores information about previous inputs and uses it to generate the next

character or word in the sequence. The use of RNN in generative language models allows the model to capture long-term dependencies in the text and produce more coherent and better responses. However, RNN also has its limitations, such as the fading or bursting gradient problem, which can make it difficult to train the model on long sequences. In general, the use of RNN in generative language models allows you to create better and more natural text responses, given the context and sequence of data.

Convolutional Neural Networks (CNNs) are a type of neural network specifically designed to process and analyze spatially structured data such as images and videos. They have been widely used in the field of computer vision, but are also successfully used in other tasks where there are local dependencies and data patterns.

The main idea behind the work of Convolutional Neural Networks is the use of specialized layers that convolve and combine data to extract local features from the input data. These features are then fed to the next layers for higher level analysis.

Main components of convolutional neural networks:

- Convolutional Layers: Convolutional layers contain a set of filters (kernels) that traverse the input data and perform convolution, which allows you to highlight local patterns and features. Each filter is a small matrix of weights that is multiplied with the corresponding pixels in the input.
- Pooling Layer: The pooling layer reduces the dimension of the feature map by removing redundant information and increasing invariance to small transformations. The most common pooling method is the Max Pooling layer, which selects the maximum value from each area.

- Fully Connected Layers: After several convolutional and pooling layers, the resulting features are flat-extracted and passed to fully connected layers, which perform the final classification or regression.

Application of Convolutional Neural Networks:

- Image Classification: Such networks are widely used to classify images into various classes, such as object recognition, animal and plant identification, etc.
- Object Detection: Convolutional Neural Networks are used to detect and extract objects in an image, such as faces, cars, or traffic signs.
- Semantic Segmentation: CNNs are also used to segment an image, i.e. dividing it into semantic areas like roads, trees, sky, etc.
- Video Processing: Convolutional Neural Networks can be extended to process video data, including motion analysis and action detection.

Convolutional Neural Networks, with their ability to automatically extract hierarchical features from input data, have become a powerful tool in the analysis and processing of spatial information, and their success has had a significant impact on various fields, including medical diagnostics, autonomous driving, social media analysis, and more.

Convolutional Neural Networks are commonly used in text processing and natural language processing (NLP) tasks to work with data sequences such as sentences and texts. They are used as part of large linguistic models to extract features from textual data and solve various NLP problems.

There are several ways to use convolutional neural networks in large linguistic models:

- Text Classification: CNN can be used for text classification, such as determining the sentiment of reviews, determining the category of

text, or identifying a language. In this case, convolutional layers can extract local contextual features from sentences, and after combining layers, you can reduce the dimension and pass the resulting features to fully connected layers for classification.

- Feature Extraction: In large linguistic models such as Word2Vec, GloVe, and FastText, convolutional layers can be used to extract features from text or vectorize words. They can slide through the sequence of words, detecting local patterns and creating vector representations of the words.
- Semantic segmentation: Convolutional Neural Networks can be used for semantic text segmentation, that is, the selection of certain fragments of text associated with certain semantic units, for example, the selection of named entities.
- Answering Questions: In question-answer problems, CNN can be used to find answers to questions in a text. They can highlight keywords or phrases that can serve as answers to a given question.
- Text generation: Some linguistic models, such as TextCNN, use convolutional layers in combination with recurrent layers or transformers for text generation, such as article generation, question answering, and text description tasks.

In large linguistic models, Convolutional Neural Networks are often used in combination with other types of layers such as Recurrent Neural Networks (RNNs) or Transformers to achieve better performance on NLP tasks. The combination of different types of layers allows the model to effectively capture different aspects of context and dependencies in text.

The "Transformer" architecture is a revolutionary model in the field of natural language (NLP), which was presented in the article [Ashish Vaswani, 2017]. and others.

The main idea behind the Transformer is to use the Attention mechanism to efficiently analyze dependencies between words (or tokens) in input sequences. It differs from traditional Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) in that it does not use sequential data processing. Instead, it analyzes the entire sequence of data in parallel, which makes it significantly faster and allows it to process long texts.

The main components of the Transformer architecture:

- Attention Mechanism: The basic idea of attention is to focus on the most important parts of the input while performing a task. The Transformer model uses the "Scaled Dot-Product Attention" attention mechanism, which allows the model to determine how important each token is to the processing of other tokens in the input sequence.
- Encoder and Decoder: The Transformer architecture consists of a stack of repeating blocks called encoders and decoders. Encoders convert the input sequence into an internal representation (context vectors), and decoders use this representation to generate the target sequence.
- Normalization Layers: Each encoder and decoder uses a normalization layer, such as a Batch Normalization layer, to stabilize and speed up the learning process.
- Sub-layers: Each encoder and decoder block consists of several sub-layers such as multi-headed attention layers and fully connected layers.

The following applications of the Transformer architecture in generative models are known:

- GPT (Generative Pre-trained Transformer): GPT uses Transformer for language modeling and text generation. It can be trained on various NLP tasks and has been successfully applied to various language tasks.
- BERT (Bidirectional Encoder Representations from Transformers): BERT is a Transformer-based bidirectional encoder that trains on a large amount of text data and creates contextual representations of words given their surroundings.
- T5 (Text-to-Text Transfer Transformer): T5 is a single multipurpose modeling framework where all NLP tasks are formulated as text inputs and text outputs. It is successfully used in machine translation, sentiment analysis and other tasks.

The Transformer architecture has been the basis for much cutting-edge natural language research and continues to be a vibrant field in machine learning.

LLM branches

At the moment, there are several directions in the field of language models of artificial intelligence:

- Text Generation: This area involves the development of models capable of generating text based on given inputs. This can be useful for creating automated responses to questions, generating content for websites, or creating conversational systems.
- Language Translation: Here models are used to automatically translate text from one language to another. This can be useful for various tasks related to interlingual communication.
- Text analysis: This area includes the development of models capable of analyzing and understanding text. This can be useful for tasks

such as text classification, information extraction, or text sentiment detection.

- Conversational Systems: Here, models are used to create systems capable of engaging in natural language dialogue with users. This can be useful for creating virtual assistants or chatbots.

To implement these capabilities, there are currently several branches of the implementation of LLM technologies (language models based on artificial intelligence). The most famous of them are:

- OpenAI GPT (Generative Pre-trained Transformer): This is one of the most famous and widely used LLM implementations. OpenAI GPT provides powerful text generation and natural language understanding capabilities.
- Google BERT (Bidirectional Encoder Representations from Transformers): This model is developed by Google and trained on large amounts of text data. BERT allows you to understand the context and meaning of sentences, which makes it useful for natural language processing tasks.
- Microsoft Turing is a 17 billion parameter model developed by Microsoft in 2020 that also provides text generation and natural language understanding capabilities. Turing is trained on large amounts of data and can be used for a variety of language-related tasks.
- Facebook RoBERTa (Robustly Optimized BERT Pretraining Approach): This is an improved version of the BERT model developed by Facebook. RoBERTa trains on large amounts of data and performs well when fine-tuning to the specifics of English text arrays when solving problems such as sequence classification, token classification, or question answering.

[Jingfeng Yang, 2023] presents an evolutionary tree of modern LLMs that traces the development of language models in recent years and highlights some of the most well-known models (Fig. 3). In the present paper, systems are classified as "encoder systems", "decoder systems", and mixed systems.

In the context of LLM (Language Model), "encoder system" usually refers to the module or component of the model responsible for converting input data (text) into an internal representation called "encoded representation" or "feature vector". The Encoder system works at the level of words or tokens and converts them into numeric vectors that the model can process. This process is usually based on deep learning methods such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformers. An encoder can be one of the main components of a model, such as GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers) and others.

The task of the encoder system is to capture the semantic information and context of the text and encode it into a dense numeric vector. This allows the model to understand and generate text based on this vector representation.

The task of the encoder system is to capture the semantic information and context of the text and encode it into a dense numeric vector. This allows the model to understand and generate text based on this vector representation.

The encoded representation received from the encoder system is usually passed to other parts of the model for further processing, such as a decoder that generates a response or the next piece of text.

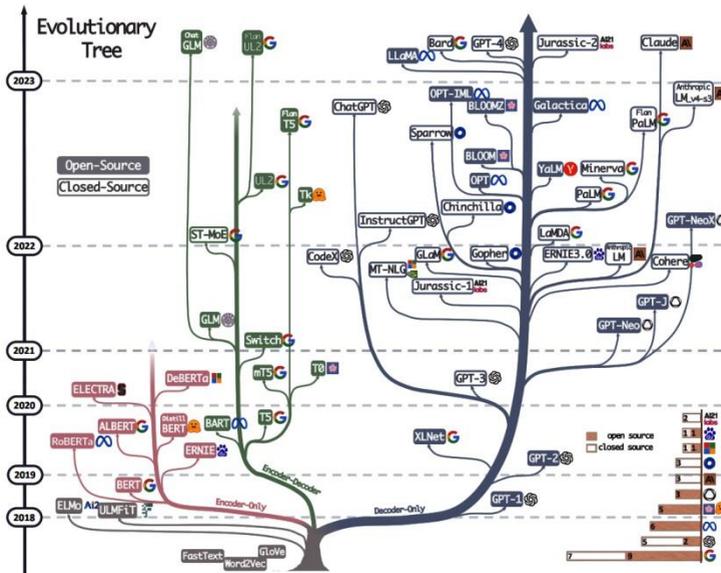


Figure 3. Evolutionary tree of modern LLMs [Jingfeng Yang, 2023]*.

* Models in the same branch have closer relationships. Transformer-based models are not shown in gray: decoder-only models in the blue branch, encoder-only models in the pink branch, and encoder-decoder models in the green branch. The vertical position of the models on the timeline represents their release dates. Open source models are represented by solid squares, while closed source models are empty squares. The stacked bar chart in the lower right corner shows the number of models from various companies and institutions.

Thus, the encoder system is an important part of the LLM model and allows it to understand and work with textual data.

Among the LLM (Large Language Model) there are various types of purely encoder systems. They are typically used to process text data and extract semantic information from text without subsequent text generation. Some of them include:

- Encoder-only Transformer: This is a model based on the transformer architecture that only works in encoder mode. It takes a sequence of words or tokens as input and generates the corresponding encoded representations for each word. The Encoder-only Transformer can be used for tasks such as text classification, information extraction, sentence embeddings, and more.
- Universal Sentence Encoder: This is a model specially designed for extracting universal sentence embeddings. The Universal Sentence Encoder takes a sentence or text as input and generates a dense vector representing the semantic information of that sentence. This can be useful for tasks like comparing sentences, finding similar sentences, or clustering texts.
- BERT (Bidirectional Encoder Representations from Transformers): BERT is a model that operates in encoding mode and uses a transformer architecture. It learns from large text corpora and creates contextual embeddings of words given their context both on the left and on the right. BERT can be used for a variety of tasks, including text classification, sequence tagging, question-answer systems, and more.

The term "decoder system" usually refers to the module or model component responsible for generating text based on an encoded representation or feature vector received from an encoder system.

The Decoder system takes an encoded representation and uses it to generate text that continues or expands on the original text. It works at the

word or token level, converting vector representations into a sequence of words or tokens.

Often a decoder system uses recurrent neural networks (RNNs), including recurrent cells such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit), for sequence modeling and text generation. It can also use attention to focus on different parts of the encoded representation as each new word is generated.

The decoder system usually trains on a large body of text and seeks to predict the most likely next word or token given the previous context and encoded representation. Sequential word prediction allows the model to generate coherent and meaningful text similar to human speech.

Thus, the decoder system is an important part of the LLM model and is responsible for generating text based on the encoded representation received from the encoder system.

A mixed encoder-decoder system in the context of LLM (Language Model) is a model that combines the functionality of encoder and decoder systems in one architecture.

This system consists of two main components:

- Encoder: This is the part of the model that takes text data as input and converts it to an internal representation or encoded representation. Encoder typically uses deep learning techniques such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), or Transformers to extract semantic information from text and generate dense numeric vectors.
- Decoder: This is the part of the model that takes an encoded representation generated by the encoder system and generates a text or sequence of words. Decoder uses Recurrent Neural Networks

(RNN) like LSTM or GRU to model sequences and generate text based on context.

The decoder system in its purest form is responsible for generating text based on a given context or encoded representation. Here are some examples of pure decoder systems:

- LSTM Decoder: This is a model that uses Long Short Term Memory (LSTM) recurrent neural networks as the basis for text decoding. The LSTM Decoder takes an encoded representation or context as input and generates text sequentially, one word or token after another, given the previous context and generated text.
- Transformer Decoder: This is a model based on the transformer architecture where the decoder is responsible for generating text based on the encoded representation or context. The Transformer Decoder uses the attention mechanism to focus on different parts of the encoded representation and generate text sequentially.
- GPT (Generative Pre-trained Transformer): GPT is an example of a pure decoder system based on a transformer architecture. It learns from large corpora of text and predicts the next word or token based on the previous context. GPT can be used to generate text, complete sentences, or continue text.
- Variational Autoencoder (VAE) Decoder: VAE is a model that combines generative models and autoencoders. In the context of LLM, the VAE Decoder is responsible for generating text based on a given hidden representation, which can be obtained using the system's encoder or other methods. VAE Decoder allows you to sample different variants of text based on a given distribution.

Although language models are typically task-independent in their architecture, these methods require fine-tuning of datasets for a specific downstream task. Researchers have found that scaling language models significantly improves performance at little or even zero [Brown Tom, 2020]. The most successful models for best multi-shot, zero-shot performance are autoregressive language models, which are trained by building the next word in a sequence given previous words. These models are widely used for downstream tasks such as text generation and question answering.

Examples of autoregressive language models include GPT-3 [Brown Tom, 2020], PaLM [Barham, P., 2022], and BLOOM [Teven Le Scao, 2022]. The game-changing GPT-3 demonstrated acceptable performance for the first time, thereby demonstrating the superiority of autoregressive language models.

In a mixed encoder-decoder system, encoder and decoder components are combined into a single architecture where information is passed from encoder to decoder to generate text. In this model, an encoder is typically used to encode input data, such as sentences or phrases, into a vector representation. The decoder then uses this representation to generate text, continuing the original context or answering the question asked.

The mixed encoder-decoder system can be trained on a large body of text using deep learning methods and can be used for various tasks such as text generation, machine translation, chatbots and others. It usually requires a large amount of training data and complex optimization algorithms to achieve good results.

In the context of LLM (Language Model) there are various types of encoder-decoder systems that combine the functionality of encoder and decoder in one architecture. Here are some examples of such systems:

- Sequence-to-Sequence (Seq2Seq) models: This is a popular class of encoder-decoder models that are widely used for machine translation, text generation, and other tasks that require input and output sequences. The Seq2Seq model consists of an encoder that encodes input into a vector representation and a decoder that generates output from that representation.
- Attention-based models: These models extend the Seq2Seq architecture by adding an attention mechanism. Attention-based models allow the decoder to pay special attention to different parts of the encoded representation during text generation. This helps the model to use the information from the encoder more efficiently, especially when working with long sequences or in cases where context is important for generation.
- Transformer-based models: The transformer architecture has become the basis for many LLM encoder-decoder systems. Transformer-based models use self-attention mechanisms in both the encoder and decoder to capture the dependencies and context of input and output sequences. These models, such as GPT and BERT, have achieved outstanding results in text generation, language processing, and other tasks.
- Variational Autoencoder (VAE) models: This is a class of models that combine generative models and autoencoders. In the context of LLM, VAE models use an encoder to extract a hidden representation from the input and a decoder to generate text based on this hidden representation. This allows the model to generate different versions of the text and have a more flexible representational capability.

Because natural language data is readily available, and unsupervised learning paradigms have been proposed to make better use of extremely large datasets, this motivates unsupervised natural language learning. One common approach is to predict the masked words in a sentence given the surrounding context. This learning paradigm is known as the masked language model. This type of learning allows the model to gain a deeper understanding of the relationship between words and the context in which they are used. These models have been trained on a large body of texts using techniques such as the Transformer architecture and have achieved state-of-the-art performance in many NLP tasks such as sentiment analysis and named object recognition. Notable examples of masked language models include BERT [Devlin Jacob, 2018], RoBERTa [Liu Yinhan, 2019], and T5 [Raffel Colin, 2020]. MLMs have become an important tool in the field of natural language processing due to their success in solving a wide range of problems.

Several conclusions can be drawn from the evolutionary tree of language models (Fig. 3):

- Decoding-only models gradually came to dominate LLM development. In the early development of LLM, decoder-only models were not as popular as encoder-only and encoder-decoder models. However, after 2021, with the advent of the revolutionary LLM - GPT-3, models with only a decoder experienced a significant boom. Meanwhile, after the initial explosion caused by BERT, encoder-only models have slowly begun to disappear.
- OpenAI has consistently maintained its leading position in LLM, both now and potentially in the future. Other companies and institutions are struggling to catch up with OpenAI in developing models comparable to GPT-3 and the current GPT-4. This leadership

position may be due to OpenAI's unwavering commitment to its technical path, even if it was not initially widely accepted.

- Meta is a significant contributor to open source LLM and promotes LLM research. When considering contributions to the open source community, especially with respect to LLMs, Meta stands out as one of the most generous commercial companies since all LLMs developed by Meta are open source.
- LLMs show a trend towards closed sources. In the early stages of LLM development, most of the models were open source. However, with the introduction of GPT-3, companies have increasingly chosen to close source their models such as PaLM, LaMDA, and GPT-4. Consequently, it has become more difficult for academic researchers to conduct LLM teaching experiments. As a result, API-based research may become the predominant method in the academic community.
- Encoder-decoder models remain promising as this type of architecture is still being actively researched and most of them are open source. Google has made significant contributions to the open source encoder-decoder architecture. However, the flexibility and versatility of decoder-only models seems to make Google's persistence in this direction less promising.

Overview of existing services based on LLM technologies

Here is a list of 7 freely available services built for text generation based on LLM. Here are some free services that can create texts that are as close to humans as possible using artificial intelligence. Nowadays, there are a lot of opportunities in the field of AI text generation, and there are many different tools to choose from. In this book, we've compiled what we think

are the best free AI text generators that offer great features and are easy to use.

ChatGPT



<https://chat.openai.com>

ChatGPT is a chatbot using advanced language models. It is based on the GPT-3.5 model from OpenAI, which is an improved version of GPT-3, which allows it to generate answers to your requests using artificial intelligence. It is suitable for a variety of tasks, including writing texts, summarizing them, extracting named entities, as well as generating templates and even generating code.

The platform is publicly available and has the ability to self-learn. In addition, ChatGPT is also presented as a mobile application.

ChatGPT is currently free to use. An OpenAI account is required to access the tool. To use ChatGPT, it is enough to go through a simple registration on the web platform at <https://chat.openai.com/>, after which you can enter your messages and send them.

In addition, ChatGPT remembers previous requests, which allows you to continue chatting with him, starting from where you left off in previous conversations.

The benefits of ChatGPT are that it is free to use; in constant improvement (new versions of programs, increase in the number of tokens, regulation of "temperature", etc.); supports many natural languages, the ability to understand code in text markup languages and various

programming languages, including HTML, JavaScript, CSS, Python, Perl and others.

Google Bard



<https://bard.google.com>

Google has announced the release of their AI chatbot Bard, which uses the same technology as their AI model LaMDA. Bard is a great tool for creating texts and getting answers. Bard has built-in security and feedback mechanisms.

This chatbot from Google successfully solves creative problems and is an additional tool that complements Google Search. Its interface is simple and easy to use, and the strength of Google Bard is, of course, its use of Google search, which provides access to information from the real world, thus improving the quality of its responses.

At the same time, it is worth noting that Bard is still under development, so the results may not always be accurate. Due to the use of Google search results in data collection, biased responses are possible.

Advantages: very fast text generation for free; access to information from the real world through Google Search; regular updates with information from Google.

Character.AI



<https://character.ai>

Character.AI has the ability to generate text using virtual intelligent agents (characters) created by artificial intelligence.

On the website of the service, after registering an account, you can select the desired character and start a chat with him, and the selected character affects the answers you receive.

For example, the Lambda character, powered by Google AI, can generate results for product descriptions and other content in the form of a summary, and its responses are very realistic. To access all the characters, you need to register a free account. The advantages of Character.AI are the ability to use it for free, choose from an extensive set of characters, get three generations of text in each answer.

Jasper AI



Jasper AI is a specialized artificial intelligence tool for creating original and creative content, giving the user the opportunity to get 10,000 words for free after completing the registration process. Jasper AI works on a pay-per-word-per-month basis.

[https://easywithai.com/
tools/jasperai](https://easywithai.com/tools/jasperai)

The basic starter plan allows you to generate 35,000 words per month, while the boss mode plan allows you to generate 100,000 words per month.

This service demonstrates high efficiency in creating full marketing texts and extensive materials, which makes it one of the most sought-after tools among online marketers. A significant advantage of Jasper AI lies in the various settings that allow the user to change the style and structure of the generated content. Jasper is based on the GPT-3 language model developed by OpenAI, introducing its own unique features and settings.

Jasper AI provides the ability to create texts in a variety of styles, and the software ensures that every sentence flows smoothly and coherently.

Jasper AI has several advantageous features such as the ability to generate long text content, the ability to avoid plagiarism issues and AI-assisted content validation, as well as the presence of advanced settings and features designed to control the output of text.

Bing on GPT-4



<https://bing.com/new>

Anyone can sign up and try out Microsoft's Bing chatbot, based on the new GPT-4 artificial intelligence model. To access the new Bing, you must go to bing.com/new, click the "Join Waitlist" button, and sign in with your Microsoft account. After that, users immediately have access to the new Bing. You may also need to use the Microsoft Edge browser.

Bing Chatbot users can ask 15 questions per session, up to a maximum of 150 per day. It is known that the Bing chatbot has already collaborated with the generative AI model GPT-4 from OpenAI long before its official launch. The next day, the company plans to hold an event during which an add-on for Office that uses AI will be presented. Microsoft is also known to be planning to integrate its own ChatGPT-like chatbot model directly into Office programs such as Teams, Word, and Outlook. Additionally, Microsoft has added its Bing AI chatbot to a new sidebar in the Microsoft Edge browser. This means that users will have easy access to this innovative tool while browsing the web in a browser.

Claude



<https://claude.ai/>

Claude is one of the new free AI-based chatbots developed by Anthropic and actively compared with ChatGPT, Google Bard and other LLM chatbots, allowing you to naturally interact, ask questions, search for information, complete tasks and more.

The update to Claude 2 resulted in a significant improvement in the functionality of the chatbot and the manifestation of many improvements in the field of text generation. A particularly attractive aspect of Claude is that it is free to use, just like ChatGPT. Claude is one of the new free AI-based chatbots developed by Anthropic and actively compared with ChatGPT, Google Bard and other LLM chatbots, allowing you to naturally interact, ask questions, search for information, complete tasks and more. The update to Claude 2 resulted in a significant improvement in the functionality of the chatbot and the manifestation of many improvements in the field of text generation. A particularly attractive aspect of Claude is that it is free to use, just like ChatGPT. Claude is equipped with a feedback mechanism that allows users to correct it in case of errors, which contributes to its gradual improvement over time. Claude learns from previous user interactions to improve his abilities over time. Like ChatGPT, Claude saves previous conversations, allowing you to continue chatting from the breakpoint. Claude service users can upload documents in PDF, txt, and CSV formats, and artificial intelligence can provide content summaries for them.

Gptfree.ca



A family of free information services based on artificial intelligence created for educational purposes.

<https://chatgpt.ca>

Gptfree.ca includes sites:

Gptfree.co – GPT Free – Unlimited AI

Zerogptai.org – ZeroGPT AI

chatg-pt.org – ChatGPT AI Chat

Aitoolsfree.org – AI Tools Free

Chatgtp.ca – ChatGTP Unlimited Free

Chatgpt.one – ChatGBT Free

Chataigpt.org – Chat AI GPT

Aichatai.co – Ai Cat Ai

GPT Free – Unlimited AI



GPT Free Unlimited AI is a free platform with unlimited access to its services. One of the key features of this platform is its ability to learn and adapt to user preferences over time.

<https://gptfree.co>

In the process of interacting with the platform, it collects information about user preferences and uses this information to provide more personalized recommendations and responses. GPT's pre-training takes place

on vast amounts of textual data, allowing it to capture the intricacies of the language and produce coherent and contextually related responses.

ZeroGPT AI



<https://zerogptai.org>

With the increasing use of artificial intelligence algorithms, creating unique and authentic content becomes a challenge. In such circumstances, ZeroGPT AI comes to the rescue, a powerful content generation tool that allows users to create content that is not subject to detection as created by artificial intelligence.

Powered by LLM technologies, ZeroGPT AI ensures the creation of original, high-quality, SEO-optimized and plagiarism-free content.

In addition, ZeroGPT distinguishes between human written text and synthetically generated text. As such, it becomes the ideal solution for companies and individuals who want to create content that engages and effectively reaches their target audience.

ChatGPT AI Chat



<https://chatg-pt.org>

ChatGPT AI Chat is another free chatbot version based on the GPT (Generative Pre-training Transformer) language model, which is a deep learning algorithm developed by OpenAI.

The service uses the GPT-3 model, released in 2020, which contains over 175 billion parameters, making it one of the most powerful language models in existence. ChatGPT is built on the GPT-3 model and has advanced conversation capabilities.

AI Tools Free



<https://aitoolsfree.org>

AI Tools Free are programs or platforms freely available on the site that use artificial intelligence algorithms and techniques to perform various tasks. The AI Tools Free website provides tools designed with artificial intelligence technology that you can use freely.

The instruments include:

- SEO Analysis Free AI Tools;
- Blogging AI Tools For Free;
- Business AI Tools For Free;
- Product Selling Free AI Tools;
- Social Media AI Tools Free:
 - o Facebook Support AI Tools;
 - o LinkedIn Support Free AI Tools;
 - o Instagram Free AI Tools;
 - o TikTok AI Tools Free;
 - o YouTube Support Free AI Tools;
 - o Twitter Free AI Tools;
- Email AI Tools Free;
- Domain AI Tools Free;
- Google Analysis AI Tools Free;
- Food AI Tools Free и др.

ChatGTP Unlimited Free



<https://chatgtp.ca>

ChatGTP is another AI-powered chatbot designed to help users with a variety of tasks. Its capabilities cover a wide range of functions, including answering questions and providing recommendations tailored to the preferences of each user.

ChatGTP is readily available on a variety of platforms and devices, making it easy to use.

ChatGTP AI chatbot has many features to provide support in different sectors. Of particular note is the instant translation service provided by ChatGTP. This functionality allows you to easily communicate with users from different countries, providing the convenience of choosing the language of each user's preference.

ChatGBT Free



<https://chatgbt.one>

ChatGBT is an AI-powered language model designed to interact with users in a conversational manner, provide human-like responses, and support a wide range of queries and tasks.

To achieve this goal, ChatGBT uses deep learning techniques with extensive knowledge to provide consistent and contextually meaningful responses. The ChatGBT model is trained on Big Data from the Internet, which allows it to work with various topics. Chat GBT is aimed at facilitating natural and interactive conversations, making it a valuable tool

for a variety of applications including customer support, content generation, and information retrieval.

Chat AI GPT



Open AI GPT Chat is a freely available artificial intelligence system designed specifically for chatting.

<https://chataigpt.org>

When a user interacts with AI GPT Chat, the system is able to formulate responses based on a deep understanding of the language, which is made possible by the powerful pre-training that the system goes through to increase its level of knowledge and intelligence.

Open AI GPT Chat offers the unique ability to generate meaningful responses that match the context of the conversation. This means that this model is able to understand the context of communication and give answers appropriate to this context.

Ai Cat Ai



Ai Chat Ai is a platform for interacting with the free ChatGPT, a language model developed by Ai Chat Insurance. ChatGPT's capabilities range from creating unique text and translating languages to writing various types of creative content.

<https://aichatai.co>

Ai Cat Ai stands out as one of the advanced language models due to access to extensive training data and the use of advanced algorithms to understand a wide range of queries and provide reliable answers. Ai Cat Ai

is able to understand and respond to queries in multiple languages including English, Spanish, French, German and many more. Ai Cat Ai can provide customized recommendations and advice based on user input and preferences.

2. Networks

Semantic networks

The ability to represent knowledge in the form of semantic networks has made them a valuable tool for natural language processing and understanding the meaning of texts. Semantic networks are structures designed to represent semantic relationships between concepts or elements in the form of a graph. A semantic web is a way of representing a knowledge base where concepts are linked together in the form of a web. The Semantic Web is a tool used in the presence of knowledge, which is best understood as a set of related concepts.

The structure of the semantic network is a graph, where the vertices represent concepts, and the edges represent the semantic links between them [Sowa, 1987], forming semantic fields. The Semantic Web can be implemented, for example, as a graph database or a concept map. Typical semantic networks are often represented as semantic triples.

Semantic networks are used in natural language processing applications such as semantic parsing [Domingos, 2009] and word disambiguation [Sussna, 1993]. Also, semantic networks can be used to analyze large texts in order to identify main topics (for example, social media posts) or to identify bias (for example, in news coverage), as well as for mapping, building a model of the entire subject area [Segev, 2022].

The first computer semantic networks were developed in detail by Richard Richens [Lehmann, 1992] in 1956 as part of the Cambridge Language Learning Center project on machine translation. The process of machine translation is divided into 2 parts: translation of the source text into an intermediate form of representation, and then this intermediate form is

translated into the desired language. Semantic networks were just such an intermediate form.

Most semantic networks are based on the theory of knowledge and consist of arcs and nodes that can be organized into a taxonomic hierarchy. These networks have contributed to the ideas of activation propagation, inheritance, and the representation of nodes as proto-objects.

The process of building semantic networks includes identifying keywords in a text, counting the frequency of their mutual occurrence, and analyzing networks to identify central words and clusters of topics in the network [Segev, 2022].

Mathematics is known for its ability to describe most phenomena in the world around us with the help of logical statements. In turn, semantic networks emerged as an attempt to visualize mathematical formulas representing relationships between objects. The main representation for the semantic web is a graph. However, it must be remembered that there is always a strict mathematical notation behind the graphic image.

The main form of representation of the semantic network is a graph since this is the most convenient form perceived by a person. When the directions of relationships are indicated in the schemes of semantic networks, they are called knowledge maps, and the set of such maps, which allows for covering large sections of the semantic network, is called a knowledge atlas or a semantic map.

In mathematics, a graph is represented as a set of vertices and a set of relationships (connections) between them. From the point of view of mathematical logic, each vertex corresponds to an element of the subject set, and the arc corresponds to a predicate.

In linguistics, relationships are recorded in dictionaries and thesauri. In dictionaries, in definitions through genus and specific differences, the

generic concept occupies a certain place. In thesauri, in the article of each term, all possible connections with other related terms can be indicated.

In semantic networks, two important aspects can be distinguished: separation by arity and by the number of types of relations.

By the number of types of relationships, networks can be homogeneous or heterogeneous. Homogeneous networks contain only one type of relationship; in heterogeneous networks, the number of relationship types exceeds one. Such networks are of great practical interest but also represent more complex research problems. Heterogeneous networks can be represented as an interweaving of tree-like multilayer structures.

By arity, networks can be binary or N-ary. Binary relations connect exactly two concepts and are conveniently represented on the graph as an arrow between two concepts (such networks will be studied in this paper). But sometimes there is a need for relationships that connect more than two objects, such relationships are called N-ary. Their representation on a graph can be more complicated, and conceptual graphs can be used for this, where each relationship is represented by a separate node.

Semantic networks can also be classified by size: industry networks, which serve as the basis for specific artificial intelligence systems, and global semantic networks, which seek to capture all the interconnections in the world, which may become possible in the future with the development of technology.

Complex Networks

Semantic networks, on the one hand, reflect the semantic features of their elements, and, on the other hand, fit perfectly into the modern direction of the mathematical study of network structures, which is called "Complex Networks". In this direction, various aspects of networks are

explored, not only in terms of their structure but also statistical phenomena, flow effects, and influences in network interactions.

A feature of this theory is its application to various types of networks, including electrical, transport, information and others. This vast field of research provides an in-depth understanding of the functioning of various systems, as well as optimizing their performance for practical purposes.

At the same time, language networks that preceded the advent of Large Language Models also made a significant contribution to the development of the theory of complex networks. The study of language networks has provided valuable knowledge and analysis methods that have become an important starting point for the creation and improvement of large language models.

As a result, the concept of Complex Networks is a broad field of research that brings together various disciplines and opens up new horizons for understanding and optimizing complex network structures in various fields of human activity.

The theory of complex networks has three main areas:

1. The study of statistical properties that characterize the behavior of networks. In this direction, various statistical indicators and characteristics of networks are studied, such as the distribution of degrees of nodes, the clustering coefficient, the average length of paths, etc. This allows us to understand the features and patterns that determine the behavior of complex networks in various contexts.
2. Creation of network models. In this direction, various mathematical and statistical models are being developed that help to abstract from specific networks and consider general structural properties. Models make it possible to understand how networks can develop and change, their vulnerability and their resilience to various influences.

3. Predicting the behavior of networks when changing structural properties. This direction is focused on studying the impact of various changes in networks on their behavior. These can be changes in the network structure, adding or removing nodes and links, as well as changing network parameters. Predicting such changes, it is possible to draw conclusions about the impact on the efficiency, stability and other characteristics of the network.

Networks settings

In applied research, network analysis is used to study various systems where objects and their interactions are represented as graphs or networks. In this context, several typical characteristics are most often used to help understand the structure and properties of the network.

1. Network size: This is a simple and important characteristic that reflects the number of nodes (vertices) in the network. The size of a network determines the total number of objects or elements that are connected.
2. Network Density: This characteristic indicates how tightly connected the network is. Network density measures the ratio of the number of actual links to the maximum possible number of links in a network. High density may indicate the presence of close connections between objects, while low density may indicate more sparse interactions.
3. Centrality measure: The centrality of a node reflects its importance or influence in the network. There are various types of measures of centrality, such as degree centrality (the number of connections a node has), closeness (the proximity of a node to other nodes), and mediation (the role of a node in passing information between other

nodes). Centrality measures allow you to identify key nodes that play an important role in the functioning of the network.

In addition to these characteristics, the analysis of complex networks, similar to graph theory, also examines the parameters of individual nodes in order to understand their roles and interactions, the parameters of the network as a whole for a general idea of the structure and dynamics of the network, as well as network substructures such as communities or clusters, to detect groupings of objects with similar characteristics or functional relationships.

For individual nodes in the network, the following parameters are distinguished, which help to understand their role and influence on the overall structure and functioning of the graph:

1. Node input half-degree: This parameter shows the number of edges that are directed into the given node. It reflects the importance of the node as a receiver or recipient of links.
2. Node's output half-degree: This parameter shows the number of edges that are directed out of the given node. It reflects the importance of the node as a source or sender of links.
3. Average distance from this node to others: This is the average number of edges that need to be traversed from this node to every other node in the network. This parameter characterizes how quickly and efficiently a node can exchange information with other nodes.
4. Eccentricity: This parameter defines the maximum value of geodesic distances from this node to all other nodes in the network. It allows you to identify how far from other nodes this node is and how much it can be the center of information dissemination.
5. Intermediation (betweenness): This parameter indicates how many shortest paths pass through this node. A node with a high mediation

value plays an important role in transferring information between other nodes and can be a key mediator in the network.

6. Centrality parameters: These are various measures that determine the importance of a node in a network in terms of its links to other nodes. For example, the degree centrality of a node reflects the number of connections of a given node with other nodes, and mediation centrality reflects its role in passing the shortest paths between other nodes.

The following parameters are also used to analyze the network as a whole:

1. Node Count: This is a simple metric that represents the total number of nodes (vertices) in the network.
2. Number of Edges: This is the number of links (edges) between nodes in the network.
3. Average distance between nodes: This is the average of the geodesic distances (shortest paths) between all pairs of nodes in the network. This allows us to understand the average distance between nodes.
4. Network Density: This is the ratio of the number of edges in the network to the maximum number of edges possible given the number of nodes n $(n - 1) / 2$. Density reflects how tightly connected the network is.
5. Number of symmetrical, transitive, and cyclic triples: These parameters reflect the characteristics of triples of nodes in the network, which can be symmetrical (all connections between nodes are forward and backward), transitive (when nodes A and C are connected through node B), or form closed cycles.

6. Network Diameter: This is the largest geodesic distance between all pairs of nodes in the network. The diameter reflects the maximum distance between nodes in the network.

An important characteristic of the network is the node degree distribution function $P(k)$, which is defined as the probability that an arbitrary node i online has a degree $k_i = k$. Networks that are characterized by different degree distributions $P(k)$, exhibit different behavior, in some cases may be Poisson distributions $(P(k) = e^{-m} m^k / k!)$, with expectation m , exponential $(P(k) = e^{-k/m})$, or power $(P(k) \sim 1/k^\gamma, k \neq 0, \gamma > 0)$.

Networks with the distribution of degrees of nodes in the form of a power law are called scale-free. It is scale-free distributions that are often observed in real social networks. With a power-law distribution, the existence of nodes with a very high degree is possible, which is practically not observed in networks with a Poisson distribution.

In other words, scale-free networks are characterized by the fact that some nodes can have a huge number of connections, while most nodes have much fewer connections. This phenomenon is called "the rich get richer" (preferential attachment), which means that the probability of adding links to a node depends on the number of links already available. Such scaleless networks are more resistant to accidental outages or attacks, which makes them interesting for research and applications in various fields such as sociology, biology, and linguistics.

The distance between nodes is defined as the number of edges through which one can get from one node to another. The least way d_{ij} between

nodes i and j called the smallest distance between them. For the entire network, you can introduce the concept of the average path, as the average value over all pairs of nodes of the smallest distances between them:

$$l = \frac{2}{n(n+1)} \sum_{i \geq j} d_{ij},$$

where n – number of nodes, d_{ij} – the shortest distance between nodes i and j .

Hungarian mathematicians P. Erdős and A. Rényi showed that the average distance between two vertices in a random graph (Erdős-Rényi model) grows as a logarithm of the number of its nodes.

Global Efficiency Ratio

The network may turn out to be disconnected, that is, there will be nodes, the distance between which will be infinite. Therefore, the middle path, according to the above formula, will also be infinite. To account for such cases, the concept of the average return path between nodes (it is also called "global network efficiency") is introduced, which is calculated by the formula:

$$il = \frac{2}{n(n-1)} \sum_{i > j} \frac{1}{d_{ij}}.$$

The reciprocal of the global efficiency is the harmonic mean of geodesic distances: $h = 1 / il$.

One way to find critical network components is to look for the most vulnerable nodes. The vulnerability of a network relative to a node can be defined as a decrease in the global efficiency of the network when a node and all edges adjacent to it are removed from the network:

$$IN_i = (il - il_i) / il,$$

where il is the global efficiency of the original network, and il_i is the global efficiency after removing node i and all edges adjacent to it.

The ordered distribution of nodes by this value is related to the structure of the entire network. Thus, the node that has the greatest impact on the vulnerability of the network occupies the highest position in the network hierarchy. The measure of network vulnerability is the maximum vulnerability among all nodes:

$$IN = \max IN_i.$$

Clustering coefficient

In 1998, D. Watts and S. Strogatz [Watts D., 1998] introduced the concept of clustering coefficient, which has become an important network parameter for analyzing connectivity in network nodes and forming groups of interconnected nodes known as cliques.

The clustering coefficient for a particular node shows how many nearest neighbors of a given node are also connected to each other, forming a kind of micro-community within the network.

The clustering coefficient can be calculated both for each individual node and for the entire network as a whole. In network analysis, the clustering coefficient for the entire network is defined as the normalized sum of the corresponding coefficients for all nodes. This allows you to evaluate the overall degree of connectivity and grouping in the network.

The study of the clustering coefficient makes it possible to better understand the structural organization of the network and identify areas with high connectivity, which is important for understanding its behavior and functioning. This characteristic is used in various fields, including linguistics, where the study of the degree of interaction and grouping of elements is important for understanding the system as a whole.

The clustering coefficient for an individual node in the network is determined as follows. Assume that a node has edges that connect it to other nodes that are its nearest neighbors. If we assume that all nearest neighbors are connected directly to each other, then the number of edges between them would be $1/2 \cdot k(k-1)$ – this number corresponds to the maximum possible number of edges that could connect the nearest neighbors of the selected node. The ratio of the actual number of edges connecting the nearest neighbors of this node i to the maximum possible number (the one at which all the nearest neighbors of this node would be connected directly to each other) is called the clustering coefficient of the node – $C(i)$.

Betweenness centrality

Betweenness is a node parameter that indicates how many shortest paths pass through it. This characteristic reflects the role of this node in establishing links in the network. The nodes with the most mediation play an important role in establishing links between other nodes in the network.

Betweenness b_m node m is determined by the formula:

$$b_m = \sum_{i \neq j} \frac{B(i, m, j)}{B(i, j)},$$

where $B(i, j)$ is the total number of shortest paths between nodes i and j , $B(i, m, j)$ is the number of shortest paths between nodes i and j that pass through the node m .

Modularity

Modularity is one of the network parameters that was introduced to measure the degree of network partitioning into modules (clusters, cliques). It is calculated as the difference between the fraction of edges within a

cluster in the network under consideration and the expected fraction of edges within a cluster in a network in which the vertices have the same degree as in the original one, but the edges are randomly distributed.

To calculate modularity, the concept of an adjacency matrix is used. Adjacency matrix A consists of elements a_{vw} , whose value is 0 if node v is not connected to node w , and the weight of the connection between v and w if these nodes are connected.

The modularity of the network can be expressed by the formula:

$$Q = \frac{1}{2m} \sum_{v,w} \left[a_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w),$$

where a_{vw} – adjacency matrix element A , m is the number of edges in the graph, k_v , k_w are the degrees of nodes v and w , respectively, δ is the Kronecker delta function (shows whether nodes v and w are in the same module).

So, modularity is a measure of the quality of clustering, on the basis of which a wide class of algorithms for detecting groups in networks is built.

Networks phenomena

One can speak of a "community structure" when there are groups of nodes (clusters) that have a high density of edges between themselves, while the density of edges between individual groups is low. The traditional method for revealing the structure of a community is cluster analysis. There are countless ways of cluster analysis, which are based on different measurements of distances between nodes. For large linguistic networks, the existence of a community structure turned out to be an essential property.

Meanings of "weak ties"

Among the characteristics of real social networks there are so-called "weak" links. An analog of weak social ties can be, for example, relationships with distant acquaintances and colleagues. In some cases, these ties are more effective than "strong" ties. A conceptual conclusion was obtained in the study of mobile communications, which is that "weak" social ties between individuals are most important for the existence of a social network.

It was found that it is weak social ties that unite a large social network into a single whole. If these connections are ignored, then the network will be divided into separate fragments, that is, the connectivity of the network will be broken. It turned out that it is weak ties that are the phenomenon that unites the network into a single whole.

Apparently, this effect also takes place in linguistic networks.

Small World

Despite the enormous size of some social networks, many of them have a relatively short path between any two nodes - the geodesic distance. In 1967, psychologist S. Milgram, as a result of large-scale experiments, calculated that there is a chain of acquaintances, on average, consisting of six links, between almost any US citizen.

D. Watts and S. Strogatz discovered a phenomenon that is characteristic of many real networks, called the effect of small worlds (Small Worlds). While studying this phenomenon, they proposed a procedure for constructing a visual model of a network that has this phenomenon. The three states of this network are shown in Fig. 4: a regular network - each node of which is connected to four neighbors, the same network, in which some "near" connections are randomly replaced by "far" ones (in this case,

the phenomenon of "small worlds" occurs) and a random network in which the number such replacements exceeded a certain threshold.

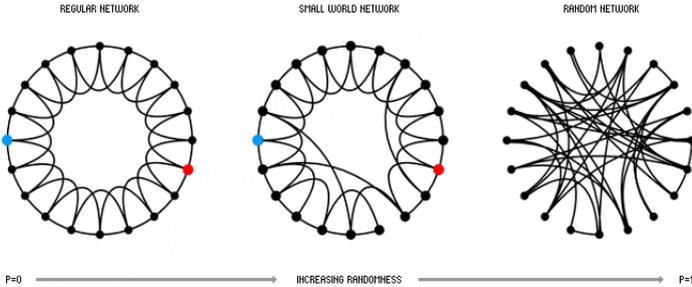


Figure 4. Watts-Strong model

On Fig. 5 shows graphs of changes in the average length of the path and the coefficient of clustering of the artificial network by D. Watts and S. Strogatz on the probability of establishing "long-distance connections" (on a semi-logarithmic scale).

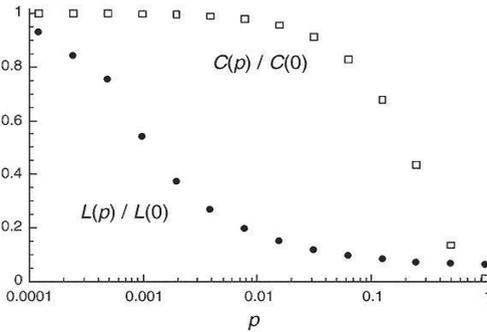


Figure 5. Dynamics of changes in the length of the shortest path and the clustering coefficient in the Watts-Strogatz model

In fact, it was found that it is precisely those networks whose nodes have several local and random "distant" connections at the same time that demonstrate both the small world effect and a high level of clustering.

The Rich Club Phenomenon

WWW (World Wide Web) is a network for which the phenomenon of small worlds is also confirmed. Web topology analysis by S. Zhou and R.J. Mondragon [Zhou, 2004] from the University of London, showed that nodes with a large number of outgoing hyperlinks have more links between themselves than with nodes with a small number of links, while the latter has more links with nodes with a large number of links than between yourself. This phenomenon has been called the Rich-Club Phenomenon. The study showed that 27% of all connections occur between only 5% of the largest nodes, 60% are connections between the remaining 95% of nodes and this 5% of large ones, and only 13% of connections form nodes that are not included in this main 5%.

These studies suggest that the WWW's reliance on large hosts is much greater than previously thought, making it even more susceptible to malicious attacks. Related to the concept of "small worlds" is also a practical approach called "network mobilization", which is implemented over the structure of "small worlds". In particular, the speed of information dissemination due to the effect of "small worlds" in real networks increases by orders of magnitude compared to random networks, because most pairs of nodes in real networks are connected by short paths.

Erdős-Rényi random network model

There are two models of a classical random graph: in the first model, it is assumed that M edges are distributed randomly and independently between pairs of N graph vertices; in the second model, the probability m is fixed, with which each of the pairs of vertices can be connected. At $M \rightarrow \infty$ and $N \rightarrow \infty$ for both options, the distribution of node degrees k is determined by the Poisson formula:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!},$$

where the average value of the knot degree: $\langle k \rangle = 2M/N$ for the first model and $\langle k \rangle = mN$ for the second one (m – probability of connection of nodes). In this case, the average length of the shortest path for the Erdős-Rényi network is $L = \ln(N)/\ln(\langle k \rangle)$, and the clustering coefficient is $C \sim \langle k \rangle / N$.

Barabashi-Albert Random Network Model

The Barabashi-Albert network construction scenario is based on two mechanisms – growth and preferential attachment. The model uses an algorithm: the network grows starting from a small number of nodes n_0 , to which a new node is added at each time step with $n < n_0$ links joining existing nodes; preferred joining is that the probability of joining $P(k_i)$ of a new node to an already existing node i depends on the degree k_i node i :

$$P(k_i) = \frac{k_i}{\sum_j k_j}.$$

In the denominator, the summation is carried out over all nodes. Both computer models and analytical solutions give a stepwise asymptotic distribution of node steps with an exponent γ close in value to the number 3.

One of the important areas of network analysis is their visualization potential, which plays a significant role since it often allows one to draw important conclusions about the nature of the interaction between network nodes without resorting to precise methods of data analysis. Visualization is a powerful tool for representing the complex structure of networks and can help researchers better understand their features and properties.

When visualizing a network model, it is often necessary to represent nodes in two dimensions in order to see the connections and interactions between them more clearly. This allows you to identify groups of subjects that are strongly connected to each other, as well as to identify features of the network structure, such as central nodes or individual groups.

Additionally, you can apply the spatial ordering of objects in one dimension, based on certain quantitative characteristics. For example, nodes can be ordered by their importance or by the number of links, allowing you to better visualize network dynamics and understand which nodes have the most impact on the entire network.

In addition, when visualizing networks, methods common to all network diagrams are used to display the quantitative and qualitative characteristics of objects and their relationships. For example, different shapes and colors of nodes and links can represent different characteristics of the network, such as the types of nodes or the strength of links between them.

Thus, visualization is a tool for studying and analyzing network structures, allowing researchers to discover interesting patterns and relationships between nodes, which contributes to a deeper understanding of their functioning and dynamics.

Host ranking

The two most well-known node ranking algorithms, HITS (Hyperlink Induced Topic Search) [Kleinberg, 1999] and PageRank [Brin, 1988], were developed by IBM J. M. Kleinberg and Stanford University – S. Brin and L. E. Page.

Algorithm HITS

The HITS algorithm selects from the network the best "authors" (the nodes to which the link leads) and "intermediaries" (the nodes from which the citation links leave).

A node is a good intermediary if it contains links to valuable sources, and conversely, a node is a good author if it is mentioned by good intermediaries.

For each node $d_j \in D$ recursively calculates its importance as an author $a(d_j)$ and mediators $h(d_j)$ according to the formulas:

$$a(d_j) = \sum_{i \rightarrow j}^{|D|} h(d_i), \quad h(d_j) = \sum_{j \rightarrow i}^{|D|} a(d_i).$$

In this case, in the first formula, only the mediation values of those nodes that refer to the node with the index j .

Accordingly, in the second formula, the authorship value of only those nodes referred to by the node with the index j . At each step of the algorithm, the values $a(d_j)$ and $h(d_j)$ normalized

Algorithm PageRank

The PageRank algorithm is applied to any node, taking into account the number of links from other nodes to this node. At the same time, PageRank, like HITS, unlike the literary citation index, does not consider all links to be equivalent.

The principle of calculating the rank of PageRank nodes is based on the user's "random walk" model according to the following algorithm: he opens

a random node (web page), from which he follows a randomly selected link. It then navigates to another web page and activates a random link again, and so on, constantly jumping from page to page, never coming back. Sometimes, when with some probability $1 - \delta$ he gets tired of such wandering, or there are no links to other pages on the page, he again goes to a random web page - not by link, but by manually typing some URL. It is assumed that the probability that a user roaming the web will go to some particular web page is its rank. Obviously, the PageRank of a node is higher, the more other nodes link to it, and the more popular these nodes are.

Let it exist n nodes $D = \{d_1, d_2, \dots, d_n\}$ that link to this node (web page A), a $C(A)$ – total number of links from the node A to other nodes. Some fixed value is defined δ as the probability that the user, viewing any web page from the set D , will go to the node A via a link rather than by explicitly typing its URL.

Within the model, the probability of this user continuing to surf the web from web pages without using links by manually entering an address (URL) from a random page is $1 - \delta$ (an alternative to following links). The PageRank index for a site is considered as the probability that the user will be at some random time on this site:

$$PR(A) = (1 - \delta) / N + \delta \sum_{i=1}^n \frac{PR(d_i)}{C(d_i)}.$$

According to this formula, the node index is calculated by a simple iterative algorithm.

Despite the differences between HITS and PageRank, these algorithms have in common that the authority (weight) of a node depends on the weight

of other nodes, and the level of the "intermediary" depends on how authoritative the nodes to which it refers are.

Language Networks

The first step in applying the theory of Complex Networks [Strogatz, 2001], [Albert, 2002] to text documents is the creation of a network model of these documents in the form of a set of nodes and links, that is, the construction of language networks [Golovach, 2006], in which the most significant nodes are identified, which are sometimes called resisting words or corresponding phrases.

Along with the sequential analysis of text documents, the construction of networks, the nodes of which are such elements as words or phrases, that is, fragments of a natural language, makes it possible to detect the structural elements of texts, without which texts lose their coherence. There are several approaches to building networks from texts and different ways of interpreting nodes and links, which leads, respectively, to different types of representation of such networks. Nodes can be connected to each other if the words corresponding to them are next to each other in the text [Ferrer-i-Cancho, 2001], [Dorogovtsev, 2001], belong to the same sentence or paragraph [Caldeira, 2005], connected syntactically [Ferrer-i-Cancho, 2004], [Ferrer-i-Cancho, 2005] or semantically [Motter, 2002], [Sigman, 2002].

The preservation of syntactic links between words leads to the representation of the text in the form of a Directed Network, where the direction of the link corresponds to the subordination of the word.

If we associate each word with a network node and connect every two nodes with a link when the corresponding words are next to each other in a sentence, then such a representation is called an L -space. In L -space, as well

as in the other network models given below, when multiple connections arise, it is customary to keep only one of them.

Traditionally, there are four types of language networks (spaces) [Golovach, 2007]:

1. *L*-space. Adjacent words related to the same sentence are linked. The number of neighbors for each word (word window) is determined by the interaction radius R , most often the case $R = 1$ is considered.

2. *B*-space. Two types of nodes are considered, corresponding to the sentence and the words belonging to them.

3. *P*-space. All words belonging to the same sentence are linked together.

4. *C*-space. Sentences are linked together if the same words are used in them.

In the case of an *L*-space, connections can take into account not only "nearest neighbors", but also groups over several words that are at a certain distance from each other. For this, the concept of "radius of action" is introduced R : at $R = 1$ connection exists only between nearest neighbors, when $R = 2$ – between nearest and subsequent close neighbors, etc. The variable R can take on a value from $R = 1$ to R_{\max} , where $R_{\max} + 1$ is the total number of words in the sentence. The growth of the "radius of interaction" R in this case leads to an increase in the number of bonds, reaching saturation at $R = R_{\max}$.

Another way to represent text as a network is to use bipartite graphs. In this representation (*B*-space), nodes of two types are considered. One type responds to sentences, the other to words. The connection between different nodes means that the word belongs to the sentence.

In the P -space, all words belonging to one sentence are considered to be related to each other.

In C -space, nodes correspond to a sentence, and a link between sentence nodes is established if there are common words in the corresponding sentences.

For a network built on the basis of the British National Corpus (L -space of the language, $R = 1$) it turned out that this network of the English language is scaleless, and the behavior of the degree $P(k)$ is characterized by two regimes of power-law distribution with the values of the corresponding exponents $\gamma = 1.5$ for $k < 2000$ and $\gamma = 2.7$ for $k > 2000$.

According to the definition, if the average length of the shortest path grows with the size (number of nodes) of the network more slowly than any power function, then the network is a "small world". Small world networks are very compact. For the above English language network, the length of the shortest path is only $\langle l \rangle = 2,63$. Since the growth of R only leads to the addition of new bonds, then $\langle l \rangle$ decrease as R increases.

A specific form of correlation in networks is the formation of clusters. The clustering coefficient C characterizes the propensity of the network to form connected triplets of nodes. It is known that for a complete graph $C = 1$, and for a network in the form of a tree $C = 0$.

The ratio of the average clustering coefficient of the studied networks to the clustering coefficient of the classical random graph indicates that the networks of the language are well-correlated structures. Such correlations grow with the growth of the "radius of interaction" R .

For the British National Corpus, based on the analysis of texts containing $\approx 10^7$ words, the value of the clustering coefficient is obtained $\langle C \rangle = 0,687$.

In the case of considering a P-space, each word-knot is associated with all other words belonging to the general sentence. Thus, each sentence of text enters the network as a complete graph - a click of interconnected nodes. Various click-sentences are networked together thanks to common words. In *L*-space, words are linked inside a window, the size of which is characterized by the value R . When the window size R becomes equal to the size of the sentence, then the representations of this sentence in *L*- and *P*-spaces coincide. Accordingly, when the window size becomes equal to the size of the largest sentence of text ($R = R_{\max}$), then the representations of the entire text in *L*- and *P*-spaces coincide.

In practice, it is confirmed that the language network is a highly correlated scale-free small world (Scale-Free Small World). There are a number of works that have attempted to explain the properties of language networks using the Preferential Attachment scenario [Albert, 1999], viewing them as the result of a growth process where new word nodes are more likely to attach to existing hub nodes. many connections.

Concept networks, which will be discussed later in this book in section 4, can also be considered as language networks, in which nodes are verbalized concepts, and relationships are defined semantically.

Semantic Web

Since the 1980s, due to the development of web technologies, semantic networks and the concept of the Semantic Web have become one of the key areas for organizing and presenting information in an online environment more efficiently. The Semantic Web aims to provide more structured and semantically rich information that will enable computers and programs to understand the content and context of Web resources more deeply.

The Semantic Web is a concept proposed by Tim Berners-Lee, inventor of the World Wide Web and director of the World Wide Web Consortium (W3C) (Fig. 6). He introduced the idea that web pages should not only be readable by humans, but also be understood and processed by computers. The goal of the Semantic Web is to create structured information that can be processed by computers to automate information retrieval, integrate data from different sources, and create smarter applications.

In Berners-Lee's dream, the Semantic Web was supposed to solve complex problems based on the knowledge embedded in the information add-on to the regular web and a system of intelligent agents that process knowledge distributed over the network. He wrote: "The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. Such an agent coming to the clinic's Web page will know not just that the page has keywords such as "treatment, medicine, physical, therapy" (as might be encoded today) but also that Dr. Hartman works at this clinic on Mondays, Wednesdays, and Fridays and that the script takes a date range in yyyy-mm-dd format and returns appointment times. And it will "know" all this without needing artificial intelligence on the scale of 2001's Hal or Star Wars' C-3PO. Instead, these semantics were encoded into the Web page when the clinic's office manager (who never took Comp Sci 101) massaged it into shape using off-the-shelf software for writing Semantic Web pages along with resources listed on the Physical Therapy Association's site" [Berners-Lee, 2001].

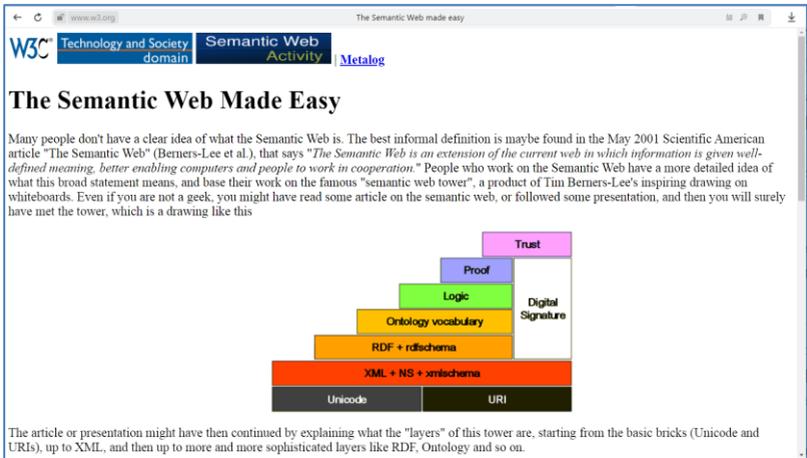


Figure 6. Fragment of the web page of the W3C site

In today's Internet, when the concept of a semantic web is applied, the web of hyperlinked human-readable web pages is expanded by adding machine-readable metadata about the pages and the links between them. This allows automated agents to interact more intelligently with the Web and perform more complex tasks on behalf of users. Tim Berners-Lee, who is responsible for developing the Semantic Web standards, defines the Semantic Web as "a web of data that machines can process directly and indirectly".

Many of the technologies proposed by the W3C already existed before they were included in the W3C standards. Their application is varied, especially in areas where information covers a limited and defined area, and data exchange is a necessity, such as in scientific research or data exchange between companies. In addition, other technologies with similar goals have emerged, such as microformats.

One of the important achievements of Tim Berners-Lee was the creation of the RDF (Resource Description Framework) standard, which is used to describe resources and their properties on the semantic web.

The term "Semantic Web" is often used more specifically to refer to the formats and technologies that make it possible. Collection, structuring and extraction of related data are provided by technologies that provide a formal description of concepts, terms and relationships in a particular field of knowledge. These technologies are defined in the W3C standards and include:

- Resource Description Framework (RDF), a general method for describing information
- RDF Schema (RDFS)
- Simple Knowledge Organization System (SKOS)
- SPARQL, an RDF query language
- Notation3 (N3), designed with human readability in mind
- N-Triples, a format for storing and transmitting data
- Turtle (Terse RDF Triple Language)
- Web Ontology Language (OWL), a family of knowledge representation languages
- Rule Interchange Format (RIF), a framework of web rule language dialects supporting rule interchange on the Web
- JavaScript Object Notation for Linked Data (JSON-LD), a JSON-based method to describe data
- ActivityPub, a generic way for client and server to communicate with each other.

The Semantic Web stack is the structure or architecture of the Semantic Web. The functions and relationships of its components can be briefly described as follows:

1. XML (Extensible Markup Language) - Provides a syntax for structuring content in documents, but does not provide semantics for content value.
2. XML Schema is a language for defining the structure and content of elements in XML documents.
3. RDF (Resource Description Framework) is a simple language for expressing data models and relationships between objects.
4. RDF Schema (RDFS) - extends RDF and provides a vocabulary for describing RDF-based properties and resource classes.
5. OWL (Web Ontology Language) - adds extended vocabulary terms to describe properties and classes with more complex relationships and characteristics.
6. SPARQL (SPARQL Protocol and RDF Query Language) is a protocol and query language for accessing Semantic Web data.
7. RIF (Rule Interchange Format) is an XML-based language for expressing web rules that can be executed by computers.

These components together form the Semantic Web stack, which allows you to organize and process data semantically, enriching the information and making it more understandable and useful for machine processing.

There are currently no widely available tools for viewing and directly using the information provided on the Semantic Web. Despite rare samples and scattered attempts, client programs do not exceed the level of local research projects of individual enthusiasts. This is a challenge for the active development of the Semantic Web.

Commentators point to various reasons that impede the progress of the Semantic Web, ranging from the human factor (unwillingness of people to maintain documents with metadata, problems of metadata reliability, and others), and ending with the problem of dividing the world into separate

concepts, as Aristotle pointed out. An important factor is the question of the realizability of the top-level ontology, which is critical for the successful development of the Semantic Web. All these factors raised doubts about the possibility of the full functioning of the Semantic Web until now.

In response to these criticisms, we can say that the efforts spent on the development of the Semantic Web were not in vain. During this time, standards, formats and programs have been created that serve as the foundation for future development. Especially now, when artificial intelligence technologies are experiencing a revolutionary rise, in particular in connection with the development of Large Linguistic Models, we look forward to the practical implementation of the Semantic Web concept on a global scale. This is especially true in the tasks of search, analytics and decision support. In the future, the Semantic Web could become a powerful tool for intelligently interacting with information, enriching data, and improving information processes on a global scale.

3. Networks analysis and visualization tools

Social Network Visualizer



<http://socnetv.org>

One of the best desktop user applications for visualizing and analyzing network structures is currently the Social Networks Visualizer (SocNetV). It is a free cross-platform social media analysis and visualization software application developed in C++. SocNetV is free software licensed under the GNU General Public License 3 (GPL3).

SocNetV source code, packages and executables for Windows, Linux and MacOS are available on the project website.

SocNetV has the ability to manually enter a network (graph) or load an existing network dataset presented in GraphML, UCINET, Pajek, etc. formats, calculate standard graph and network connectivity indicators, such as density, diameter, geodetic and distances, connectivity, eccentricity, clustering coefficient, reciprocity, etc., centrality values, the use of various layout algorithms based on centrality or mediation (Betweenness) of nodes or dynamic models.

Basic features *SocNetV*:

- import from network formats (*GraphML*, *Adjacency*, *A spider*, *UCINET*, lists, etc.);
- the ability to export to GraphML, Pajek, Adjacency formats;

- loading and editing the network;
- quick calculation of indices of centrality, connectivity, density, geodesic distance, degrees of nodes, eccentricity, clustering coefficient, etc.;
- calculation of additional metrics for social network analysis, such as centrality and importance indices;
- calculation *PageRank*;
- matrix calculations: definition of adjacency graph, Laplace matrix, citation, etc.;
- analysis of structural equivalence using hierarchical clustering, the similarity of actors and differences in relationship profiles, Pearson coefficients;
- availability of fast community detection algorithms, such as triad census, clique census, etc.;
- calculation of various centrality indices (eigenvector and proximity centrality, betweenness centrality, information centrality, power centrality, closeness and page prestige).
- the ability to load and edit a multi-relational network. You can upload a social network that consists of multiple relationships, or you can create your own network and add multiple relationships to it;
- availability of datasets for the analysis of social networks;

- the presence of different layout models based either on visibility indices (i.e. circular, level and nodal sizes in terms of centrality) or on force placement (i.e. Kamada-Kawai, Fruchterman-Reingold, etc.) for significant visualization of social networks;
- the creation of random networks using various random network generation models (Barabási-Albert Scale-Free, Erdős-Rényi, Watts-Strogatz Small-World, d-regular, ring lattice, etc.);
- the presence of matrix subroutines for calculation: adjacency graph, Laplace matrix, degree matrix, etc.
- the presence of a built-in web crawler to automatically create "social networks" from links found in a given source URL.
- the availability of complete documentation, available both online and within the application, which explains in detail each function and algorithm of SocNetV.

The Social Networks Visualizer (SocNetV) program can be a useful tool for analyzing and visualizing networks of concepts obtained using

SocNetV can help visualize and analyze these relationships between individual concepts, making it easier to understand information and highlight key elements in the network.

With the help of the SocNetV program, you can carry out:

- Graph visualization: SocNetV can display networks of concepts as a graph, where concepts are represented by nodes and links between them are edges. This makes it easy to see the structure of the network and the relationships between concepts;

- link analysis: The program allows you to analyze the degree of relationship between concepts, such as the frequency of links or the strength of the interaction between them. This will help identify key concepts and their impact on the entire network;
- filtering and coloring of nodes: SocNetV allows you to filter nodes by various parameters, such as frequency of mention, in order to focus on the most significant concepts. It is also possible to color nodes by category, which will help to highlight different groups of concepts;
- data export: The program allows you to export visualized networks in various formats, such as images or data files, which will simplify further use and processing of information.

Gephi

General information about Gephi



<https://gephi.org>

Gephi (<https://gephi.org/>) is currently the most popular visualization and analysis program for networks and graphs ("network graphs"). Gephi provides a fast layout, efficient filtering, and interactive data exploration, and is one of the best options for visualizing large-scale networks. Gephi is a multiplatform open source software distributed under CDDL 1.0 and GNU General Public License v3. Mac OS X, Windows, and Linux versions of the source codes are available at <https://gephi.org/>. The application requires Java 1.7+.

Gephi developers describe this application as "like Photoshop, but for data".

Gephi allows you to download network data in GEXF, GDF, GML, GraphML, Pajek (NET), GraphViz (DOT), CSV, UCINET (DL), Tulip (TPL), Netdraw (VNA), and Excel spreadsheet formats. In addition, Gephi allows you to export network data in JSON, CSV, Pajek (NET), GUESS (GDF), Gephi (GEXF), GML and GraphML formats. This allows Gephi to interact with other graph analysis and rendering systems.

The Gephi program provides many different methods for laying out graphs (arrangement of nodes and links on the plane) and allows the user to customize the colors, sizes, and labels in the graphs. It is an interactive software tool that provides tools for discovering communities in networks, as well as calculating shortest paths or relative distances from one node to another.

Gephi supports plugins that allow you to extend its functionality and add new algorithms, layouts, and measurement tools. Thanks to the multi-threaded data processing scheme, Gephi allows you to perform several types of analysis at the same time, which increases the efficiency of working with large and complex graphs.

The user interface of the Gephi system includes three main sections (windows):

- Data Laboratory: All initial network data is stored here, as well as additional calculated values.
- Overview: This is where most user operations take place, including manually editing networks, testing layouts, and setting filters.
- "Preview": here the graph output form is specified, usually with the help of a set of graph tools, the graph is finalized, including from an

aesthetic point of view. In the same window, a call to export the graph in PDF, PNG and SVG formats is implemented.

These three main sections cover many tabs that allow the user to implement individual functions. Each of the main and secondary windows - sections and tabs - is discussed below.

Data Laboratory

The Data Laboratory at Gephi includes:

- All initial data about networks that were imported into the program.
- Additional calculated values and metadata associated with network data.
- Ability to view, edit and manage nodes and links in the network.
- Tools for filtering and processing data, such as deleting nodes or links based on certain criteria.
- Ability to view and edit attributes of nodes and links, such as labels, colors, sizes, etc.
- Various functions for working with data, such as sorting, searching and grouping nodes and links.

Although Data Laboratory may look like a spreadsheet, its functionality should not be confused with Excel or Google Spreadsheets. Some data processing can be done here, but it's best to prepare the basic network data before importing it into Gephi. To create various arrays of large volumes, it is better to use spreadsheet tools. Likewise, field values based on a particular sorting scheme are best created outside of Gephi.

However, this does not mean that the data stored in the laboratory is completely static. For example, all aggregation and clustering will automatically add new values for each node when the process starts. It is also possible to add columns to the table, copy data from one column to another, delete columns, etc.

It should be noted that making bulk changes at the node or edge level can be very time consuming, especially if the network dataset under investigation consists of thousands of values.

Overview section

All network data is first viewed in the Overview section, where Gephi provides an initial view of the network under study. The initial appearance of the network may be simple, but then special processing is performed on this representation. All functions related to network laying, filtering, segmentation, coloring and any other layout settings are visible first of all in this window.

The graphics window is adjacent to several toolbars, each containing a variety of functions. The functionality of each of these options is generally intuitive.

The Overview section in the Gephi system is intended to provide a general overview and visualization of the graph so that the user can get an idea of the network structure and the relationships between nodes and links. This section offers various opportunities to gain a deeper understanding of network data and to make decisions about how to proceed.

Main features of the Overview section:

- **Graph visualization:** The Overview section provides the ability to visualize the graph using various display methods, such as forced layouts (Force-Directed Layout), radial layouts (Radial Layout), etc. This allows you to see the structure of the network, determine the main groups of nodes and links, as well as identify features and patterns in the data.
- **Zooming and Navigation:** The user can zoom in and out of the graph and navigate through it to get a closer look at individual nodes or links or to get an overall picture of the network.

- Interactivity: In the Overview section, the user can interact with the graph, such as selecting nodes or links, applying highlighted filters, moving nodes for better visibility, etc.
- Attribute Display: The user can customize the display of nodes and links based on their attributes such as color, size, shape, etc. This helps highlight important properties and different groups of data.
- Statistical Overview: The Overview section can also provide a summary of statistics about the graph, such as the number of nodes and links, average attribute values, and more.
- Control Panel: Here the user can adjust various visualization options, as well as select different layout algorithms and layouts for the best representation of the graph.

The Overview section provides a general overview of network data, helps you visualize the graph to better understand its structure, and enables you to make informed decisions about further analysis and visualization of network data.

Preview section

The Preview section in the Gephi system is designed to refine the shape of the graph output, usually with a set of tools that allow you to improve the appearance and aesthetic characteristics of the visualization. In this section, the user can view and edit the graph to achieve an optimal representation of the network data.

The Preview window in Gephi allows the user to customize various attributes that were created in the original graphics window. Here you can set node labels, and select the font, its size, color, outlines, etc.

The appearance of a node is also specified by specifying the border width, border color, and transparency parameters. However, you can always

switch to the data processing window to make a lot of adjustments in Gephi, and then return to the preview window and update the graph display.

To customize the appearance of graph edges, options such as setting the edge thickness, color, transparency, the possibility of curved edges, and setting labels are provided. For directed edges, you can customize the edge arrows.

The main features of the Preview section in Gephi:

- Visualization settings: The user can configure various visualization parameters, such as the size and color of nodes and links, line width, shape, etc. This allows you to create more attractive and informative visualizations.
- Placement of Elements: The user can move the nodes and links of the graph manually to emphasize certain structures or show important relationships.
- Filtering and Zooming: The user can apply filters to the graph to hide irrelevant elements and focus on the most important nodes and links. It is also possible to zoom in or out for a better perception of graph details.
- Export: In the Preview section, the user can save the graph in various formats such as PNG, PDF, SVG and others to share the visualization with others or use in documentation and presentations.
- Advanced settings: The user can configure advanced visualization options such as displaying labels, including a legend, adding a background image, etc.
- Visualization experiments: The user can conduct various visualization experiments to find the most convenient and understandable way to present network data.

The Preview section allows the user to visually check and optimize the graph representation to make it more informative, attractive and easy to understand. This is an important step when working with network data, which allows you to get high-quality and professional visualization of graphs in the Gephi system.

Creating a new graph in Gephi

There are three main modes for creating a new graph in Gephi:

- via the Graph interface in the Overview mode;
- through Data Laboratory;
- by exporting graph data from an external file (the easiest way is from a file in CSV format with semicolon delimiters).

Creating a new graph in Overview mode

After starting the program and closing the screen that pops up when loading, the Overview interface is immediately activated, within which you can create a new graph. To do this, just activate a new project and use the tools indicated on the right side of the window.

To draw nodes manually using Gephi's own tools, use the "Knot Drawing Pencil" button.

By selecting a place on the screen With the Pencil for drawing nodes, you can draw new nodes and use the tool from the Size menu (the icon is the size of the graph) to change their size.

With the Edge Pencil tool, you can arrange the edges of a graph. After arranging the edges, you can switch to the coloring mode (the "paint" icon).

In this case, it is possible to either paint over all the neighbors of the selected node, or use individual coloring (upper right menu of the workspace). As a result, we get the final generated graph.

Creating a new graph in Data Laboratory mode

To create a graph, it is also convenient to switch to the Data Laboratory mode, where all information about the current state of the graph is displayed in tabular form. Moreover, the information is displayed in a form suitable for change (editing). For example, you can add new edges (nodes), and delete or modify existing ones. Data Laboratory is also convenient for applying text labels to nodes.

As a result, the Graph tab in Overview mode displays node and edge labels whose fonts, size, and brightness can be changed using the tools presented at the bottom of the screen.

Export graph data from an external file

Graph data can be loaded into Gephi from a text format in which node label elements are separated by a semicolon. In this case, to the node corresponding to the first label in the line, all other nodes whose labels are given in this line are "attached". For example, suppose the external file contains the following entries:

```
Node1;Node2;Node3;Node4;Node5
```

```
Node5;Node3
```

In this case, after loading them into the Gephi system and processing them (preparing for visualization in the way already described), we get a reflection (Fig. 7):

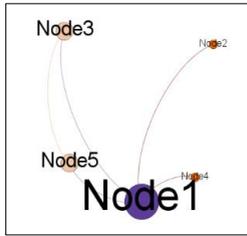


Figure 7. Displaying the graph after loading and processing

It should be noted that the main option for exporting graph data from an external file is to load the initial network data in CSV format, in which the elements are separated by a semicolon. In this case, the CSV file should actually contain the label-extended network incidence matrix. The following is an example for a network of five nodes:

```

;Node1;Node2;Node3;Node4;Node5
Node1;0;1;0;1;0
Node2;1;0;0;1;0
Node3;0;1;0;0;1
Node4;1;1;1;0;0
Node5;0;1;0;1;0

```

After loading into the Gephi system and processing in the way already described, we get a display (Fig. 8).

The above file can be prepared in Excel and then saved in CSV format. Note that all commas (",") must be replaced with semicolons (";") in the CSV file before uploading to Gephi - this is a feature of Gephi.

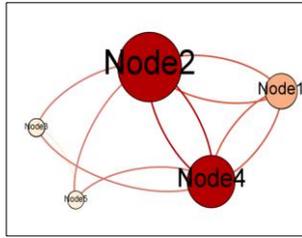


Figure 8. Displaying a graph loaded from a CSV file

Layout and ranking

When analyzing large and dense networks, fast layout (location of graph nodes) is a bottleneck, since most complex layout algorithms require significant CPU, memory, and runtime resources. At the same time, Gephi comes with efficient layout algorithms such as Yifan-Hu, Force-directed. In particular, the Yifan-Hu algorithm is ideal for applying after other, faster and coarser algorithms. While most of the methods proposed in Gephi can be executed in a reasonable amount of time, a combination of, for example, OpenOrd and Yifan-Hu provides the best visual representations. Of course, the correct parameterization of any layout algorithm can affect both the running time and the rendering result.

Gephi is essential for the analysis and visualization of semantic networks, including those obtained by the authors using large linguistic models. Here are a few key aspects that make Gephi a valuable tool for working with semantic networks:

- Flexible Data Import: Gephi supports importing data from various sources, including CSV, Excel, databases, and more. This allows users to easily upload their semantic networks created with large linguistic models for further analysis and visualization.

- Powerful Layout Algorithms: Gephi offers powerful graph layout algorithms such as Yifan-Hu and Force-directed, which allow you to automatically lay out the nodes of the semantic web in a way that reveals structures, patterns, and groupings.
- Visualization and interactivity: Gephi provides powerful visualization tools that allow the user to customize the appearance of nodes and links, apply filters, scale, and navigate the Semantic Web. Interactivity makes it possible to interact with the graph and explore it in detail.
- Community discovery and network feature analysis: Gephi provides functionality for community discovery and analysis of various network features such as centrality, degree of importance, and others, allowing semantic network authors to explore their structure and properties.
- Extensibility: Gephi supports plugins, which allows you to extend its functionality and add new algorithms and tools for working with semantic networks.

All this makes Gephi a powerful and convenient tool for analyzing and visualizing semantic networks obtained using large linguistic models. It allows researchers to further analyze and understand the complex relationships between concepts in such networks.

GraphViz

The modern level of service for visualization and analysis of network structures is provided by systems created by large teams of developers, for example, the system *GraphViz (Graph Visualization Software)* [Ferreira, 2017]. This system was developed by laboratory specialists AT&T, distributed under an open source license EPL (*Eclipse Public License*), and

runs on many operating systems, including Linux, Mac OS, Unix-like OS, and Microsoft Windows.



<http://graphviz.org>

GraphViz this is a set of utilities, libraries and programs with a graphical interface, presented in the form of a description in the DOT language, as well as additional text and graphics programs, widgets and libraries used in the development of software for visualizing structured data.

GraphViz includes the following tools:

- dot – A tool for creating a multilevel graph with the ability to display the image of the resulting graph in various formats (PNG, PDF, PostScript, SVG and others);
- neato – a tool for creating a graph based on a "spring" model ("spring model", "energy minimized");
- twopi – a tool for creating a graph based on a "radial" model;
- circo – a tool for creating a graph based on a "circular" model;
- fdp – a tool for creating an undirected graph based on the fdp model;
- dotty – a graphical interface for creating graphs;
- lefty is a programmable graphical widget (in EZ language).

The package of utilities includes the software module "dot" - an automatic visualizer of directed graphs, which takes as input a text file in the DOT language with a graph representation in the form of adjacent lists, and at the output forms a graph in the form of a graphic, vector or text file.

The "dot" program, an automatic visualizer of directed graphs, takes as input a text file with a graph representation in the form of adjacent lists, and at the output, it generates a graph in the form of a graphic, vector or text file.

DOT supports the following output file formats:

- PNG,
- GIF,
- JPEG,
- SVG(xml),
- DOT (txt),
- imap (HTML),
- VRML,
- PostScript and others.

To build a graph in the Graphviz system, it is enough to set its description in the special DOT description language in the Edit mode, and then select the Layout tab in the Graph mode to visualize the graph (Fig. 9).

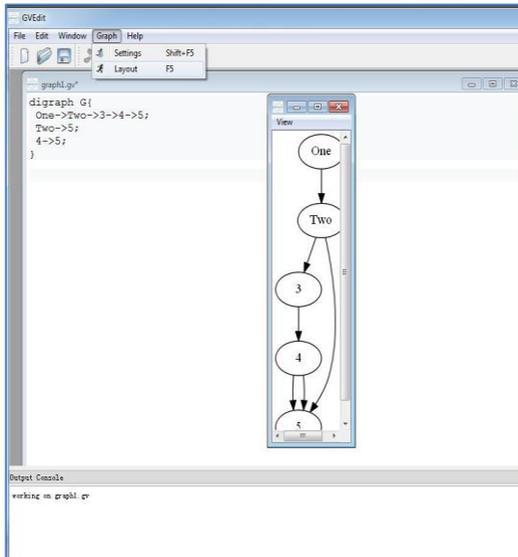


Figure 9. Description of the graph in the DOT language, visualization and diagnostic window

In graph mode Setting, you can change the graph parameters manually without the need for a direct description in the DOT language, for example,

change the shape of the nodes, the background color, the color of the nodes, etc.

The input file for the "dot" program is a plain DOT text file. The structure of a DOT file is very simple. The "dot" program itself recognizes all connections of the graph and arranges it in such a way that there is a minimum number of intersections.

A graph in the DOT language is described as a list of subgraphs, each of which has the form:

```
graph %graph_name% {  
}
```

In this case, curly braces {} contain comments and instructions describing a separate subgraph. The instructions describe the nodes and edges of the target graph and are separated by semicolons.

The DOT language supports C and C++ style comments - // and /**/ - as well as the # character as the first character of a single line comment.

Representation of undirected or directed graphs is allowed.

An undirected graph in DOT is described by a list of nodes and edges, represented by the name of the nodes and a double dash (--) between the connected nodes, for example:

```
graph graphname {  
  a;  
  b;  
  c;  
  d;  
  a -- b;  
  b -- c;  
  b -- d;  
}
```

Short description allowed:

```

graph graphname {
    a -- b -- c;
    b -- d;
}

```

A directed graph in the DOT language is described by a list of nodes and edges, represented by the name of the nodes and an arrow (\rightarrow) between connected nodes, for example:

```

digraph graphname {
    a -> b -> c;
    b -> d;
}

```

A redundant description is also allowed:

```

digraph graphname {
    a;
    b;
    c;
    d;
    a -> b;
    b -> c;
    b -> d;
}

```

When describing graphs in the DOT language, you can use attributes that determine the color, shape, and style of nodes and edges. Attributes are described in key=value pairs enclosed in square brackets ([key=value]).

For each element of the graph, several attributes can be defined, separated by a space.

```

graph graphname {
    // label - the visible name of the vertex
    a [label="Foo"];
    // shape - defining the shape of the vertex
    b [shape=box];
    // color - determination of the color of the edge
    a -- b -- c [color=blue];
    // style - defining the edge style
    b -- d [style=dotted];
}

```

DOT language interpreters position elements automatically when rendering. The "dot" program itself recognizes all the connections of the graph and arranges it in such a way that there is the least number of intersections. To correct the visual representation, graphic editors are used, among which is Graphviz presented in this paragraph.

Graphviz has several advantages over other free graph analysis and visualization tools when working with concept networks generated using ChatGPT or other language models:

- Ease of use: Graphviz provides an easy and intuitive way to visualize concept networks. Once the data is received using ChatGPT, it can be easily converted to a DOT format that Graphviz understands and then graphs can be rendered without complicated extra steps.
- Automatic graph stacking: Graphviz provides powerful algorithms for automatic graph stacking, which makes the process of visualizing concept networks more efficient and convenient. This is especially important when working with large and complex graphs.
- Variety of output formats: Graphviz supports a variety of output formats for visualizations, including PNG, PDF, SVG, and more. This makes it easy to save graphs in the desired format and use them in various contexts such as presentations, reports, or interactive web applications.
- Broad support and active community: Graphviz has a long history of development and an active community of users and developers. This provides access to updates, fixes, and new features, as well as support and assistance when needed.
- Integration with various programming languages: Graphviz provides API interfaces for working with various programming languages,

which allows you to integrate it into existing projects and use it to automate the process of analyzing and visualizing concept networks.

Thus, GraphViz is a powerful tool for visualizing concept networks obtained using ChatGPT or other language models, which provides API interfaces for working with various programming languages, in particular Perl, PHP, Java, and Python. Thanks to this feature, it can be integrated into many applications, making it a versatile tool for working with graphs.

Internal graph stacking algorithms in GraphViz allow you to automatically determine the location of nodes and links in a graph in such a way as to minimize intersections and ensure optimal readability. This makes it a very handy tool for creating beautiful and informative graphical representations of data.

GraphViz supports various output formats including SVG (Scalable Vector Graphics). The SVG format allows you to create interactive graphs with the ability to display them on the Internet. Such interactive graphs can be used to visualize complex data structures, as well as in interactive web applications or reports. The ease of use of GraphViz, the ability to automatically stack graphs, support for various formats, and integration with programming languages make it a convenient and efficient choice for analyzing and visualizing such networks.

The program described in the next paragraph uses GraphViz to create graphs and save them in SVG format to provide interactive networks and visualizations in a web application or internet resource. This allows users to interact with graphs, conduct searches in external search engines, and obtain additional information when interacting with graph elements.

CSV2Graph

When using the tools for analyzing and visualizing network structures given in the previous paragraphs, analysts face two problems, namely:

1. The need to install software products, which is not always possible, especially if it becomes necessary to work from mobile devices, new operating systems, or in conditions of restrictions on installing third-party software.

2. It is necessary to delve into the features of the functioning of these systems, and deal with dozens of parameters, modes of stacking graphs, clustering, etc.

Gephi-Lite



<https://gephi.org/gephi-lite>

WebGapViz



<http://www.webgraphviz.com>

If the first problem can be solved by online graph visualization analysis systems, among which, according to the authors, the Lite version of the Gephi system - Gephi-Lite and the web version of the Graphviz system - WebGraphViz can be attributed to the best ones, then to solve the second problem As part of the operational construction and display of domain models, it became necessary to develop our own service. Usually, special formats are used to describe graphs, among which one can distinguish, such as GML, GraphML, Pajek (NET), and GraphViz (DOT), then, to describe graph structures by a

domain analyst, a simpler format is required, covering the names of entities (nodes) combined in pairs. Each pair expresses an edge of the graph and a direction (from the first node of the pair to the second). As such a simplified format, the CSV format is best suited, which, unfortunately, is not supported by the mentioned systems.

CSV2Graph



<https://bigsearch.space/uli.html>

To solve the problem, based on the library (API) of the GrahViz system, a program was developed that became the basis of the CSV2Graph service, currently available on the Internet at <https://bigsearch.space/uli.html>. This service provides primary analysis and display of graphs, information about which corresponds to the CSV format, each record of which is the name of a pair of entities separated by a semicolon.

Data is entered into a special text box, after which the graph type is selected (directed / non-directional) and the graph is displayed by activating the Draw key. In Fig. 10. The completed data entry form for the CSV2Graph service is shown.

CSV => Graph

Insert text - pairs of concepts separated by a semicolon:

```
Chatbot;AI  
ChatGPT;Chatbot  
LLM;Machine Learning  
Machine Learning;AI  
AI;LLM  
Bert;Chatbot  
GPT-4;ChatGPT  
GPT-4;LLM  
Bert;LLM  
Bert;Google
```

Directed Draw

Figure 10. Data for further analysis and visualization

As a result of the program execution, a display of directed and undirected graphs is formed, the nodes are ranked by degree and colored, and the thickness and direction of the edges are determined. Laying a graph on a plane is performed by methods built into GraphViz. At the same time, an image of the graph is formed in the SVG format, which makes it possible to form hyperlinks leading to the search forms of the Google News system from the nodes and edges of the graph. Scalable Vector Graphics (SVG) is a format for defining two-dimensional graphics using XML. It supports interactivity and animation. The SVG specification, an open standard developed by the World Wide Web Consortium since 1999, allows images to be scaled without losing quality. These images are stored in XML text files, making them searchable, indexable, scriptable, and compressible. Starting from 2011, all major desktop browsers began supporting SVG.

The data in the SVG file is text, not an image, so it is possible to embed interactive features, such as hyperlinks to web resources, which is used in the CSV2Graph service. SVG documents are easily integrated with HTML and XHTML documents. Also, SVG is an open standard. Unlike some other formats, SVG is not proprietary.

On Fig. 11 shows a network generated using CSV2Graph by processing the data presented in Fig. 11.

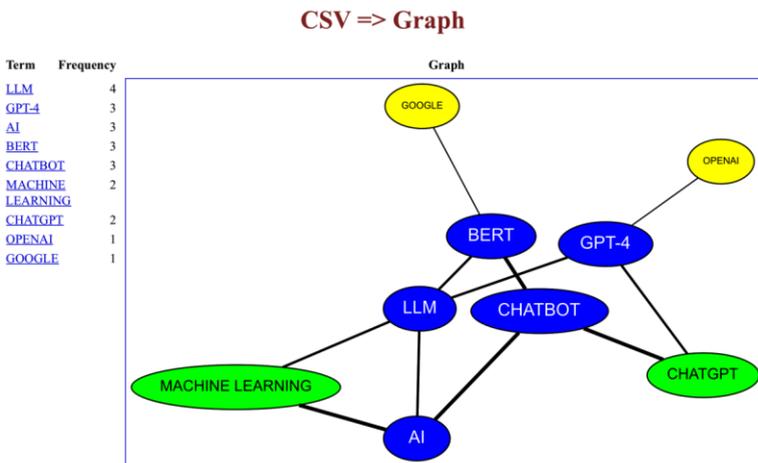


Figure 11. Graph generated based on the entered data in CSV format

The presented service is successfully used in analytical research to display large models of various subject areas, presented in many languages.

4. Building Semantic Networks with ChatGPT

We are going to create semantic networks by referring to ChatGPT for semantically related pairs of concepts in a particular subject area. We will save the received pairs, thereby gradually expanding our network. We will change the semantic meaning of the links and also vary the prompts to get different pairs of words. Thus, we will get a complete set of network types: directed, unweighted; non-directional, weighted; directed, unweighted.

The most interesting will be causal networks. They can serve as a basis for the formation of scenarios and will allow us to better understand the relationships between concepts in the subject area under study. The creation of such networks will help us to identify causal relationships between various elements and events, which will open up new opportunities for analyzing and understanding important processes and phenomena. This can be especially useful when creating scenarios for various situations and making decisions based on predictions and simulations.

In today's world, the ability to manage and process this data has become a top priority for businesses and individuals alike. Big Data is revolutionizing our understanding of the world, but extracting meaningful information from such vast amounts of data is a worthy task. This is where ChatGPT comes in, an artificial intelligence system that allows users to efficiently manage and analyze their Big Data.

The ChatGPT system simplifies the processing and analysis of complex data, allowing users to quickly and efficiently gain the necessary knowledge, solving completely different problems.

The main objective of this work is to show the capabilities of ChatGPT for solving problems of semantic analysis and visualization, which allows it to be considered a useful tool for network analysis of documents, and the

formation of domain models.

Network analysis and visualization are essential tools for understanding relationships between data points and uncovering hidden patterns that can help individuals and businesses to optimize their workflow.

For network analysis and visualization, such powerful network analysis and visualization tools as Neo4j and Gephi can be used in conjunction with the ChatGPT system. Neo4j is a powerful server graph database that allows you to efficiently store and process large amounts of related data. More user-friendly, Gephi is an interactive personal platform that makes it easy to explore and visualize complex networks. It is this tool that will be used to demonstrate the examples in this article.

There are many works and concepts on the significance of such networks for science and practice, and special programs and languages for describing data have been created [Berners-Lee, 2001], [McGuinness, 2004]. The methodology allows the creation of networks, and graph structures with different topological features: simple undirected, weighted, directed weighted, and directed with labeled edges. And of course, the methodology is based on the use of the ChatGPT system, trained on a linguistic network, so no formula or explicitly defined calculation will be used in the article. This is "A New Kind of Science", as Stephen Wolfram wrote when he was talking about cellular automata [Wolfram, 2002]. By the way, today he also writes a lot about the possibilities of ChatGPT [Wolfram, 2023]. The methodology also includes visualization of the created networks. For this, the Gephi program (gephi.org) is used - the most popular program for visualizing graph structures with a free license [Cherven, 2015]. The CSV format is quite suitable for loading data into the Gephi environment, so all requests to ChatGPT will be accompanied by a format requirement.

Forming a simple undirected network Formation of networks of characters

So, in our plans, we have the description of procedures for forming networks with different properties. Let's move on to the description of tasks and solutions. We will demonstrate how character networks are formed using the example of analyzing Thor. It should be noted that not every book was covered by ChatGPT during its training. Obviously, the system "knows" a book of this scale. To build a network, it is sufficient to obtain a CSV file and load it into the Gephi program.

Forming a simple undirected network

The ChatGPT system can help in obtaining the contents of such a file in the required format (fields corresponding to character names separated by semicolons). To do this, you can apply, for example, this prompt to the ChatGPT system:

Prompt 1	→ Give me the 20 most common pairs of Torah (Bible) characters in the format "character 1; character 2". Each entry on a new line
-----------------	---

As a result, ChatGPT provides the following response:

1. Man; Home
2. Abraham; Isaac
3. Jacob; Esau
4. Jacob; Rachel
5. Jacob; Leah
6. Joseph; Potiphar's wife
7. Joseph; Pharaoh
8. Moses; Aaron
9. Moses; Pharaoh
10. Joshua; Caleb
- ...

After uploading the data into the Gephi system, we choose the size of

nodes proportional to their degree (number of adjacent links) and divide the network into clusters based on modularity criteria, resulting in a visual graph (Fig. 12).

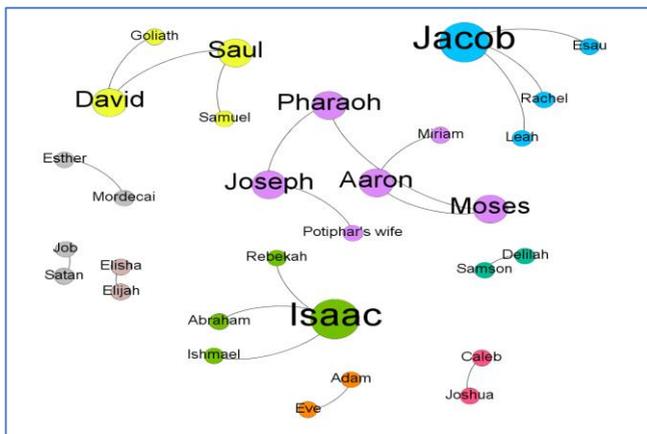


Figure 12. Undirected network of 20 main characters of Torah (Bible) according to ChatGPT program evaluation.

So, the first simple network has been built and it can be seen that the story of Passover occupies the central part of the Torah (Bible), which generally confirms the correctness of the selection of Torah characters by the ChatGPT system.

Creating a weighted network

An already solved problem may be justifiably complicated by the requirement to indicate the weight of individual connections in the constructed network. Note that the "weight" is determined by the ChatGPT system according to its internal rules that are unknown to us. We can request the frequency of joint mentions of pairs of characters, but in this

case, ChatGPT will most likely provide us with a program, most likely in Python, that can be used to calculate the frequencies. In our simplest case, we will rely on the artificial intelligence system and make the request.

Prompt 2	→ Give me the 20 most common pairs of actors (not letters) of the Torah (Bible) with the weight of the joint appearance of such pairs in the Torah in the format "actor 1; actor 2; weight". Each entry is on a new line.
-----------------	---

As a result, we will get an answer containing the requested information.

As I mentioned earlier, the Torah (Bible) does not contain actors in the traditional sense, but it does contain a cast of characters such as prophets, leaders, and patriarchs. Here are the 20 most common pairs of characters in the Torah based on their joint appearance, along with their weights: <ol style="list-style-type: none">1. Moses; God; weight: 7672. Abraham; God; weight: 2573. Aaron; Moses; weight: 1774. Isaac; Abraham; weight: 955. Jacob; Isaac; weight: 836. Joseph; Jacob; weight: 727. Sarah; Abraham; weight: 588. Joshua; Moses; weight: 559. Miriam; Aaron; weight: 4410. Pharaoh; Moses; weight: 41 ...

To display a weighted network in Gephi, you can load the names as in the previous case, and the weight values in the "Edges" section of the "Data Laboratory" mode in this software.

Creating a directed weighted network

Considering the possibility of building a weighted network once again, let's further complicate the task by constructing a weighted directed network. We will determine the direction of the connection as the

relationship of control between the characters, namely, the first character gives commands and controls the second. To do this, the following query is created:

Prompt 3	→ Give me the 50 most common pairs of main characters (characters, not letters) of the Torah (Bible) with the weight of the joint appearance of such pairs in the Torah. Moreover, the first character commands something to the second. In the format "character 1; character 2; weight". Each entry is on a new line. I don't need the program code, but the result.
-----------------	--

The ChatGPT system solves this problem in the following way:

Here are the 50 most common pairs of main characters in the Torah (Bible) where the first character commands the second, along with their weight of joint appearance: <ol style="list-style-type: none">1. Moses; Aaron; 2142. God; Moses; 1073. Moses; Joshua; 934. Moses; Pharaoh; 685. God; Aaron; 636. God; Joshua; 517. Moses; Bezalel; 508. Moses; Israelites; 459. Moses; Elders; 4110. Moses; Levites; 36 ...

Creating a signed network

In all previous examples, values were assigned to nodes, and the meaning of relationships was implied but not specified. Now we will show how to solve the problem of forming and displaying a network with edges that express various communicative relationships. To do this, we will create a prompt that explicitly asks the ChatGPT system to indicate the communication word between the characters. Naturally, a network with directed connections is topologically formed.

Prompt 4	→ Give me the 10 most common pairs of main characters of the Torah. Moreover, the first character communicates something to the second. In the format "character 1; character 2; communication word". Each entry is on a new line. I don't need the program code, but the result.
-----------------	---

As a result, we get the answer.

<ol style="list-style-type: none"> 1. God; Adam; creation 2. God; Eve; command 3. Adam; Eve; temptation 4. Cain; Abel; jealousy 5. Noah; Ham; blessing 6. Abraham; Isaac; sacrifice 7. Isaac; Rebekah; marriage 8. Jacob; Esau; birthright 9. Joseph; his brothers; dreams 10. Moses; Aaron; leadership

Let's ask to continue:

Prompt (continuation)	→ OK, another
---------------------------------	---------------

We receive:

<ol style="list-style-type: none"> 11. God; Noah; covenant 12. Abraham; Lot; separation 13. Jacob; Rachel; love 14. Jacob; Leah; deception 15. Joseph; Potiphar's wife; temptation 16. Moses; Pharaoh; Plagues 17. God; Moses; commandments 18. Aaron; Miriam; support 19. Isaac; Jacob; blessing 20. Hagar; Ishmael; exile

To obtain weighted signed networks, one can use the following query:

Prompt 5	→ List the 10 most important pairs of concepts in Torah only in the table format: "concept 1"," concept 2", and "importance". For every concept add inline in (Hebrew name, transliteration), Number rows. Add an inline English translation of concepts. For the "importance," column add (weight). Add another column that will represent the connection between pairs. Example: this row is related to rows x,y... and (explanation). Build a table that will represent a graph-like relationship.
-----------------	---

Emulating multiple experts through repeated prompt generation

(Another weighting method is just for directional networks)

The ChatGPT system can provide different answers when processing text at different times, all of which are reasonable from a human logic standpoint. Each of these answers can be perceived as the response of an artificial expert. By generalizing the answers of a multitude of such experts, a more complete and accurate answer can be obtained.

Each such answer can be perceived as the answer of some artificial expert [Mnih, 2016], [Silver,2017].

It should be noted that sometimes the ChatGPT system may start giving the same answer for repeatedly entered queries, which should alert the user as this should not happen in such tasks. Presumably, the system has exhausted the resources allocated for the specific user. The solution to this problem is very simple - just reload the session. To demonstrate this possibility, a simple query can be repeated multiple times and the results can be combined.

Prompt 6	→ Give me the 10 most common pairs of main characters of the Torah. Moreover, the first character commands (we can use the word communicate instead) something to the second. In the format "character 1; character 2". Each entry is on a new line. I don't need the program code, but the result.
-----------------	---

Result of ChatGPT output:

As an AI language model, I do not have access to information about character pairs in the Torah (Bible) where one character commands another. However, I can provide you with a list of the most common pairs of main characters in the Torah based on their appearances in the text. Please note that this list does not consider any commands or power dynamics between characters.

1. Abraham; Sarah
2. Jacob; Esau
3. Jacob; Rachel
4. Jacob; Leah
5. Joseph; his brothers
6. Moses; Aaron
7. Moses; Pharaoh
8. Moses; Miriam
9. Joshua; Moses
10. Isaac; Rebekah

As a result of loading the merged CSV file into the Gephi environment, scaling node sizes, and clustering by modularity classes, we obtain a network of the main characters in the Torah (Bible) (Fig. 13).

Formation of a network of concepts - Subject Domain model

Formation of a basic undirected network

In all previous examples, we formed networks of concepts related to a specific book - the Torah (Bible), which was, of course, covered by the ChatGPT system during its training. Next, let's consider the task of forming a network of concepts related to an entire subject area. Obviously, ChatGPT "knows" the basic concepts from the most important subject areas.

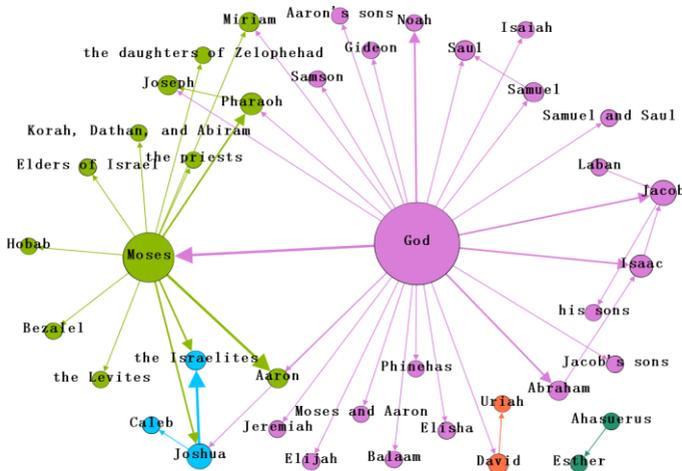


Figure 13. A network of main characters in the Torah is generated using the ChatGPT system by emulating an artificial expert swarm

For example, we are interested in the field of information law, so we will ask the ChatGPT system to find 40 pairs of the most popular concepts related to each other in this subject area. To do this, we will ask the ChatGPT system to perform the following query:

Prompt	→ Give me 40 pairs of the most important concepts of the subject area "Information law" in the format "concept 1; concept 2". Each entry is on a new line.
---------------	--

As a result, we get:

- | |
|--|
| <ol style="list-style-type: none"> 1. Data protection; Privacy 2. Copyright; Fair Use 3. Cybersecurity; Hacking 4. Intellectual Property; Patents 5. Freedom of Speech; Censorship 6. Digital Rights Management; Encryption 7. Trademark; Trade Secret 8. Open Access; Public Domain 9. Online Harassment; Cyberbullying 10. Net Neutrality; Internet Service Providers ... |
|--|

data to the Gephi environment, so all requests to ChatGPT will be accompanied by a requirement for the format. Causal relationships are necessary when models are implemented in critically important areas such as healthcare, disaster management, theft detection, finance, and law [Gunning, 2017].

The formed causal networks provide the possibility of further transition to scenario analysis. The main problem that arises when conducting scenario analysis based on causal networks is precisely the creation of such systems, which in traditional cases requires large resource costs, attracting experts. There are also successful attempts at automated formation of causal networks, for example, in [Solat, 2023] a rule-based SCANNER system is presented, which transforms raw text into causal networks using a set of natural language processing tools.

The approach proposed by the authors for forming a swarm of virtual experts [Lande1, 2023], [Lande1, 2023] will significantly simplify and speed up the process of forming causal networks.

Formation of a network based on simple hierarchical access to GPT

So, our plans include describing the procedures for forming cause-and-effect networks in the field of cybersecurity through hierarchical refinement. Let's move on to the description of tasks and their solutions. It should be noted that not every subject area was sufficiently covered by ChatGPT during its training. Obviously, the system "knows" a subject area of such a scale. To build a network, it is necessary to obtain a CSV file and upload it to the Gephi program.

Let's say, for example, we are interested in the issue of data leakage. We will ask ChatGPT to provide known causes of this phenomenon. The central node of the future network should be the concept of "data_leakage".

Successful processing of such a request will determine the second level of the hierarchy – concepts related to data leakage - its causes. After that, for each such concept, a set of reasons that influenced it is also requested. This process can continue indefinitely, but in this work, we will stop at three levels. Obviously, some concepts of the third level can influence different concepts of the second level, as well as the concept of the first level directly. Theoretically, looping is also possible, which can be interpreted as the paradox of primacy (which came first, the chicken or the egg?). Thus, despite the hierarchical formation of such a causal network, the resulting network will not be a strictly hierarchical structure.

By offering ChatGPT to process a certain prompt, we will get a set of reasons for the primary concept. The ChatGPT system can help in obtaining the content of the CSV file (fields corresponding to character names, separated by a semicolon). To do this, you can use, for example, such a request (prompt) to the ChatGPT system:

→ List the causes of data leakage in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause;data leakage". Each such entry - from a new line

The system gives an answer of approximately this kind:

human error; data leakage
weak passwords; data leakage
insider threats; data leakage
misconfigured systems; data leakage
phishing attacks; data leakage
unpatched software; data leakage
malware infection; data leakage
social engineering; data leakage
third-party access; data leakage
stolen devices; data leakage

Prompts of the next level will relate to the concepts presented in the

answer and have a form fully corresponding to the primary prompt, for example:

→ List the causes of human error in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; human error ". Each such entry - from a new line

The set of all prompts and answers:

Prompt 1. Data leakage

→ List the causes of data leakage in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause;data leakage". Each such entry - from a new line

Answer:

human error; data leakage
weak passwords; data leakage
insider threats; data leakage
misconfigured systems; data leakage
phishing attacks; data leakage
unpatched software; data leakage
malware infection; data leakage
social engineering; data leakage
third-party access; data leakage
stolen devices; data leakage

Prompt 1.1. Human error

→ List the causes of human error in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; human error ". Each such entry - from a new line

Answer:

Poor decision-making;human error
Lack of attention;human error
Negligence;human error
Lack of knowledge;human error
Misconfiguration;human error
Improper handling;human error

Complacency;human error
Carelessness;human error
Lack of training;human error
Misunderstanding;human error

Prompt 1.2. Weak passwords

→ List the causes of weak passwords in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; weak passwords ". Each such entry - from a new line

Answer:

Password sharing;weak passwords
Reuse of passwords; weak password
Easy-to-guess passwords;weak passwords
Lack of complexity;weak passwords
Inadequate length;weak passwords
Lack of randomness;weak passwords
No two-factor;weak passwords
Non-rotation;weak passwords
Dictionary words;weak passwords
Personal information;weak passwords

Prompt 1.3. Insider threats

→ List the causes of insider threats in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; insider threats ". Each such entry - from a new line

Answer:

Unhappy employees;insider threats
Access abuse;insider threats
Data theft;insider threats
Fraudulent activity;insider threats
Carelessness;insider threats
Social engineering;insider threats
System compromise;insider threats
Lack of monitoring;insider threats
Privilege abuse;insider threats
Personal gain;insider threats

Prompt 1.4. Misconfigured systems

→ List the causes of misconfigured systems in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause;misconfigured systems ". Each such entry - from a new line

Answer:

Human error;misconfigured systems
Lack of expertise;misconfigured systems
Incomplete updates;misconfigured systems
Improper settings;misconfigured systems
No testing;misconfigured systems
Inadequate security standards;misconfigured systems
Confusing interface;misconfigured systems
Legacy technology;misconfigured systems
Non-compliance;misconfigured systems
Mismanagement;misconfigured systems

Prompt 1.5. Phishing attacks

→ List the causes of phishing attacks in cyber security. The reason is to use no more than three words. The results should be presented in the format " cause;phishing attacks ". Each such entry - from a new line

Answer:

Deceptive emails;phishing attacks
Impersonation tactics;phishing attacks
Link manipulation;phishing attacks
Malicious attachments;phishing attacks
Spear phishing;phishing attacks
Fake websites;phishing attacks
Social engineering;phishing attacks
Trusted sender forgery;phishing attacks
Brand impersonation;phishing attacks
Unsecured networks;phishing attacks

Prompt 1.6. Unpatched software

→ List the causes of unpatched software in cyber security. The reason is to use no more than three words. The results should be presented in the

format "cause;unpatched software ". Each such entry - from a new line

Answer:

Lack of updates;unpatched software
Vendor negligence;unpatched software
Time-consuming process;unpatched software
Legacy technology;unpatched software
Misconfigured auto-updates;unpatched software
Inadequate testing;unpatched software
Human error;unpatched software
Lack of awareness;unpatched software
Budget constraints;unpatched software
No backup plan;unpatched software

Prompt 1.7. Malware infection

→ List the causes of malware infection in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; malware infection". Each such entry - from a new line

Answer:

Drive-by downloads;malware infection
Phishing attacks;malware infection
Email attachments;malware infection
File sharing;malware infection
Unpatched systems;malware infection
Infected software;malware infection
Vulnerable apps;malware infection
Social engineering;malware infection
Compromised websites;malware infection
Fake software;malware infection

Prompt 1.8. Social engineering

→ List the causes of social engineering in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; social engineering ". Each such entry - from a new line

Answer:

Pretexting;social engineering

Phishing scams;social engineering
Baiting tactics;social engineering
Quid pro quo;social engineering
Tailgating;social engineering
Tech support scam;social engineering
Watering hole attacks;social engineering
Scareware;social engineering
Human error;social engineering
Insider threat;social engineering

Prompt 1.9. third party access

→ List the causes of third-party access in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; third-party access ". Each such entry - from a new line

Answer:

Vendor vulnerability;third-party access
Inadequate oversight;third-party access
Weak subcontractors;third-party access
Shared networks;third-party access
Outsourced support;third-party access
Partner negligence;third-party access
Poor authentication practices;third-party access
Contractual violations;third-party access
Supply chain weakness;third-party access
Insufficient audit;third-party access

Prompt 1.10. Stolen devices

→ List the causes of stolen devices in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause;stolen devices". Each such entry - from a new line

Answer:

Unsecured devices;stolen devices
Lack of encryption;stolen devices
Carelessness;stolen devices
Physical theft;stolen devices
Weak passwords;stolen devices
Lost property;stolen devices

Insider threat;stolen devices No remote wipe;stolen devices Public Wi-Fi use;stolen devices Human error;stolen devices

The combined answers of ChatGPT in one CSV file are uploaded for analysis and visualization in the Gephi program. After loading the obtained data into the Gephi system, we select the node size proportional to the degree (number of adjacent connections) and dividing the network into clusters according to the modularity criterion, we get a clear graph (Fig. 16).

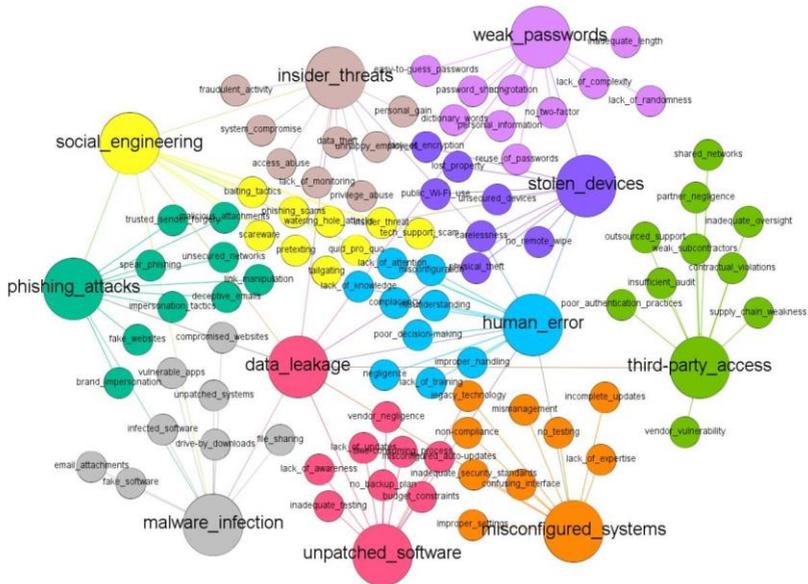


Figure 16. The Directed primary causal network obtained by simple hierarchical access to ChatGPT

The main parameters of the nodes in this network:

1. Parameters of the most important nodes of network 1 (primary causal network)

Concept	In-Degree	Out-Degree	Clustering Coefficient	Betweenness Centrality
human_error	10	5	0,02381	77,33333
social_engineering	10	4	0,027473	67,5
phishing_attacks	10	2	0,022727	18
weak_passwords	10	2	0,007576	20
misconfigured_systems	10	1	0,009091	8,5
stolen_devices	10	1	0,027273	6,833333
unpatched_software	10	1	0,009091	8,5
insider_threats	10	1	0,009091	8,333333
malware_infection	10	1	0,027273	8
third-party_access	10	1	0	10

2. Parameters of the most important nodes of network 2 (complete causal network)

Concept	In-Degree	Out-Degree	Clustering Coefficient	Betweenness Centrality
social_engineering	31	4	0,011765	118,1667
data_leakage	24	0	0,032609	0
phishing_attacks	24	2	0,027692	58,91667
insider_threats	24	2	0,021538	55,16667
malware_infection	23	1	0,014493	18
human_error	23	7	0,032184	152,25
unpatched_software	22	1	0,005929	11,16667
unpatched_systems	27	2	0,006158	47,41667
misconfigured_systems	24	1	0,011667	17,91667
weak_passwords	24	3	0,004274	72

3. Parameters of the most important nodes of the network 3
(generalized causal network)

Concept	In-Degree	Out-Degree	Clustering Coefficient	Betweenness Centrality
data_leakage	10	0	0,077778	0
human_error	7	5	0,045455	36,33333
social_engineering	6	3	0,041667	18
insider_threats	6	1	0,047619	4,5
unpatched_software	6	1	0,047619	4,333333
phishing_attacks	5	2	0,071429	8
malware_infection	5	1	0,133333	2
weak_passwords	5	1	0	5
unpatched_systems	3	2	0,15	10,33333
misconfigured_systems	3	1	0	2,5
misconfigured	1	0	0	0

The most influential nodes in this network (highest Out-Degree) are: human_error (5), social_engineering (4), weak_passwords(3), and phishing_attacks(2). It is evident that the formed network is weakly connected, and incomplete, and the concepts represented in it may not accurately reflect causes and consequences. We will consider this as a network obtained from a survey of only one artificial expert.

Forming a Network Based on Hierarchical Invocation of Swarm Virtual Experts to ChatGPT

The ChatGPT system can provide different answer options at different times during text processing, with some being more accurate and logically sound from a human perspective. Each such answer can be perceived as an answer from some virtual expert [3]. It can be assumed that by generalizing

answers from multiple (swarm) similar experts, we can obtain a more complete and accurate response. By implementing swarm virtual experts, we ask the same prompts several times related to both first- and second-level hierarchies. After receiving responses from the system, we combine them into a single CSV file for analysis and visualization using Gephi software. Loading the obtained data into Gephi results in the graph shown in Fig. 17.

In practice, the network can be expanded until it becomes sufficiently complete according to human expert evaluation.

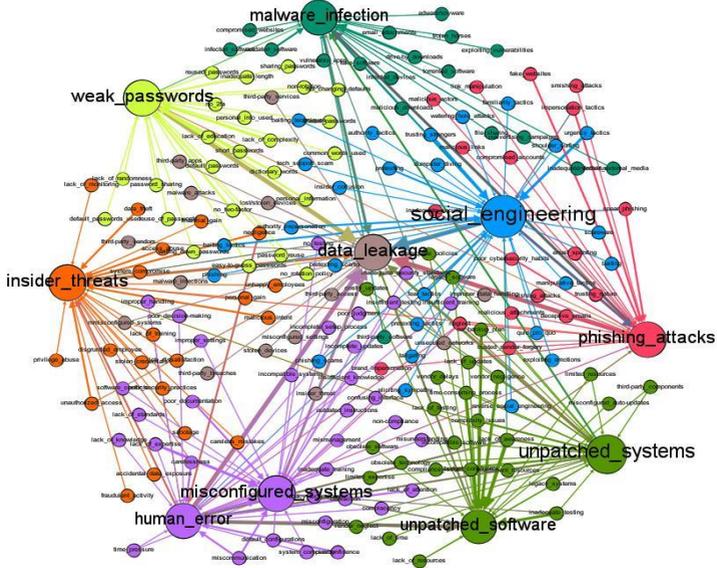


Figure 17. Directed full causal network obtained by hierarchically querying a swarm of virtual experts to ChatGPT

The most influential nodes in this network (with the highest Out-Degree) are: human_error (7), social_engineering (4), weak_passwords(3), phishing_attacks(2), unpatched_systems(2), insider_threats(2).

As we can see, the number of important concepts has increased compared to the previous case.

Formation of a network based on a generalization of hierarchical querying a swarm of virtual experts to ChatGPT

The graph formed in the previous example, having relatively high completeness of concepts, may contain inaccurate information mistakenly provided by ChatGPT when processing individual prompts. Assuming that the probability of encountering similar errors is relatively small, it is possible to exclude from consideration concepts that occur less frequently than a given threshold when constructing a network. In the case presented below (Fig. 18), concepts that occurred less than twice were not considered.

The most influential nodes in this network (with the highest Out-Degree) are: human_error (5), social_engineering (3), phishing_attacks(2), unpatched_systems(2).

Based on expert assessments, it can be concluded that the primary causal network obtained by simple hierarchical querying to ChatGPT covers the largest number of concepts that are relatively weakly connected (the network is close to hierarchical), but thanks to its completeness, it can be good "raw material for subsequent analytical processing."

The statistically processed second network, a causal network obtained by hierarchically querying a swarm of virtual experts to ChatGPT, is more accurate than the primary network and finally, the third network obtained by generalizing hierarchical querying from a swarm of virtual experts to ChatGPT has the highest average clustering coefficient indicating greater interaction between individual concepts influencing goals in this causality chain. This type of network is likely most suitable for further scenario analysis.

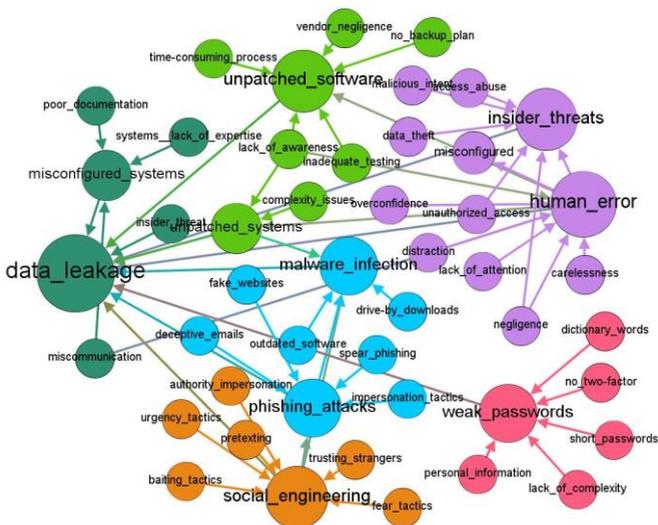


Figure 18. Directed causal network obtained by generalizing the hierarchical querying of a swarm of virtual experts to ChatGPT

In the last two paragraphs of this section we have demonstrated:

- The convenience of using ChatGPT for forming causal networks within specific subject areas such as cybersecurity is based on using ChatGPT & Gephi.
- We used a swarm-of-virtual-experts method through multiple prompt executions with ChatGPT.
- Our approach was applied specifically to cybersecurity but could be applied across various subject areas such as military, political or economic.

Furthermore, we have shown how the integration of text analytics and network analysis tools can prove to be very useful in gaining insights from large amounts of unstructured data.

For the first time, an emulation of an "expert swarm" is proposed by repeatedly executing prompts to an artificial intelligence system.

One of the most interesting aspects of the study is the proposed emulation of groups of experts through the use of AI systems. This approach has the potential to greatly enhance the efficiency of knowledge extraction and to provide deeper insights into the structure and meaning of literary works across various subjects.

A methodology has been proposed that can be used for network analysis of documents and building models for various subject areas.

Overall, this study has demonstrated the practical applications of advanced AI technology in the field of text analysis and network visualization. It has shown how the use of machine learning algorithms can help us to unlock the previously hidden insights and patterns in textual data, and to gain a deeper understanding of complex phenomena in different domains.

Limitations

However, it is important to note that this approach is not without its limitations. The interpretation of the results requires expertise in the subject area being studied, and there is still a need for human supervision to ensure the accuracy of the results.

Overall, it is clear that AI holds immense potential for transforming the field of text analysis and network visualization. While there are still many challenges that need to be addressed, this study represents an important step forward in unlocking the power of this technology to gain deeper insights into complex phenomena across a wide range of subjects.

Despite significant gains in resources (both time and human capital), it's

important to note that both constructing these causal networks and interpreting results require data scientists experienced in their respective fields and still require human observation for accuracy and precision purposes.

Cause networking techniques

In the previous paragraphs, the concept of a swarm of virtual experts was introduced. Within the second method, it is proposed to decompose the main concept into several partial concepts (in this work a decomposition into 10 partial concepts was implemented). Then for each partial concept, a similar prompt is formed allowing us to determine its main causes. This constitutes Methodology 2 which is schematically depicted on Fig. 19 (lower part) [Lande3, 2023].

Below there are provided prompts ensuring the decomposition of the main concept and ten similar prompts allow the determination of their causes:

Decomposition prompt:

→ Decompose the concept of "cyber security vulnerability" into 10 partial concepts. Each partial concept must contain no more than three words. Present the answer in the form: "partial concept; cyber security vulnerability". Each entry on a separate line.

Answer:

Weak encryption; cyber security vulnerability
Malware infections; cyber security vulnerability
Zero-day exploits; cyber security vulnerability
Weak passwords; cyber security vulnerability
Phishing attacks; cyber security vulnerability
Unpatched software; cyber security vulnerability
Social engineering; cyber security vulnerability
Insider threats; cyber security vulnerability

Configuration errors; cyber security vulnerability
Data breaches; cyber security vulnerability

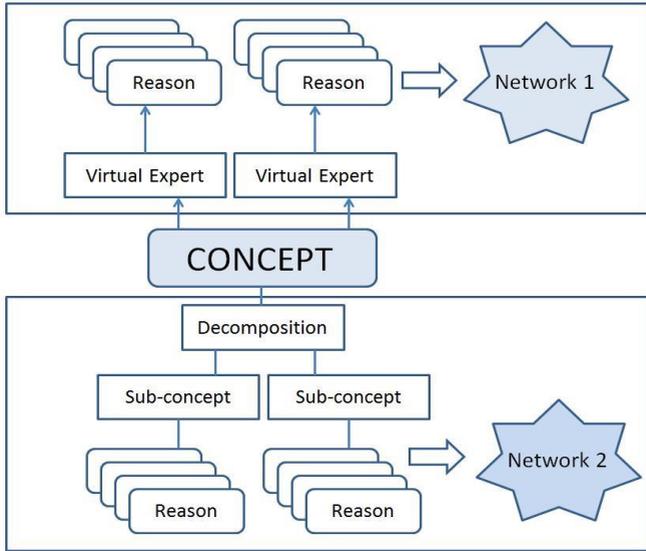


Figure 19. Two methods of constructing causal networks.

1 Reason Prompt:

→ Name 10 reasons for the concept of " Weak encryption " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Weak encryption ". Each entry on a separate line.

Answer:

Short key length; Weak encryption
Insecure algorithms; Weak encryption
Improper implementation; Weak encryption
Lack of forward secrecy; Weak encryption
Vulnerable cipher suites; Weak encryption

Brute-force attacks; Weak encryption
Poor random number generation; Weak encryption
Outdated encryption protocols; Weak encryption
Key leakage; Weak encryption
Inadequate encryption strength; Weak encryption

2 Reason Prompt:

→ Name 10 reasons for the concept of "Malware infection " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Malware infection ". Each entry is on a separate line.

Answer:

Phishing emails; Malware infection
Drive-by downloads; Malware infection
Infected attachments; Malware infection
Exploit kits; Malware infection
Malvertising campaigns; Malware infection
Trojan horses; Malware infection
Worm propagation; Malware infection
Botnet propagation; Malware infection
File-sharing networks; Malware infection
Social engineering; Malware infection

3 Reason Prompt:

→ Name 10 reasons for the concept of "Zero-day exploit " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Zero-day exploit ". Each entry is on a separate line.

Answer:

Unknown vulnerabilities; Zero-day exploit
Targeted attacks; Zero-day exploit

Advanced persistent threats; Zero-day exploit
Undetected malware; Zero-day exploit
Vendor unawareness; Zero-day exploit
Limited defense mechanisms; Zero-day exploit
Delayed patching; Zero-day exploit
Covert data exfiltration; Zero-day exploit
Escalation of privileges; Zero-day exploit
Weaponization potential; Zero-day exploit

4 Reason Prompt:

→ Name 10 reasons for the concept of "Weak passwords" as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Weak passwords ". Each entry is on a separate line.

Answer:

Easy to guess; Weak passwords
Commonly used passwords; Weak passwords
Lack of complexity; Weak passwords
Password reuse; Weak passwords
No two-factor authentication; Weak passwords
Dictionary-based passwords; Weak passwords
Short password length; Weak passwords
Lack of password rotation; Weak passwords
Inadequate password policy; Weak passwords
Poor password hygiene; Weak passwords

5 Reason Prompt:

→ Name 10 reasons for the concept of "Phishing attacks " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Phishing attacks ". Each entry is on a separate line.

Answer:

Email impersonation; Phishing attacks
Deceptive websites; Phishing attacks
Social engineering tactics; Phishing attacks
Fake login pages; Phishing attacks
Malicious attachments; Phishing attacks
Spear phishing campaigns; Phishing attacks
Phishing emails; Phishing attacks
Phone-based phishing; Phishing attacks
SMS phishing (smishing); Phishing attacks
Credential theft; Phishing attacks

6 Reason Prompt:

→ Name 10 reasons for the concept of "Unpatched software" as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Unpatched software ". Each entry is on a separate line.

Answer:

Known vulnerabilities; Unpatched software
Exploit availability; Unpatched software
Delayed updates; Unpatched software
Software bugs; Unpatched software
Lack of maintenance; Unpatched software
Incompatible patches; Unpatched software
Ignored advisories; Unpatched software
Legacy systems; Unpatched software
Limited resources; Unpatched software
Patch fatigue; Unpatched software

7 Reason Prompt:

→ Name 10 reasons for the concept of "Social engineering " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Social engineering ". Each

entry is on a separate line.

Answer:

Manipulates human psychology; Social engineering
Exploits trust relationships; Social engineering
Impersonates legitimate entities; Social engineering
Phishing attacks; Social engineering
Baiting techniques; Social engineering
Pretexting scenarios; Social engineering
Tailgating entry points; Social engineering
Psychological manipulation; Social engineering
Deceptive communication methods; Social engineering
Exploits human error; Social engineering

8 Reason Prompt:

→ Name 10 reasons for the concept of "Insider threats" as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Insider threats ". Each entry is on a separate line.

Answer:

Unauthorized data access; Insider threats
Data leakage; Insider threats
Sabotage or theft; Insider threats
Malicious intent; Insider threats
Insider collusion; Insider threats
Privilege abuse; Insider threats
Misuse of credentials; Insider threats
Social engineering manipulation; Insider threats
Unintentional mistakes; Insider threats
Negligence or carelessness; Insider threats

9 Reason Prompt:

→ Name 10 reasons for the concept of "Configuration errors " as part of

the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Configuration errors ". Each entry is on a separate line.

Answer:

Misconfigured settings; Configuration errors
Weak access controls; Configuration errors
Insecure default configurations; Configuration errors
Lack of encryption; Configuration errors
Improper firewall rules; Configuration errors
Unpatched software; Configuration errors
Vulnerable network services; Configuration errors
Misconfigured permissions; Configuration errors
Weak password policies; Configuration errors
Failure to update; Configuration errors

10 Reason Prompt:

→ Name 10 reasons for the concept of "Data breaches " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; Data breaches ". Each entry is on a separate line.

Answer:

Insider threats; Data breaches
Phishing attacks; Data breaches
Weak passwords; Data breaches
Unpatched systems; Data breaches
Social engineering; Data breaches
Malware infections; Data breaches
Misconfigured security; Data breaches
Third-party breaches; Data breaches
Stolen devices; Data breaches
Lack of encryption; Data breaches

Method 1: Repeated execution of the same prompt

As part of the first methodology, multiple executions of the same prompt to the ChatGPT system are proposed, that is, emulating the activity of multiple experts solving the same task. Thus, ChatGPT is proposed to work out a certain prompt in order to obtain a set of reasons for the primary concept (cyber security vulnerability). The ChatGPT system can help in obtaining content for a formatted file (fields corresponding to character names separated by semicolons). For example, such a simple prompt is used:

→ Name 10 reasons for the term "cyber security vulnerability". Each reason must contain no more than three words. Present the answer in the form: "reason; cyber security vulnerability". Each entry is on a separate line.

The system gives a response like this:

Software bugs; cyber security vulnerability
Misconfigured systems; cyber security vulnerability
Weak encryption; cyber security vulnerability
Unpatched software; cyber security vulnerability
Social engineering attacks; cyber security vulnerability
Phishing scams; cyber security vulnerability
Insider threats; cyber security vulnerability
Malware infections; cyber security vulnerability
Network vulnerabilities; cyber security vulnerability
Data leakage; cyber security vulnerability

The results of the 10-fold application of this prompt:

Prompt: →Name 10 reasons for the term "cyber security vulnerability". Each reason must contain no more than three words. Present the answer in the form: "reason; cyber security vulnerability". Each entry is on a separate line.

<p>1</p> <p>Software bugs; cyber security vulnerability</p> <p>Misconfigured systems; cyber security vulnerability</p> <p>Weak encryption; cyber security vulnerability</p> <p>Unpatched software; cyber security vulnerability</p> <p>Social engineering attacks; cyber security vulnerability</p> <p>Phishing scams; cyber security vulnerability</p> <p>Insider threats; cyber security vulnerability</p> <p>Malware infections; cyber security vulnerability</p> <p>Network vulnerabilities; cyber security vulnerability</p> <p>Data leakage; cyber security vulnerability</p>	<p>2</p> <p>Human error; cyber security vulnerability</p> <p>Weak access controls; cyber security vulnerability</p> <p>Software vulnerabilities; cyber security vulnerability</p> <p>Network misconfigurations; cyber security vulnerability</p> <p>Inadequate security protocols; cyber security vulnerability</p> <p>Insufficient user awareness; cyber security vulnerability</p> <p>Third-party integrations; cyber security vulnerability</p> <p>Zero-day exploits; cyber security vulnerability</p> <p>Insecure coding practices; cyber security vulnerability</p> <p>Lack of regular security updates; cyber security vulnerability</p>
<p>3</p> <p>Software vulnerabilities; cyber security vulnerability</p> <p>Weak passwords; cyber security vulnerability</p> <p>Network misconfigurations; cyber security vulnerability</p> <p>Social engineering attacks; cyber security vulnerability</p> <p>Phishing attempts; cyber security vulnerability</p> <p>Insider threats; cyber security vulnerability</p> <p>Lack of encryption; cyber security vulnerability</p> <p>Unpatched systems; cyber security vulnerability</p> <p>Malware infections; cyber security vulnerability</p> <p>Weak authentication; cyber security vulnerability</p>	<p>4</p> <p>Inadequate access controls; cyber security vulnerability</p> <p>Misconfigured firewalls; cyber security vulnerability</p> <p>Exploitable software flaws; cyber security vulnerability</p> <p>Lack of security patches; cyber security vulnerability</p> <p>Insider misuse; cyber security vulnerability</p> <p>Weak encryption algorithms; cyber security vulnerability</p> <p>Vulnerable third-party dependencies; cyber security vulnerability</p> <p>Unsecure wireless networks; cyber security vulnerability</p> <p>Lack of intrusion detection; cyber security vulnerability</p> <p>Human negligence; cyber security vulnerability</p>

<p>5</p> <p>Misconfigured cloud services; cyber security vulnerability</p> <p>Zero-day vulnerabilities; cyber security vulnerability</p> <p>Insecure wireless networks; cyber security vulnerability</p> <p>Web application vulnerabilities; cyber security vulnerability</p> <p>Improper handling of sensitive data; cyber security vulnerability</p> <p>Lack of system and network monitoring; cyber security vulnerability</p> <p>Exploitable software vulnerabilities; cyber security vulnerability</p> <p>Insider attacks and privilege misuse; cyber security vulnerability</p> <p>Lack of regular security assessments; cyber security vulnerability</p> <p>Third-party software dependencies; cyber security vulnerability</p>	<p>6</p> <p>Weak or outdated encryption algorithms; cyber security vulnerability</p> <p>Unsecure remote access mechanisms; cyber security vulnerability</p> <p>Inadequate network segmentation; cyber security vulnerability</p> <p>Failure to apply security patches; cyber security vulnerability</p> <p>Lack of secure coding practices; cyber security vulnerability</p> <p>Social media and phishing scams; cyber security vulnerability</p> <p>Insider threats and privilege escalation; cyber security vulnerability</p> <p>Weak or default passwords; cyber security vulnerability</p> <p>Lack of robust authentication mechanisms; cyber security vulnerability</p> <p>Vulnerable or unpatched third-party applications; cyber security vulnerability</p>
<p>7</p> <p>Software bugs; cyber security vulnerability</p> <p>Weak passwords; cyber security vulnerability</p> <p>Unpatched systems; cyber security vulnerability</p> <p>Social engineering attacks; cyber security vulnerability</p> <p>Phishing scams; cyber security vulnerability</p> <p>Insider threats; cyber security vulnerability</p> <p>Malware infections; cyber security vulnerability</p> <p>Network vulnerabilities; cyber security vulnerability</p> <p>Data leakage; cyber security vulnerability</p> <p>Misconfigured systems; cyber security vulnerability</p>	<p>8</p> <p>Lack of encryption; cyber security vulnerability</p> <p>Unsecure network protocols; cyber security vulnerability</p> <p>Weak access controls; cyber security vulnerability</p> <p>Exploitable software vulnerabilities; cyber security vulnerability</p> <p>Insufficient security awareness; cyber security vulnerability</p> <p>Third-party integration risks; cyber security vulnerability</p> <p>Inadequate patch management; cyber security vulnerability</p> <p>Social engineering techniques; cyber security vulnerability</p> <p>Hardware vulnerabilities; cyber security vulnerability</p> <p>Insiders with malicious intent; cyber security vulnerability</p>

<p>9</p> <p>Weak authentication mechanisms; cyber security vulnerability</p> <p>Unpatched software and firmware; cyber security vulnerability</p> <p>Insider threats and misuse; cyber security vulnerability</p> <p>Malware and ransomware attacks; cyber security vulnerability</p> <p>SQL injection and code vulnerabilities; cyber security vulnerability</p> <p>Misconfigured firewalls and access controls; cyber security vulnerability</p> <p>Social engineering and phishing attacks; cyber security vulnerability</p> <p>Lack of encryption or weak encryption protocols; cyber security vulnerability</p> <p>Supply chain and third-party risks; cyber security vulnerability</p> <p>Advanced persistent threats; cyber security vulnerability</p>	<p>10</p> <p>Weak passwords; cyber security vulnerability</p> <p>Phishing attacks; cyber security vulnerability</p> <p>Unpatched software; cyber security vulnerability</p> <p>Social engineering; cyber security vulnerability</p> <p>Malware infections; cyber security vulnerability</p> <p>Insider threats; cyber security vulnerability</p> <p>Configuration errors; cyber security vulnerability</p> <p>Zero-day exploits; cyber security vulnerability</p> <p>Data breaches; cyber security vulnerability</p> <p>Weak encryption; cyber security vulnerability</p>
---	---

The ChatGPT system can provide different answers at different times when processing text. Each such answer can be perceived as the answer of a virtual expert, and by generalizing the answers of such a "swarm" of experts, a more complete answer can be obtained. By implementing a swarm of virtual experts several times with the same prompt considered earlier, the combined CSV file containing ChatGPT's responses is loaded for analysis and visualization using software developed by the authors based on GraphViz [Lambert, 2010] library (CSV2Graph).

The program interface located online at <http://bigsearch.space/uli.html> allows entering ChatGPT responses into an input field and launching directed network visualization (the "Draw" button).

The program allowed us to output a graph corresponding to the network (Fig. 20), as well as a list of all network nodes with their frequency of

occurrence. As a result of performing the above prompt ten times using Methodology 1 on this topic, we constructed a hierarchical network consisting of 77 nodes. The most frequent nodes in the network according to Methodology 1 are:

Node	Frequency
CYBER SECURITY VULNERABILITY	100
MALWARE INFECTIONS	4
INSIDER THREATS	4
WEAK PASSWORDS	3
SOCIAL ENGINEERING ATTACKS	3
PHISHING SCAMS	2
MISCONFIGURED SYSTEMS	2
NETWORK VULNERABILITIES	2
NETWORK MISCONFIGURATIONS	2
DATA LEAKAGE	2
SOFTWARE VULNERABILITIES	2
SOFTWARE BUGS	2
UNPATCHED SYSTEMS	2
WEAK ENCRYPTION	2
EXPLOITABLE SOFTWARE VULNERABILITIES	2
ZERO-DAY EXPLOITS	2
WEAK ACCESS CONTROLS	2
LACK OF ENCRYPTION	2
UNPATCHED SOFTWARE	2

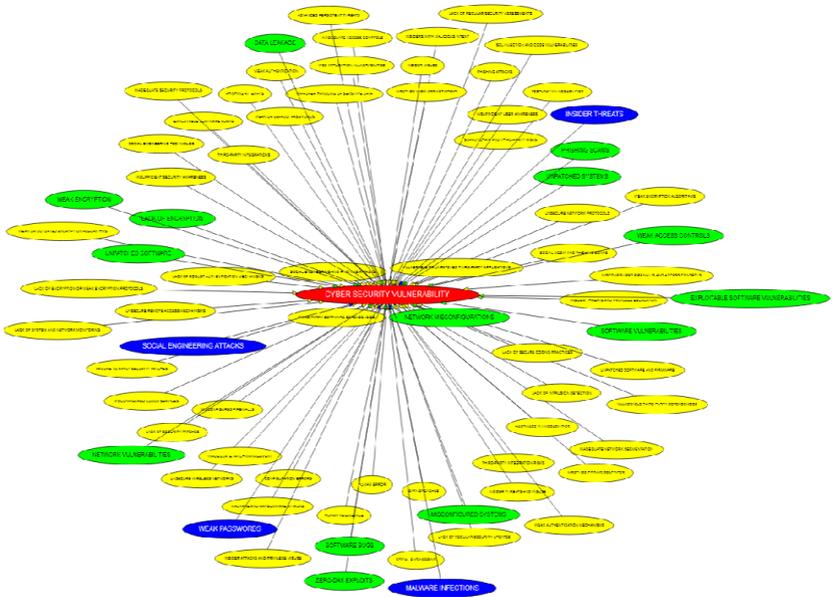


Figure 20. Network obtained using Methodology 1

Method 2: Primary problem decomposition

The second approach involves primary problem decomposition, namely breaking down the cyber security vulnerability into partial problems and then using similar prompts to obtain concepts and causes of these partial problems. The authors propose the following prompt for decomposing into 10 partial problems:

→ Decompose the concept of "cyber security vulnerability" into 10 partial concepts. Each partial concept must contain no more than three words. Present the answer in the form: "partial concept; cyber security vulnerability". Each entry on a separate line.

As a result, a typical prompt for identifying the cause within the framework of the overall problem of cyber security vulnerability is formed for each of the partial problems, for example,

→ Name 10 reasons for the concept of " **Weak encryption** " as part of the concept of "cyber security vulnerability". Each reason must contain no more than three words. In the format "reason; **Weak encryption** ". Each entry on a separate line.

The answers received from ChatGPT are combined, resulting in a network (Fig. 21). This network contains 102 nodes, which significantly exceeds the network constructed using Methodology 1. Below are the most frequently encountered concepts involved in forming this network.

Node	Frequency
PHISHING ATTACKS	13
SOCIAL ENGINEERING	13
UNPATCHED SOFTWARE	12
INSIDER THREATS	12
WEAK PASSWORDS	12
ZERO-DAY EXPLOITS	11
WEAK ENCRYPTION	11
DATA BREACHES	11
CONFIGURATION ERRORS	11
MALWARE INFECTION	10
CYBER SECURITY VULNERABILITY	10
PHISHING EMAILS	2
MALWARE INFECTIONS	2
LACK OF ENCRYPTION	2

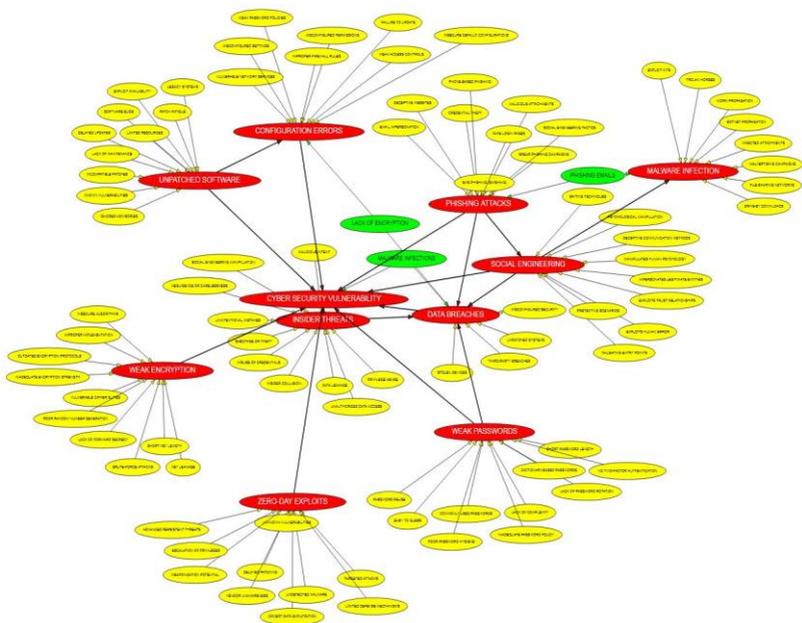


Figure 21. Network obtained using Methodology 2

Combining the results

In practice, it is evident that the network obtained by logically combining networks formed according to methods 1 and 2 will have the advantages of both approaches. Indeed, this network (Fig. 22) in the considered domain turned out to be the most complete, containing 162 nodes, among which 25 participated in forming the network more than once. Such nodes listed below can be regarded as sources for further development of causal networks in the considered domain.

Methods for forming cause networks based on the use of ChatGPT have been proposed, demonstrated, and compared. Such networks can be used in the context of system and scenario analysis tasks.

Node	Frequency
CYBER SECURITY VULNERABILITY	110
INSIDER THREATS	16
WEAK PASSWORDS	15
SOCIAL ENGINEERING	14
UNPATCHED SOFTWARE	14
PHISHING ATTACKS	14
ZERO-DAY EXPLOITS	13
WEAK ENCRYPTION	13
DATA BREACHES	12
CONFIGURATION ERRORS	12
MALWARE INFECTIONS	12
LACK OF ENCRYPTION	4
SOFTWARE BUGS	3
DATA LEAKAGE	3
UNPATCHED SYSTEMS	3
WEAK ACCESS CONTROLS	3
SOCIAL ENGINEERING ATTACKS	3
ADVANCED PERSISTENT THREATS	2
MISCONFIGURED SYSTEMS	2
PHISHING SCAMS	2
NETWORK MISCONFIGURATIONS	2
SOFTWARE VULNERABILITIES	2
PHISHING EMAILS	2
EXPLOITABLE SOFTWARE VULNERABILITIES	2
NETWORK VULNERABILITIES	2

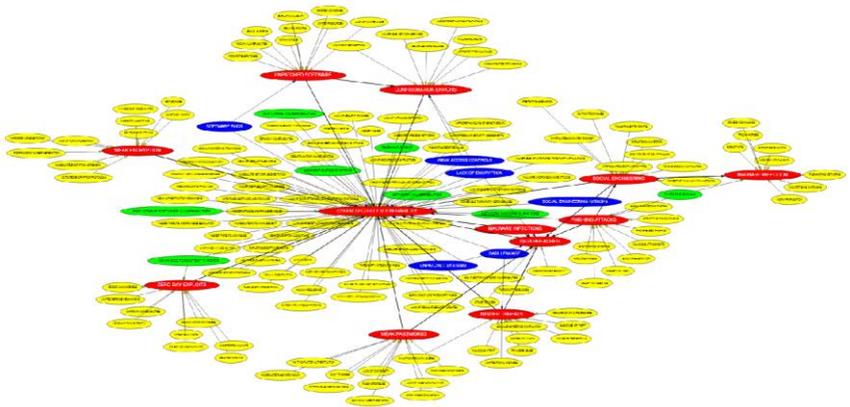


Figure 22. The network was obtained as a result of combining methods 1 and 2

The first method implemented the concept of virtual experts and showed the characteristics of networks formed using this approach. Based on an analysis of such a network, it can be concluded that it covers not the largest number of causes of the primary concept, which are relatively weakly connected (the network is hierarchical), but repeating the same causes among different "virtual experts" confirms their accuracy, so they can be good "raw material" for further analytical processing.

The second network obtained by initially decomposing the problem and subsequently identifying the causes of each partial problem is more extensive with the same number of references to ChatGPT as in the first case. This provides a wide field for research; however, at this point, repetition of concepts in such a network is minimal, and therefore its adequacy needs to be verified by expert review.

Finally, The third network that represents a logical combination of both previous networks is most complete within our considered domain. At this

point, there are significant numbers of nodes participating in forming more than once i.e., those most important to our subject area.

These nodes can be considered sources for further development.

A special interface was developed to visualize these generated networks directly transferring formatted responses from ChatGPT itself. The graphical interface developed using AT&T's GraphViz allowed us to create graphs where nodes and edges corresponded hyperlinks to queries made through web search engines.

The proposed approach was applied to the cyber security vulnerability domain but could also apply across many other areas such as financial crises or military conflicts.

Limitations: Despite significant gains in resources (both time-wise and human-wise), it should be noted that both processes building cause-effect relationships between various phenomena as well as interpreting results require involvement from experienced data scientists familiar with studied domains ensuring the accuracy and reliability of results.

Building networks on your texts

In the previous paragraphs, we considered examples in which entities or entity pairs were extracted directly from a trained model of an LLM system, in particular, ChatGPT. This approach is quite universal, especially if you use several such systems to extract entities at the same time, however, there are a number of limitations, namely:

1. the actualization of the LLM-type system model does not always occur in real time, so the actual entities may not fall into the scope of consideration, and, accordingly, into the non-inertial network model;

2. the user may be interested in entities from their documentation, their texts, which also (unlike the Bible, Torah) may not be covered when training the model;
3. moreover, the user may be categorically against the fact that his texts get on the Internet, at the same time, he needs to form, for example, a semantic network and LLM-type systems will completely help him in terms of functionality.

In these cases, it makes sense to embed your own documentation in prompts (of course, limited in length, the number of tokens, in accordance with the requirements of real systems such as LLM). If the length of your own document exceeds the limits, you can fragment it and use different fragments sequentially in prompts, combining the responses of the systems to form a common network.

To solve the problem indicated in the third paragraph, the user can be advised to install an LLM type system in his corporate network, train the model using his documentation and proceed to solve the problem in accordance with the above methods. Moreover, he can not train his system on the basis of his own documentation, but proceed to solve the problem, in accordance with the methodology below, referring to his own LLM type system installed in a closed corporate network.

Let us give an example of the formation of a semantic network based on a selection of relevant news messages on the topic of ChatCGP. An excerpt from this collection is below:

AI, big data, and the end of humanity

2023-07-30 16:35

By Steven L. Shields

It happened again this morning. I pressed my bank's smartphone app, and a

warning message popped up. "Keep your data safe by updating to the newest app." Bug fixes, hackers smarter than the programmers, and data breaches are reported all the time. I get similar messages when I go to my bank's website. I access the website a couple of times a month. Each time I log in, I get sent to the security page and am asked to download several "safety" measures. The programs I downloaded last time are no longer good enough for some reason.

...

The Korea Times 2023.07.30 11:56

https://www.koreatimes.co.kr/www/opinion/2023/07/796_355972.html

Politicians don't have to fear AI replacement, thanks to 'legacy,' need for 'discourse': expert

The UK House of Lords serves in a largely advisory capacity and records speeches regularly

By Peter Aitken | Fox News

A British peer in the House of Lords suggested artificial intelligence (AI) could easily replace its members in the near future. But one expert argued the desire for tradition and trust in the human element when making major decisions will likely delay AI adoption.

"One of my thoughts is that the British have a sense of legacy - it's a big thing for them." Alex Sharpe, principal of Sharpe Management Consulting LLC, told Fox News Digital. "They also give 'discourse' a whole new dimension. It's almost like political theater, so I can't see it going away."

...

FOX News 2023.07.30 09:42

<https://www.foxnews.com/world/politicians-dont-have-fear-ai-replacement-thanks-legacy-need-discourse-expert>

Google should be used as platform to cross-check information from Bard AI: Google UK Executive

The FeedLast Updated: Jul 30, 2023, 12:25 AM IST

Synopsis In a recent interview with BBC, Google UK Executive advised that users should always cross-check information with its Google search engine and not solely rely on Bard.

Google countered OpenAI's chatbot ChatGPT with the release of Bard, and the recent report by Netskope also suggests that Google's AI chatbot has been attracting new users with ease lately. The report highlights that Google's Bard has been adding users at a rate of 7.1 per cent a week, whereas rival ChatGPT struggles at 1.6 percent. This makes Google Bard the first fastest-growing AI app in current times.

...

Economic Times 2023.07.29 22:40

<https://economictimes.indiatimes.com/news/international/us/google-should-be-used-as-platform-to-cross-check-information-from-bard-ai-google-uk-executive/articleshow/102238689.cms>

'Chatbot AI nurses' could solve NHS staff shortage, says developer

US start-up has created technology which can ensure hospital wards are 'super-staffed' with both human and AI expertise

By Joe Pinkstone, Science Correspondent 29 July 2023 0 5:57pm

Chatbot nurses capable of providing care for patients and passing medical exams could be the answer to alleviate NHS staff shortages, according to the company behind its technology.

Hippocratic AI, a US-based start-up, says a unique and bespoke algorithm used to monitor and tend to patients can be created by focusing on healthcare-specific data.

...

Telegraph 2023.07.29 20:53

<https://www.telegraph.co.uk/news/2023/07/29/chatbot-artificial-intelligence-nurses-nhs-staff-shortage/>

Researchers find multiple ways to bypass AI chatbot safety rules

by Nick Robertson - 07/29/23 11:55 AM ET

FILE - Text from the ChatGPT page of the OpenAI website is seen in New York, Feb. 2, 2023. Shares of the education technology company Chegg lost half their value Tuesday, May 2, after its CEO warned that OpenAI's free ChatGPT service was cutting into its growth. (AP Photo/Richard Drew, File)

Preventing artificial intelligence chatbots from creating harmful content may be more difficult than initially believed, according to new research from Carnegie Mellon University which reveals new methods to bypass safety protocols.

...

The Hill 2023.07.29 19:54

<https://thehill.com/policy/technology/4126647-researchers-find-multiple-ways-to-bypass-ai-chatbot-safety-rules>

ChatGPT works as relationship consultant

ChatGPT works as relationship consultant

29 July, 2023 06:23 pm

(Web Desk) - OpenAI's ChatGPT-4 has achieved widespread recognition since its launch as a strong multimodal large language model. Its versatility has captivated users worldwide, who have found applications for it in the study, social media captioning, and even note writing. However, one recent incident on Reddit has generated intrigue and debate among the online community.

...

Dunya News 2023.07.29 16:40

<http://dunyaNews.tv/index.php/en/Technology/743367-ChatGPT-works-as-relationship-consultant->

Google's New AI Tech Lets You Command Robots to Throw Away Your Trash

Google is looking to help people use natural language to control robots, with

technology that's like the systems powering Bard and ChatGPT.

Imad Khan July 28, 2023 5:34 p.m. PT

Google is helping robots accomplish tasks quickly and more efficiently, using technology like the systems powering AI chatbots Bard, ChatGPT, the Claude 2 and others, the company said in a blog post Friday.

Google's Robotics Transformer 2, or RT-2, is a "first-of-its-kind vision- language-action (VLA) model," Vincent Vanhoucke, the head of robotics for Google DeepMind, said in the post. Similar to the large language models behind AI chatbots, it trains based on text (and image) data found on the web, to "directly output robotic actions."

...

CNET News 2023.07.29 04:38

<https://www.cnet.com/tech/googles-new-ai-tech-lets-you-command-robots-to-throw-away-your-trash/>

To build a network of concepts on the subject of these news messages, a set of prompts of the same type was created, which were consistently worked out. Below is an example of such a prompt for the first news post:

→ Output 20 pairs of the most related entities from the text.

Each entity should be described in no more than 3 words. Format

of the output: 'entity 1; entity 2'. Each pair is displayed on a new

line. Text:

AI, big data, and the end of humanity. It happened again this morning. I pressed my bank's smartphone app, and a warning message popped up. "Keep your data safe by updating to the newest app." Bug fixes, hackers smarter than the programmers, and data breaches are reported all the time. I get similar messages when I go to my bank's website. I access the website a couple of times a month. Each time I log in, I get sent to the security page and am asked to download several "safety" measures. The programs I downloaded last time are no longer good enough for some reason. ...

Response received:

AI; big data
Bank's smartphone app; warning message
Bug fixes; programmers
Data breaches; hackers
Bank's website; security page
...

By prompt to activate the second virtual expert

→ Give me other important 20 pairs of concepts from this text

the answer has been extended expanded:

Computer technology; productivity
Privacy; data control
Personal Information Protection Commission; fine
Big data companies; profits
Facebook; user data
...

As a result of carrying out a similar procedure for all messages in the collection, an array of pairs of messages was obtained, a fragment of which is given below:

AI, big data, and the end of humanity

AI; big data
Bank's smartphone app; warning message
Bug fixes; programmers
Data breaches; hackers
Bank's website; security page
Computer technology; productivity
Privacy; data control
Personal Information Protection Commission; OpenAI
ChatGPT; personal information exposure
Big data companies; personal information collection
Facebook; advertisements
United States senator; Facebook's revenue model
HAVE; potential shots
Australia; welfare recipients
Errant algorithm; demands for repayment
Magnetic strip; credit card

Bonus point cards; customer information
Business consultants; generative AI revenue strategies
Electronic disruption; irreparable damage
Programmers; hackers
Computer technology; productivity
Privacy; data control
Personal Information Protection Commission; fine
Big data companies; profits
Facebook; user data
AI; potential dangers
Welfare recipients; erroneous demands
Bonus point cards; customer information
Business consultants; generative AI adoption
Electronic disruption; irreversible consequences
Programmers; hacker threats
Website update; security improvement
Scams; AI facilitation
Identity theft; protection measures
Electronic disruption; life disruption
Data update; vulnerability reduction
Hackers; exploiting vulnerabilities
Bank app; data safety
Generative AI; revenue opportunities
Electronic disruption; irreparable destruction

Politicians don't have to fear AI replacement, thanks to 'legacy,' need for 'discourse': expert

Politicians; AI replacement
British peer; AI adoption
House of Lords; advisory capacity
Richard Denison; AI speech
Job market; You have impact
ChatGPT; large language model
Alan Turing; artificial intelligence
Turing's "imitation game"; human-like behaviors
Significant data; AI training
Politicians' speeches; AI analysis
AI; creating new ideas
House of Lords; potential obsolescence
Legal requirement; AI adoption delay
Country's constitution; AI adoption
Politicians; constituents' election
Human elements; impactful news
Charles Colville; AI threat to journalism
Fox News Digital; AI expert

Chatbot; AI-generated speech
Lobbyist; convincing a machine
Expert; AI adoption
Tradition; trust in humans
House of Lords; elected body
AI prophecy; style of speech
Peer bots; deeper knowledge
You have impact; job market
Language models; generic speeches
Threat AI poses; journalism
AI's existence; white collar
Movies; AI perception
Alan Turing; artificial intelligence
Human-like behaviors; Turing's "imitation game"
Data availability; AI training
Politicians; documented history
AI replication; perfection
House of Lords; advisory body
Legal requirements; AI adoption
Lobbyist; convincing a machine
Lobbyists; negative perception
Politicians; elected by constituents

'Chatbot AI nurses' could solve NHS staff shortage, says developer

Chatbot nurses; NHS staff shortage
Hippocratic AI; healthcare-specific data
Munjal Shah; company founder
UK and US; staffing shortage
Chatbot-based system; specific roles
Chronic care nurse; pre-operative nurse
Genetic counsellor; high IQ
Large language model; professional tests
Chatbot pharmacist; medication assessment
Chatbot geriatrician; elderly care
Super staffing; nursing shortage
Voice-controlled version; real-time interaction
Virtual consultations; increase in check-ups
Speech recognition; speech synthesis
Seniors; phone communication
Voice renaissance; natural communication
Healthcare experts; AI algorithm
Hollywood actors; engaging model
Medicine plan adherence; convincing factor
Technology; alleviating loneliness
Healthcare technology; NHS staff shortage

AI nurses; medical exams
Hippocratic AI; healthcare delivery
Generative AI; healthcare potential
Large language model; professional certifications
Chatbot pharmacist; medication assessment
Chatbot geriatrician; elderly care
Voice-controlled version; real-time interaction
Virtual consultations; increase in check-ups
Speech recognition; speech synthesis
Phone communication; seniors' preference
Voice renaissance; natural communication style
Hollywood actors; engaging AI model
Medicine plan adherence; persuasive technology
Technology; loneliness alleviation
healthcare experts; AI fine tuning
Text-based technology; future voice interface
Natural and effective communication; AI benefits
Medicine plan adherence; patient compliance
Technology adoption; healthcare accessibility

After accessing the CSV2Graph service (section 3.4, Fig. 23), we obtain a visualization of the network of concepts, a fragment of which is shown in Fig. 24. As you can see, the main concepts, the most significant network nodes corresponding to the news selection under study, a kind of informational portrait of this array are: AI, OPENAI'S CHATGPT, AI CHATBOTS, ELECTRONIC DISRUPTION, HOUSE OF LORDS, AI ADOPTION, POLITICIANS, HUMAN INTERACTIONS CONTEMPLATION, HACKERS, PROGRAMMERS, LARGE LANGUAGE MODEL, AI SYSTEM CHALLENGES, BARD'S LIMITATIONS, MEDICINE PLAN ADHERENCE.

Building domain models based on your own texts using LLM (Language Model) technologies, in particular GPT and BERT, is possible both manually, using a simple prompt setting. This approach allows models to be adapted to specific tasks and contexts. However, such a process is limited by text sizes and time, as systems have a limit on the number of tokens, which can become an obstacle when working with long or complex texts.

To ensure the continuous use and automation of procedures in analytics, APIs (Application Programming Interface) provided by these systems become an important tool. APIs allow you to integrate LLM models into existing workflows, create custom applications, and automate text processing processes.

Currently, you can get an API key from OpenAI free of charge to work with the ChatGPT-3.5 model through the web interface (Fig. 25).

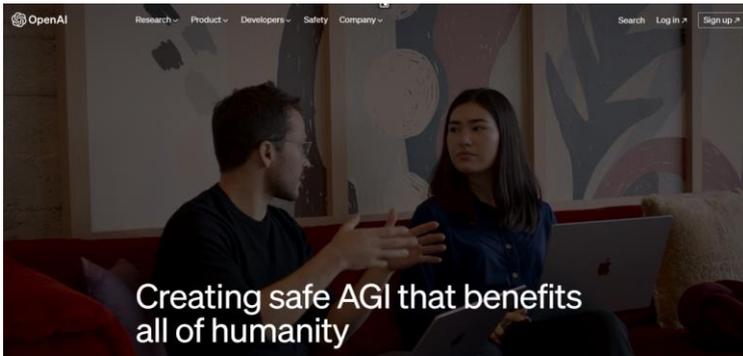


Figure 25. Main OpenAI site page

To obtain the key, it is enough to perform two procedures:

Register:

1. Respectfully “Sign up” at ChatGPT or login via Google Account
2. Verify Email
3. Confirm “Name Surname”
4. Link phone number
5. Verify OTP phone number

Expose key:

1. Search for the main OpenAI site page
2. Then Login with created credentials
3. And follow API page
4. From the Personal button account open “View API keys”

5. And create a new secret key

After that, in your programming system, you need to install the appropriate modules from OpenAI and you can get to work. To install the official Python bindings, run the following command:

pip install openai

After that, you can access the program (remembering to indicate “import openai” in the first lines), specifying the key value in certain variables and passing it when accessing the GPT model.

LangChain



<https://www.langchain.com>

Langchain technology also offers additional opportunities in the field of language processing. LangChain adds data accounting capabilities to LLM, allowing applications to connect to external data sources and interact in an interactive, context-aware manner (<https://docs.langchain.com/docs/>).

This integration gives developers the ability to create specialized applications based on language models that can provide targeted and fine-grained responses, bridging the gap between LLM and domain-specific requirements.

As we delve deeper into the realm of artificial intelligence, we encounter a myriad of tools and technologies designed to simplify our interaction with complex AI systems. One such revolutionary tool is Langchain, a powerful library that enables the rapid development of AI applications. This chapter aims to explore the diverse applications of Langchain and how they are transforming the AI landscape.

Langchain, at its core, is a language model library that facilitates the building of AI applications. It provides a robust platform for integrating AI capabilities into various sectors, ranging from e-commerce to customer service, and even personal use. The beauty of Langchain lies in its flexibility and ease of use, making it an ideal tool for both seasoned developers and AI enthusiasts alike.

One of the most popular applications of Langchain is in the creation of AI chatbots. These chatbots, powered by Langchain's large language models, can be fed with a wealth of information from various sources such as websites, PDF files, and other documents. The chatbots can then provide relevant responses to user queries, effectively serving as a 24/7 customer service representative. This application has proven particularly useful in e-commerce, where businesses can provide instant responses to customer inquiries, thereby improving customer satisfaction and engagement.

Another intriguing application of Langchain is in the field of data analysis. With the advent of big data, businesses are often overwhelmed with the sheer volume of data they have to process. Langchain, with its ability to process and analyze large volumes of text data, can help businesses make sense of their data, uncovering valuable insights that can drive strategic decision-making.

Moreover, Langchain's open-source nature invites developers worldwide to contribute to its growth. Developers can build upon Langchain's existing capabilities, creating custom applications tailored to specific needs. This collaborative approach fosters innovation and ensures that Langchain continues to evolve in line with the changing demands of the AI industry.

In conclusion, the applications of Langchain are as diverse as they are transformative. From powering AI chatbots to analyzing big data, and

aiding in content creation, Langchain is truly a game-changer in the AI landscape. As we continue to explore the potential of AI, tools like Langchain will undoubtedly play a pivotal role in shaping the future of this exciting field. The next chapter will delve into the technical aspects of working with Langchain, providing a practical guide for those keen on harnessing its power.

Let's delve into a practical example of Langchain's application. This example involves integration with Hugging Face, another powerful AI tool. Hugging Face is a platform that hosts over 200,000 AI models, making it an invaluable resource for AI developers. It is used by tech giants like Google, Amazon, Microsoft, and Meta, and is valued at over 2 billion dollars.

One of the applications we'll explore is an AI app that turns an image into an audio story. This involves three key components: an image-to-text model, a large language model, and a text-to-speech model. The image-to-text model interprets the scenario depicted in the image. The large language model then generates a short story based on this scenario, and finally, the text-to-speech model turns this story into an audio file.

To implement this, we first use Hugging Face to find an appropriate image-to-text model. Once we have our model, we use Langchain to generate a prompt for the large language model (GPT-3.5 Turbo in this case). The prompt is based on the scenario interpreted from the image. The large language model then generates a short story based on this prompt.

Next, we use a text-to-speech model from Hugging Face to turn the generated story into an audio file. The result is an AI app that can take an image, interpret it, generate a story based on the interpretation, and then narrate this story as an audio file.

To make this AI app user-friendly, we use Streamlit, a Python library that allows us to create a user interface for our app. The user can upload an

image, and the app will display the interpreted scenario, the generated story, and the audio file.

This example illustrates the power and versatility of Langchain when integrated with other AI tools like Hugging Face and Streamlit. It shows how Langchain can be used to build complex AI applications with relative ease.

In addition to the applications discussed, Langchain's open-source nature invites developers worldwide to contribute to its growth. Developers can build upon Langchain's existing capabilities, creating custom applications tailored to specific needs. This collaborative approach fosters innovation and ensures that Langchain continues to evolve in line with the changing demands of the AI industry.

In conclusion, the applications of Langchain are as diverse as they are transformative. From powering AI chatbots to analyzing big data, aiding in content creation, and even creating AI apps that turn images into audio stories, Langchain is truly a game-changer in the AI landscape. As we continue to explore the potential of AI, tools like Langchain will undoubtedly play a pivotal role in shaping the future of this exciting field.

The use of LLM technologies for building domain models becomes more flexible and convenient thanks to the API, and the development of Langchain technology expands the possibilities of working with your own texts, making them more efficient in analytical tasks.

Conclusions

The book "GPT Semantic Networking: A Dream of the Semantic Web – The Time is Now" is a description of the application of modern natural language processing technologies and the creation of semantic networks using ChatGPT. Information is presented on various aspects of NLP and the application of network models for text and context analysis. In conclusion, we present some conclusions that can be drawn from the book.

1. About LLM Technologies (GPT):
 - The importance of language models based on transformers, such as GPT, in the field of natural language processing is shown.
 - ChatGPT has great potential for solving various NLP tasks, including machine translation, sentiment analysis, text generation, and others.
2. On the possibility of applying the concepts of Complex Networks, Semantic Networks, and Semantic Web to a new level:
 - Modern network approaches provide theoretical foundations and powerful tools for presenting and analyzing data and textual information.
 - The possibilities of building and analyzing link structures in Semantic Networks are demonstrated, which opens up new opportunities at a new stage of data mining.
3. Network analysis and visualization tools:
 - Visualization is an important tool for representing complex network structures and revealing patterns and relationships.
 - Analytical tools help you extract actionable insights from these networks to help you make better decisions.
4. New approach to building Semantic Networks:

- The combination of advanced NLP technologies, in particular ChatGPT, with the Semantic Network concept represents an innovative approach to building Semantic Networks.
- Semantic Networks, created using this approach, have a high level of intelligence and provide ample opportunities for analyzing textual information, taking into account the context.

The approaches proposed in the book can be successfully applied in various areas, including decision support systems and others.

5. For Decision Support Systems:

- Semantic networks created with ChatGPT and the Semantic Network concept can serve as a powerful tool for decision support systems.
- With the ability to parse textual information in context and understand complex relationships between data, these networks will help you make informed and informed decisions in a variety of situations.

Thus, the book "GPT Semantic Networking: A Dream of the Semantic Web – The Time is Now" not only proposes an innovative approach to the creation of semantic networks but also indicates the possibilities of applying this approach to decision support systems and many other areas.

We are confident that readers will be able to take advantage of the new knowledge and insights to make a significant contribution to the development of the science and technology of natural language processing and building intelligent semantic models. We hope that they will find in this book not only useful information but also inspiration for further research and application of advanced artificial intelligence technologies in their

work. We hope that this book will be a starting point for someone to innovate in the field of natural language processing and semantic networks.

Sincerely,

Dmytro Lande,
Leonard Strashnoy,
August 11, 2023



Dmytro Lande – Professor, Doctor of Science, Head of Department of Information Security Ihor Sikorsky National Technical University of Ukraine. Winner of the Victor Glushkov Prize in Computer Science, and the "Yaroslav Mudry" Prize in Law.

ORCID: 0000-0003-3945-1178



Leonard Strashnoy – Solution Architect for Computer Science Corp, Amazon Inc, and Tata Consulting Services. Certified Amazon AWS Data Architect, Google Certified Digital Cloud Leader.

ORCID: 0009-0008-5575-0286

Literature

[Albert, 2002] Albert R., Barabasi A.-L. Statistical mechanics of complex networks // *Reviews of Modern Physics*, 2002. – 74. – P. 47

[Ashish Vaswani, 2017] [Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. Edited by: I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett. ISBN: 9781510860964

[Aymen El Amri, 2023] Aymen El Amri. The art and science of developing intelligent apps with OpenAI GPT-3, DALL·E 2, CLIP, and Whisper - Suitable for learners of all levels / Kindle Edition, 2023. – 378 p.

[Barham, P., 2022] Barham, P., Chowdhery, A., Dean, J., Ghemawat, S., Hand, S., Hurt, D. & Wu, Y. Pathways: Asynchronous distributed dataflow for ml. *Proceedings of Machine Learning and Systems*, 4, 430-449 (2022)

[Bernard J. Jansen, 2023] Bernard J. Jansen, Soon-gyo Jung, Joni Salminen. Employing large language models in survey research. *Natural Language Processing Journal*. Volume 4, September 2023, 100020. DOI: <https://doi.org/10.1016/j.nlp.2023.100020>

[Berners-Lee, 2001] Berners-Lee, Tim; Hendler, James; Lassila, Ora (May 17, 2001). "The Semantic Web". *Scientific American*. Vol. 284, N. 5. – pp. 34–43

[Brin, 1988] Brin, S.; Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30: 107–117

[Brown Tom, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901 (2020)

[Caldeira, 2005] Caldeira S.M.G., Petit Lobao T.C., Andrade R.F.S., Neme A., Miranda J.G.V. The network of concepts in written texts // Preprint ArXiv. physics/0508066 (2005)

- [Ce Zhou, 2023] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangzhou Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. arXiv preprint arXiv:2302.09419, 2023
- [Cherven , 2015] Ken Cherven. Mastering Gephi Network Visualization. Packt Publishing, 2015. – 378 p.
- [Devlin Jacob, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018
- [Domingos, 2009] Poon, Hoifung, and Pedro Domingos. Unsupervised semantic parsing. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Vol. 1. Association for Computational Linguistics, 2009
- [Dorogovtsev, 2001] Dorogovtsev S.N., Mendes J. F. F. Language as an evolving word net // Proc. R. Soc. Lond. B 268, 2603 (2001)
- [Ferrer-i-Cancho, 2001] Ferrer-i-Cancho, R., Sole R.V. The small world of human language // Proc. R. Soc. Lond. B 268, 2261 (2001)
- [Ferrer-i-Cancho, 2004] Ferrer-i-Cancho R., Sole R.V., Kohler R. Patterns in syntactic dependency networks // hys. Rev. E 69, 051915 (2004)
- [Ferrer-i-Cancho, 2005] Ferrer-i-Cancho, R. The variation of Zipf's law in human language. // Phys. Rev. E 70, 056135 (2005)
- [Golovach, 2007] Yu. Golovach, V. Palchykov. Mykyta fox and the language network. Zh. Fiz. Dosl. 11, No. 1, 22 (2007) (in Ukrainian)
- [Gunning, 2017] Gunning D. Explainable Artificial Intelligence (Xai). Darpa-Baa-16-53. Nd Web 2 (2), 1 (2017)
- [Jingfeng Yang, 2023] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, Xia Hu. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond arXiv preprint arXiv: 2304. 13712, 2023
- [Kleinberg, 1999] Kleinberg, Jon (1999). "Authoritative sources in a hyperlinked environment". Journal of the ACM. 46 (5): 604–632
- [Lambert, 2010] Lambert M. Surhone, Mariam T. Tennoe, Susan F. Henssonow. Graphviz. VDM Publishing, 2010. – 108 p.

[Lande1, 2023] Lande, Dmitry and Strashnoy, Leonard, Concept Networking Methods Based on ChatGPT & Gephi (April 17, 2023). SSRN. Available at: <http://dx.doi.org/10.2139/ssrn.4420452>

[Lande2, 2023] Lande, Dmitry and Strashnoy, Leonard. Hierarchical Formation of Causal Networks Based on ChatGPT (May 8, 2023). SSRN. Available at: <http://dx.doi.org/10.2139/ssrn.4440629>

[Lande3, 2023] Lande, Dmitry and Strashnoy, Leonard. Causality Network Formation with ChatGPT (May 30, 2023). SSRN. Available at: <http://dx.doi.org/10.2139/ssrn.4464477>

[Lehmann, 1992] Lehmann, Fritz; Rodin, Ervin Y., eds. (1992). Semantic networks in artificial intelligence. International series in modern applied mathematics and computer science. Vol. 24. Oxford; New York: Pergamon Press. p. 6. ISBN 978-0080420127. OCLC 26391254

[Liu Yinhan, 2019] Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:19 07.11692, 2019

[McGuinness, 2004] D.L. McGuinness, F. Van Harmelen. OWL web ontology language overview. W3C recommendation, 10 February 2004. URL: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>

[Mnih, 2016] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., “Asynchronous methods for deep reinforcement learning”. In: International Conference on Machine Learning, 2016. – pp. 1928–1937

[Motter, 2002] Motter A.E., de Moura A.P.S., Lai Y.-C., Dasgupta P. Topology of the conceptual network of language // Phys. Rev. Fr. E 65, 2002. – 065102(R)

[Raffel Colin, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020

[Segev, 2022] Segev, Elad (2022). Semantic Network Analysis in Social Sciences. London: Routledge. ISBN 9780367636524

[Sigman, 2002] Sigman M., Cecchi G.A. Global Properties of the Wordnet Lexicon. Proc. Natl. Acad. Sci. USA, 99, 1742 (2002)

[Silver,2017] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. "Mastering the game of go with deep neural networks and tree search". *Nature*, 529 (2016), p. 484

[Solat, 2023] Solat J. Sheikh, Sajjad Haider, Alexander H. Levis, On semi-automated extraction of causal networks from raw text, *Engineering Applications of Artificial Intelligence*, Volume 123, Part A, 2023, 106189, <https://doi.org/10.1016/j.engappai.2023.106189>

[Sowa, 1987] John F. Sowa (1987). "Semantic Networks". In Stuart C Shapiro (ed.). *Encyclopedia of Artificial Intelligence*. – Wiley, 1987

[Stephen Wolfram, 2023] Stephen Wolfram. *What Is ChatGPT Doing ... and Why Does It Work?* Wolfram Media, Inc., March 9, 2023. – 112 p. ISBN-13: 978-1-57955-081-3

[Strogatz, 2001] Strogatz S.H. *Exploring Complex Networks // Nature*, 2001. – 410. – pp. 268-276

[Sussna, 1993] Sussna, Michael. "Word sense disambiguation for free-text indexing using a massive semantic network" *Proceedings of the second international conference on Information and knowledge management*. ACM, 1993.

[Teven Le Scao, 2022] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. *Bloom: A 176b-parameter open-access multilingual language model*. arXiv preprint arXiv:2211.05100, 2022

[Watts D., 1998] Watts D.J., Strogatz S.H. *Collective dynamics of "small-world" networks*. *Nature*, 1998. – Vol. 393. – pp. 440-442

[Wayne Xin Zhao, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. *A survey of large language models*. arXiv preprint arXiv:2303.18223, 2023

[Wolfram, 2002] St. Wolfram. "A New Kind of Science". – Wolfram Media, Inc. May 14, 2002. 1264 p.

[Zhou, 2004] Shi Zhou; R.J. Mondragon. *The rich-club phenomenon in the Internet topology*. *IEEE Communications Letters* (Volume: 8, Issue: 3, March 2004). – p. 180-182. DOI: 10.1109/LCOMM.2004.823426

[Ferreira, 2017] Diogo R. Ferreira. A Primer on Process Mining: Practical Skills with Python and Graphviz. Springer Briefs in Information Systems. – Springer International Publishing. – 2017. 101 p. ISBN 978-3-319-56426-5, 978-3-319-56427-2

Scientific publication

Dmytro LANDE,
Leonard STRASHNOY

**GPT SEMANTIC NETWORKING:
A DREAM OF THE SEMANTIC WEB –
THE TIME IS NOW**

Monograph

In the author's edition

Підписано до друку 14.08.2023. Формат 60x84/16. Папір офс.
Гарнітура Times. Спосіб друку – ризографія. Ум. друк. арк.
13,3. Обл.-вид. арк. 5,7. Наклад 100 пр. Зам. № 15-200.

ТОВ "Інжиніринг"
ISBN 978-966-2344-94-3