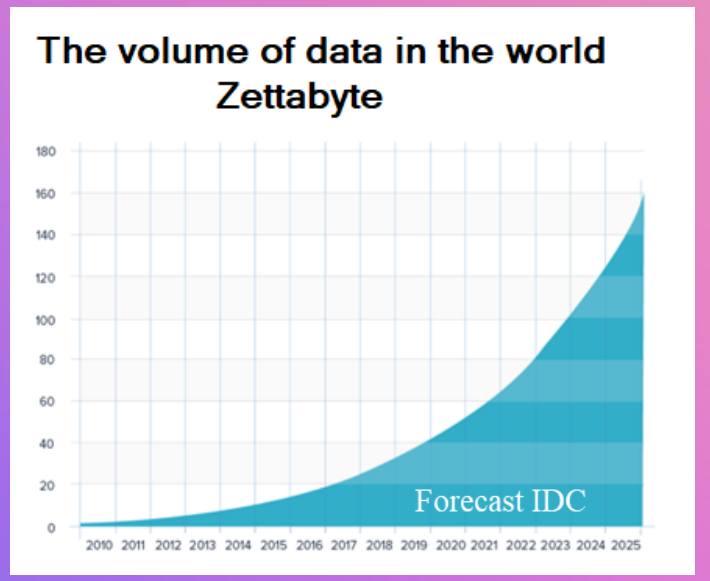
Third International Conference on Artificial Intelligence for Sustainable Development,
Jordan-Amman, November 19-20, 2025

COMPARATIVE ANALYSIS OF MODELS AND METHODS OF SEMANTIC SEARCH

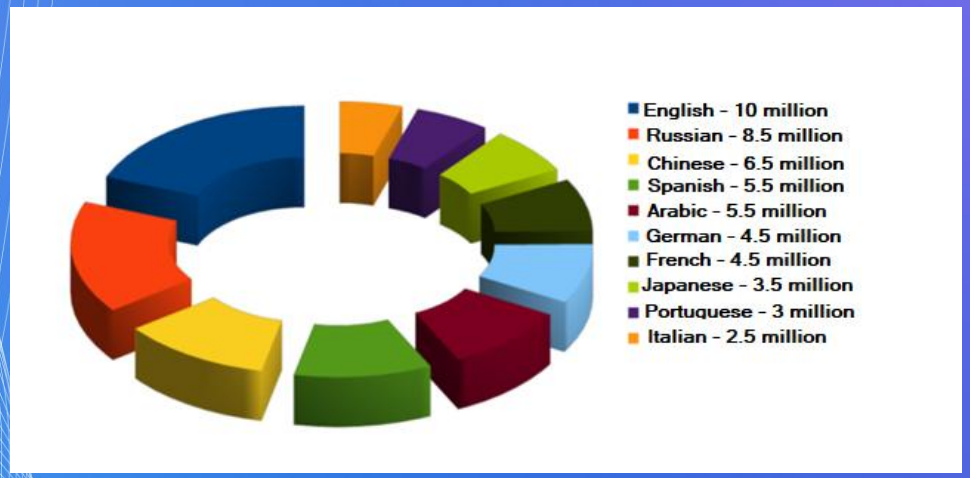
V. I. Khabirov (*didager@yandex.ru*)
E. Yu. Sazonova (*rassadnikova_ekaterina@mail.ru*)
O. N. Smetanina (*smoljushka@mail.ru*)
G. R. Shakhmametova (*shakhgouzel@mail.ru*)
Ufa University of Science and Technology, Ufa

Relevance

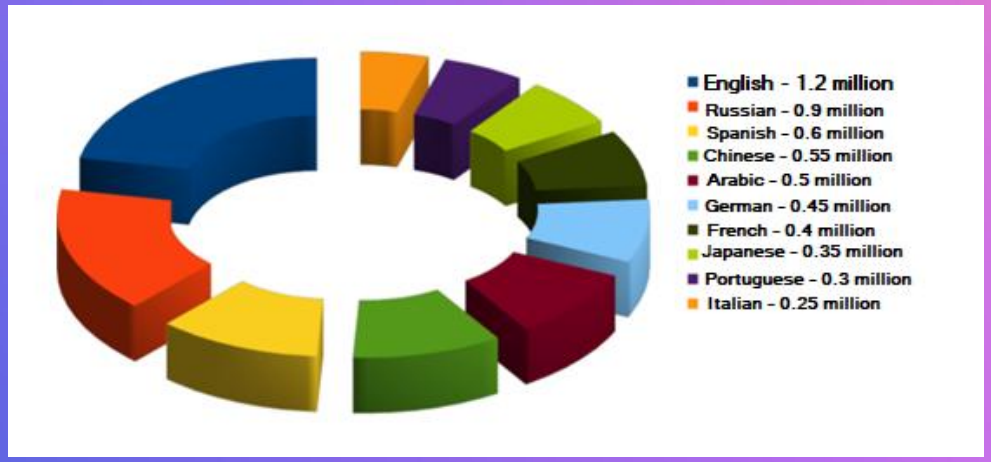
- ✓ Annual data growth
- ✓ Challenge of semantic search among growing volumes of data
- ✓ Continuous development of semantic search technologies



Statistics of search engine requests in various languages



Popularity of languages in pages viewed by users

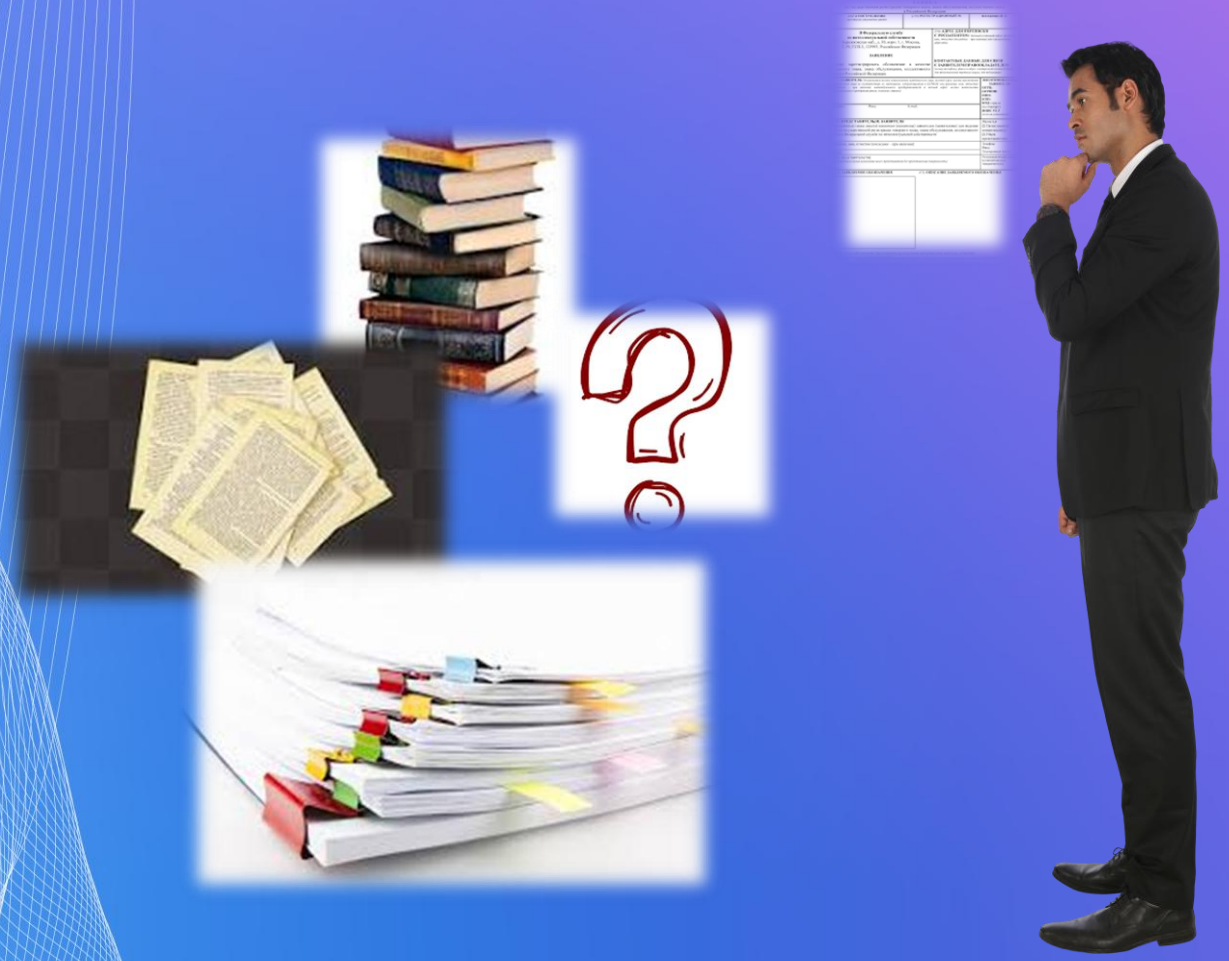


https://catalog-777.com/blog/2023-10/yaziki-interneta-statistika-i-populyarnost.html

Plan

- ✓ Semantic search task for small and large documents
- ✓ Development of semantic search technologies. History
- ✓ Classification of methodologies for semantic retrieval of small documents
- ✓ Classification of methodologies for semantic search of large documents (≈300 pp.)
- ✓ The most common metrics applied to different tasks, technologies, and development stages
- ✓ Statement of the semantic search problem
- ✓ Modern methods of semantic search. Vector methods
- ✓ Modern methods of semantic search. Language models
- ✓ Approach, experimentation and comparison of results of selected methods for small documents
- ✓ Approach to implementing semantic search for large documents
- ✓ Conducting experiment and interpreting results

Semantic search task for small and large documents



An expert in a certain subject area needs to analyze documents currently available in various sources that are semantically close to a certain application.

Selection of methodology
tools metrics

Development of semantic search technologies. History

Period	Technologies / tasks
2000 - 2005	- Attempts to implement NLP technologies to improve the quality of search (*Yandex, Google , IBM*), in search engines taking into account synonyms and using morphological analysis. - Idea: Structured data and early prototyping of the Semantic Web (*T. Berners -Lie , Ja . Hendler , E. Miller and D . Brickley , W3C* and others). - Search engines (taking into account basic factors of user behavior: ranking models based on clicks and interaction with results) (*Google, Microsoft, Yahoo*).
2006 – 2010	- The first version of Google Universal Search (traditional web queries + maps, images, news and video) - the beginning of a comprehensive approach to search. - Personalization models (browsing history + user preferences) (*Amazon, Netflix, Yahoo and Google*). - Latent Concept Dirichlet Allocation (LDA) and topic modeling for document classification (*D. Blandford*).
2011 – 2015	- Launch of the Google Knowledge project Graph (providing direct factual information on popular questions in search results). - Formation of the market of virtual assistants (Siri , Google Now , Amazon Alexa), voice search. - Semantic database projects such as DBpedia and Wikidata . - Deep learning neural networks in the analysis and understanding of the natural language, analysis of relationships between words - Word2Vec (*Google , OpenAI , etc.*)
2016 – 2020	- Widespread use of deep learning neural networks to solve problems of text search and analysis. Standards in query processing and context understanding include ELMo , ULMFiT , and BERT models, among others (*Google).* - Improvements to voice search and conversational agents. Accurate interpretation of spoken commands and questions. - Development of recommender systems (behavioral activity + social interactions).
2021 - 2025	- Use of LLMs in commercial products, a wide range of applications on OpenAI GPT, Microsoft Turing NLG and other major LLMs technologies . - Semantic search - full-fledged conclusions and generalized knowledge in response to user requests. - Application of hybrid approaches (classical algorithms and deep learning), machine understanding of complex sentence structures and placement of accents in speech). - Study of multimodal search (integration of visual, auditory and textual data simultaneously).

Classification of methodologies for semantic retrieval of small documents

Methodology class	Methods	Advantages	Disadvantages
Classical text methods	***Bag-of-Words (BoW)*** : A simple method that represents a document as a set of words without regard to order or grammatical relationships. ***TF-IDF*** : A frequency-based approach that assigns importance to words based on their frequency in a document and collection. ***BM25*** : A ranking model that takes document length and term frequency into account.	Simplicity, high performance	They ignore semantics and internal dependencies between words.
Lexical-semantic methods	***WordNet*** : A graphical dictionary used to establish relationships between words. ***Dependency Parsing*** : Grammatical analysis that studies syntactic relationships between words. ***POS Tagging*** : Automatic tagging of parts of speech for better understanding of sentence structure.	They take into account the structure of the text and the relationships between words	They are resource intensive and require additional marking.
Topic modeling	***Latent Dirichlet Allocation (LDA)*** : A generative model designed to discover abstract topics in a text collection. ***Non- negative Matrix Factorization (NMF)*** : A matrix factorization method useful for extracting the main topic of a document	Identification of hidden connections and general characteristics of the text	Limited by case size and consume significant computing resources
Vector spaces and embeddings	***Word2Vec , FastText*** : Algorithms for transforming words into n-dimensional vectors that preserve semantic properties. ***Doc2Vec*** : An extensive extension of Word2Vec that creates embeddings of entire documents. ***BERT , RoBERTa*** : Transformer architectures capable of capturing contextual features of text.	Ability to understand subtle nuances of text and make accurate recommendations	High resource requirements and need for prior training
Hybrid methods	***Hybrid Retrieval Models*** : Combine classical retrieval (BoW , TF-IDF) with modern methods (BERT, Doc2Vec). ***Neural Network Enhanced Approaches*** : Neural networks on top of traditional models to increase performance.	Improving accuracy and reliability by combining the strengths of different methods.	Complexity of implementation and high support costs.

Classification of methodologies for semantic search of large documents (≈300 p.)

Methodology class	Methods	Features
Indexing and search engine analysis technologies	***Inverse indexing*** : Creates index files containing lists of the locations of all words in a document. ***Combination of indexing and filters*** : Implementation of combined methods that include a pre-filter to reduce search volume. ***Index compression*** : Compresses index files to save memory and reduce loading times.	Systems that support efficient mechanisms for indexing large volumes of data.
Topic Modeling Concepts	***Latent Dirichlet Allocation (LDA)*** : a topic modeling method based on Bayesian statistics. ***Probabilistic Latent Semantic Analysis (PLSA)*** : a probabilistic approach to identifying latent topics. ***Hierarchical Topic Modeling*** : Building hierarchical topic structures to make data easier to find and interpret.	The methods are useful for organizing and analyzing large text collections, allowing one to effectively identify the general theme and direction of a document.
Graph analysis and network structures	***Analysis of Document Structure*** : examination of the document structure (headings, tables, formulas) to highlight the most important parts. ***Graph Representation*** : a representation of a document as a graph, where nodes correspond to text elements and edges correspond to relationships between them. ***Network Analysis*** : Applying graph theory methods to study the relationships between document components.	The methods are effective in identifying key information and eliminating noise that occurs in large documents.
Fuzzy search and skip search	***Fuzzy String Matching*** : Fuzzy string search methods that allow for minor deviations in the spelling of words. ***Skip-Gram Models*** : Models that skip words in sentences to preserve the integrity of the search. ***Approximate Nearest Neighbor Search*** : Fast nearest neighbor search in high-dimensional spaces.	The methods improve search accuracy in situations where precise searching is impossible due to differences in word forms or errors.
Artificial intelligence and machine learning	***Machine Learning for Classification*** : Using classifiers to predict the relevance of a document to a query. ***Natural Language Processing (NLP)*** : The application of natural language processing techniques to understand the content of a document. ***Deep Learning with Neural Networks*** : Using deep neural networks to extract complex information from large documents.	The methods provide a powerful tool for analyzing and interpreting large documents, increasing the depth of text understanding.

The most common metrics applied to different tasks, technologies and development stages

Challenge / Technology	Metrics
Classification and clustering	Precision, Recall, F-measure, Silhouette Coefficient , ROC-AUC , etc.
Analysis of meanings and connections	Precision, Recall, F1-score , Event Detection F1-score , Accuracy, Link Accuracy, Coverage, Completeness, Consistency , Entity resolution accuracy, Semantic overlap, Classification-based measures, etc.
QA systems	Exact Match Accuracy, BLEU score, ROUGE-N , etc.
Context-sensitive search	**MRR** , **Recall ,** DCG, Click Through Rate , **Precision ,** F1-SCORE , Mean Average Precision , Normalized Discounted Cumulative Gain , MSE , Cross Entropy Loss , Perplexity , etc.
Extracting key concepts	Precision, Recall , Entity-level F1 score, Micro-F1 and Macro-F1 , Slot Error Rate , Overlap Ratio (OR) , Redundancy Metric (RM) , Average Overlap Degree (AOD) , Keyword Density (KD) , Corpus-specific Metrics , etc.
User interfaces	Task Success Rate, User Satisfaction Surveys , etc.

MRR, Recall and Precision
complement each other and allow to fully evaluate the quality of search

Statement of semantic search problem

Given:

$M = \{ m_1, m_2, \ldots, m_k \}$ – set of semantic search methods

$Q = \{ q_1, q_2, \ldots, q_k \}$ – set of queries

$D = \{ d_1, d_2, \ldots, d_k \}$ – set of text documents

$R(q_i) \subseteq D$ – set of documents relevant to the query q_i, specified manually or according to the "gold standard".

Need to find:

Method that produces a set of ordered relevant documents.

$$m_k(q_i) \rightarrow \{d_{i1}, d_{i2}, \ldots, d_{in}\},$$

where $d_{ij} \in D$

The goal of the experiment is to find a method $m_k^* \in M$ that provides the best values of the selected quality metrics for the set of all queries from Q.

$$m^* = \arg\max_{m_k \in M} L(m_k)$$

where:

$L(m_k)$ is the aggregated quality function of the method m_k, calculated on the basis of metrics.

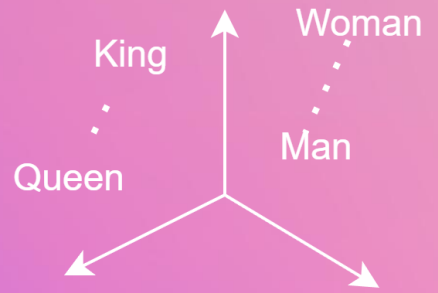
The final choice of method can be based on an aggregate metric that takes into account the trade-off between recall and ranking - Recoll and MRR . The Precision metric is also used .

$$Q(m_k) = \alpha \cdot Recall(m_k) + \beta \cdot MRR(m_k),$$

where:

α , β – weighting coefficients depending on the system priorities

Modern methods of semantic search. Vector methods



Suggested by: Word2Vec + Faiss

Word2Vec - family of algorithms (T. Mikolov , Google) for creating vector representations of words in natural language processing.

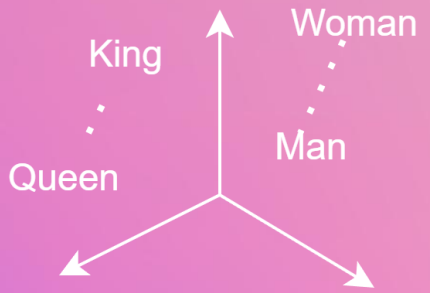
Two main approaches:

- **CBOW (Continuous Bag-of-Words -** predicts the central word based on the words surrounding it.
- **Skip-gram** - predicts surrounding words based on a central word.

Given: a sentence of length T : $w_1, w_2, \ldots, w_T$, where each w_t is a separate word.
We need to create a function that will predict either the central word w_t (if we use CBOW) or the surrounding words $w_{t-c}, \ldots, w_{t+c}$, where c is the width of the context window (if we use Skip-gram).
Network architecture : Single-layer neural network with an input layer, a hidden layer, and an output layer

CBOW: Sequence of words $w_1, w_2, \ldots, w_T$, c is the context window size.
The problem is reduced to reconstructing the central word w_t from its context $w_{t-c}, \ldots, w_{t+c}$.
Model: $h = f(W^{(1)}x + b^{(1)})$
where: $f(\cdot)$ is a nonlinear activation (usually softmax), $W^{(1)}$ is the input layer weights, $b^{(1)}$ is the bias, x is the input vector (represents a concatenation of one - hot vectors of context words).
We calculate the output layer: $y = W^{(2)}h + b^{(2)}$
The goal is to minimize the cross-entropy error: $E = -\log p(w_t \mid x) = -\sum y_t(i)\log(y'(i))$
where y' - output probability distribution (softmax), and $y_t(i)$ is the true distribution (one-hot vector).
Optimization - gradient descent method with updating weights $W^{(1)}$, $W^{(2)}$ and displacements $b^{(1)}$, $b^{(2)}$.

Modern methods of semantic search. Vector methods



Suggested by: Word2Vec + Faiss

The main goal of FAISS is to quickly find vectors closest to a given candidate vector:
1. Euclidean Norm and Cosine Distance
There are two main distance measures commonly used:

Euclidean norm (L2-distance) : $\|\vec{a}-\vec{b}\|^2=\sum(\vec{a}_i-\vec{b}_i)^2$, where $i=1, n$
Cosine distance : $\cos(\vec{a},\vec{b})=(\vec{a}\cdot\vec{b})/(\|\vec{a}\|\|\vec{b}\|)$

Measures determine how close two vectors are to each other.

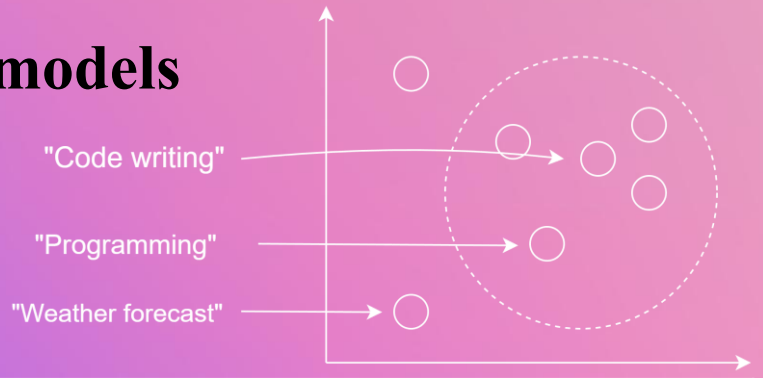
2. Data indexing
FAISS supports various indexing methods, including flat indexes (IndexFlat), hierarchical indexes (HNSW, PQ), and others.
Indexes allow you to reduce search time by sacrificing accuracy, or maintain full accuracy by incurring only a minor slowdown.
IndexFlat : Stores all vectors in RAM and performs exhaustive search.
Product Quantization (PQ) : Quantizes the vector space, reducing memory and speeding up search.
Hierarchical Navigable Small World graphs (HNSW) : Builds a graph, reducing search time to sublogarithmic.

Modern methods of semantic search. Language models

Language methods: BERT

MiniLM-L6-v2 - a miniature version of BERT

The main components of the BERT architecture



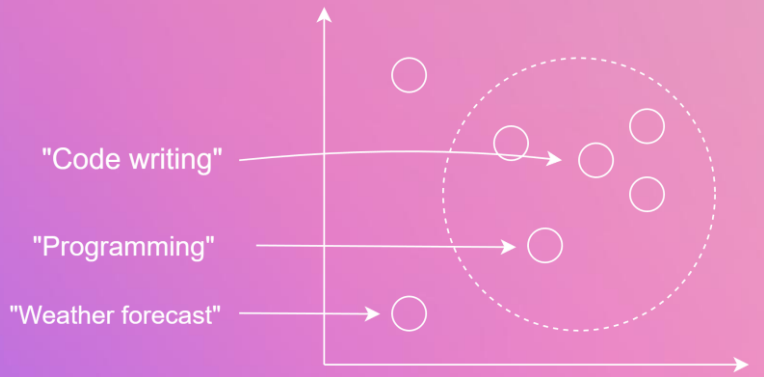
Component	Component characteristics
Encoder Stack	The core of BERT is a stack of encoders , each of which is a separate Transformer layer . The number of layers varies from model to model (for example, *BERT- base* has 12 layers, *BERT- large* has 24 layers).
Self- attention Mechanism	Within each layer, a self-attention mechanism is used - it allows the model to pay attention to different parts of the incoming text, giving more weight to important sections.
Feedforward Neural Network	The component of each layer is f *eedforward* Neural network - performs nonlinear transformations of input data.
Positional Encoding	Each token is assigned a special *positional encoding*), which allows taking into account the word order.
Token Types	BERT takes two sentences as input and uses a special mask (*token types*) to indicate the first and second sentences, which helps maintain context.
Special Tokens	There are special markers at the input: [CLS] (*classification token*), which is inserted at the beginning of a sentence and is used to output the final class, and [SEP], which separates the first sentence from the second.
Masked Language Model (MLM)	During training, half of the words are randomly masked with a special marker [MASK]. The model's task is to predict the original words based on context—a predictive approach.
Next Sentence Prediction (NSP)	BERT is trained to predict whether a second sentence is a natural continuation of the first. This task adds a second source of training signal, strengthening the model's understanding of text structure.

Modern methods of semantic search. Language models

Language methods

MiniLM-L6-v2 - a miniature version of BERT

- a model optimized for text-related tasks representations and semantic comparisons, based on the BERT architecture, underwent distillation training.



1. Argumentation of architecture

The basic structure of MiniLM-L6-v2:
Encoder layers : 6 layers (L6).
Hidden state size : 384.
Number of attention elements : 12.

2. Distilled learning

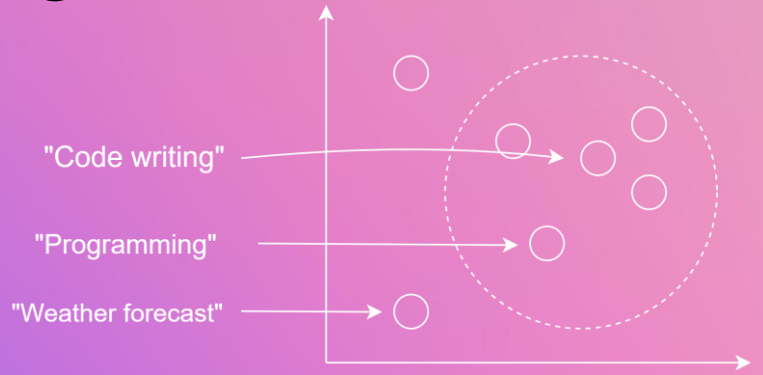
The distillation process involves the transfer of knowledge from a large model (teacher model , BERT) to a smaller (student model , MiniLM-L6-v2). The goal is to convey to the student (MiniLM-L6-v2) all the information sufficient for high-quality completion of tasks.

3. Loss of function and learning

Losses consist of two parts:
Distillation loss : Learns to approximate the teacher's output (logits).
Regularization loss : Minimizing the difference between the student's and teacher's outputs.
The general loss function looks like this: $L = L_{distill} + \lambda L_{reg,}$ where:
$L_{distill}$ - loss of distillation.
L_{reg} - regularization loss.
λ is a hyperparameter that regulates the balance between losses.

Modern methods of semantic search. Language models

Language methods
MiniLM-L12-v2 - a miniature version of BERT



Main differences:

1. **Encoder layers** :
   - **MiniLM-L12-v2** : 12 layers, which is more than the MiniLM-L6-v2.
   - **MiniLM-L6-v2** : Contains only 6 layers.

2. **Performance** :
   - **MiniLM-L12-v2** : A more complex model, potentially more accurate on tasks requiring deep text analysis.
   - **MiniLM-L6-v2** : Compact model, faster and requires less computing power, but may be inferior in accuracy in some complex tasks.

3. **Application** :
   - **MiniLM-L12-v2 :** Suitable for tasks where maximum accuracy is required, especially in situations where the data contains complex or noisy text.
   - **MiniLM-L6-v2 :** Best used in applications where speed and resource savings are more important than maximum accuracy.

Approach, experimentation and comparison of results of selected methods for small documents

Metrics for evaluating search methods

Completeness - $Recall(q_i, m_k) = \frac{|R(q_i) \cap m_k(q_i)|}{|R(q_i)|}$

Average inverse rank - $MRR(m_k) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{i,k}}$

where :
$Q = \{ q_1, q_2, \dots q_n \}$ is set of queries
$R(q_i)$ – set of documents relevant to the query q_i
$M = \{ m_1, m_2, \dots, m_n \}$ – set of search methods
$rank_{i,k}$ – position of first relevant document to the method m_k and query q_i

Precision = True Positives /(True Positives+False Positives)

True Positives (TP) - number of correctly classified positive examples.

False Positives (FP) - number of falsely classified positive examples.

Approach, experimentation and comparison of results of selected methods for small documents

MS MARCO is a large multilingual dataset developed

Microsoft to evaluate search methods and machine reading

Examples of request and document formats

Requests
define extreme
what does chattel mean on credit history
what was the great leap forward brainly
tattoo fixers how much does it cost
what is decentralization process.
sanitizer temperature
what is a bank transit number

Documents
The presence of communication amid scientific minds was equally important to the success of the Manhattan Project as scientific intelligence was. The only cloud hanging over …
The most common cause for liver transplantation in adults is cirrhosis caused by various types of liver …
Xylem transports water and soluble mineral nutrients from roots to various parts of the plant. It is responsible for replacing water lost through transpiration and photosynthesis. Phloem translocates sugars made by photosynthetic areas of plants to storage organs like roots, tubers or bulbs. aetiology.combination …
If the lower set of ribs on the right side of the rib cage get damaged due to an injury, then one is likely to …

Approach, experimentation and comparison of results of selected methods for small documents

Example of implementation in Russian

Precision value is acceptable for the task being solved.

Experimental results

	Recall @10	MRR @10	Recall@100	MRR@100	Average search execution time
Language models					
MiniLM - L 12- v 2	0.7925	0.6873	0.9492	0.5817	0.0249 sec
MiniLM-L6-v2	0.7636	0.6694	0.9369	0.5521	0.0231 sec
Vector methods					
Word2Vec	0.0 91 6	0.3 5 57	0.1662	0.2946	0.1375 sec
Word2Vec + FAISS	0.1005	0.5331	0.1898	0.3011	0.0014 sec

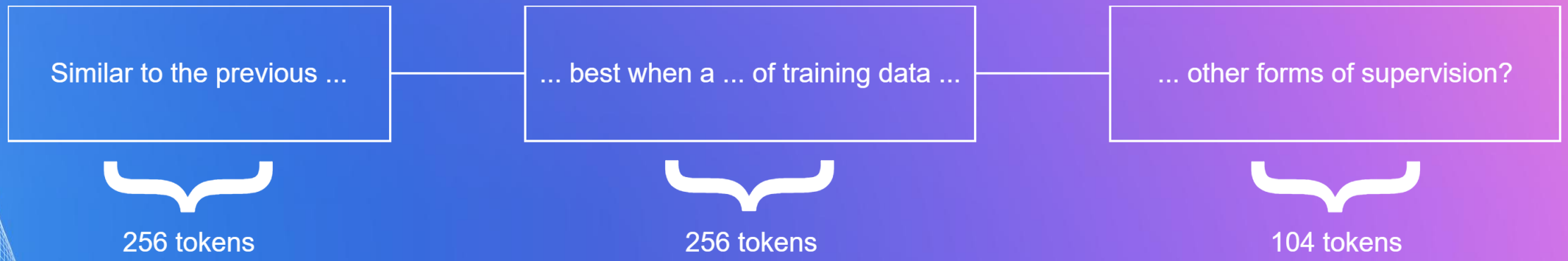
Approach to implementing semantic search for large documents

Problem :

MiniLM family models cannot handle documents of length > 256 tokens.

Solution :

1. Split documents *into overlapping* (to preserve contextual relationships) token segments
2. Calculate the mean vector for each set of segments



Conducting experiment and interpreting results

The experiment used a subset of MS-MARCO-Document-v2, which included 200,000 documents. To evaluate the quality of semantic search, 500 user queries were selected. Each The query is pre-labeled with 400 relevant documents.

	Recall @10	MRR @10	Recall@100	MRR@100	Precision@10	Precision@100	Average search execution time
MiniLM - L 12- v 2							
Without segmentation	0.0274	0.9453	0.1424	0.9453	0.8474	0.4609	0.0249 sec
With segmentation	0.0272	0.9518	0.1567	0.9518	0.8342	0.5029	0.0231 sec
MiniLM - L 6- v 2							
Without segmentation	0.0260	0.9715	0.1329	0.9715	0.8158	0.4296	0.0187 sec
With segmentation	0.0265	0.9251	0.1547	0.9278	0.8171	0.4958	0.0214 sec

Conducting experiment and interpreting results

- ✓ **High MRR@10 (>0.92) -** Indicates that the first relevant document is returned very early, often in the first position. This indicates high ranking accuracy at the initial stage of search.
- ✓ **Low Recall@10 (<0.03) -** Means that less than 3% of all relevant documents (out of 400) are among the top 10 results. Even with a good ranking, the system only covers a small portion of the full relevant set.
- ✓ **Significant increase in Recall@100 (up to ~15%) -** Shows that significantly more relevant documents make it into the top 100 than into the top 10. This is typical for tasks where relevant information is widely distributed and requires deeper scanning.
- ✓ **High Precision@10 (>0.8) -** Indicates high reliability of the initial results: out of every 10 returned documents, more than 8 are relevant. The system effectively filters out noise at the top positions.
- ✓ **Precision@100 (~0.43–0.50) -** Means that about half of the documents in the top 100 are relevant. This is an acceptable level of "pollution" for systems where the user views many results or reranking is used .
- ✓ **MRR@10 ≈ MRR@100 (almost identical) -** Indicates that the first relevant document typically appears within the first 10 positions, and further expansion of the list to 100 does not improve the position of the first hit. The ranking is stable.

Why the MiniLM-L12-v2 is better for long documents?

- ✓ More layers - deeper understanding of semantics, even in fragments.
- ✓ Best embeddings chunks - they encode the meaning more accurately, despite the cropping.
- ✓ Fragmentation resistance - does not lose relevance when aggregated.
- ✓ Stable ranking - maintains quality regardless of text processing method.

Conclusion

- ✓ The stated task of semantic search of documents (small and large) corresponding to a certain request (application), and the analysis of models and methods in conducting semantic search for small and large documents showed that it is necessary to use different approaches.

- ✓ An analysis of the metrics used for evaluation revealed the need to tailor them to the task at hand. Metrics that consider recall, accuracy, and ranking were selected – Recall , Precision and MRR . An aggregate metric that takes into account the tradeoff between recall and ranking was also proposed – Recall and MRR .

- ✓ When organizing semantic search for small documents, it was proposed to use vector methods (Word2Vec) , as well as a combination of vector methods with data indexing (Word2Vec + Faiss) , and language models based on the BERT language model, in particular: MiniLM-L6-v2 and MiniLM-L12-v2 .

- ✓ When organizing semantic search for large documents, it was proposed to use their partitioning into intersecting sequences of fixed length and the MiniLM-L6-v2 and MiniLM-L12-v2 language models .

- ✓ The obtained experimental results and their interpretation allow us to choose an approach depending on the needs of the task: speed or accuracy and completeness.

Thank you
for your attention!