BG Pulse Journal USA

ISSN:1947-2536 http://www.bgpulseusa.com/index.php/bg

SEMANTIC MATCHING METHODS AND THEIR APPLICATION IN INTELLIGENT INFORMATION SYSTEMS

Boboqulov J.J., Sharipova D.J., Saidrasulov Sh.N.

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi,

Tashkent, Uzbekistan

Doi: https://doi.org/10.5281/zenodo.24015216

Abstract. This article examines semantic matching — the estimation of correspondence between textual objects by meaning rather than by surface lexical form — which has become the computational core of modern intelligent information systems. Four generations of methods are distinguished and analysed: lexical and statistical methods (TF-IDF, BM25, LSA), knowledge-based ontological methods (WordNet, knowledge graphs), distributional word vectors (Word2Vec, GloVe, FastText) and contextual neural models (BERT, Sentence-BERT, dense retrievers). For each generation the mathematical basis, the computational cost and the limits of applicability are stated. The cross-encoder, bi-encoder and late-interaction architectures are compared on accuracy, query latency and index size, and a six-stage reference model of retrieval is constructed in which the cost of each successive stage rises while the number of documents it examines falls. Approximate nearest-neighbour indexing and hybrid sparse–dense fusion are described, evaluation corpora and ranking metrics are systematised, and the application of the methods to semantic search, retrieval-augmented generation, recommendation, classification and data integration is set out. A separate section addresses the morphologically rich and low-resource case of the Uzbek language.

Keywords: semantic matching, semantic similarity, embedding, dense retrieval, BERT, Sentence-BERT, ontology, knowledge graph, intelligent information systems, retrieval-augmented generation, approximate nearest neighbour search, low-resource languages.

Annotatsiya. Ushbu maqolada semantik moslikni aniqlash — matniy obyektlar o‘rtasidagi mos kelishni ularning sirtqi leksik shakli bo‘yicha emas, balki ma’nosi bo‘yicha baholash masalasi ko‘rib chiqiladi. Usullar to‘rt avlodga ajratib tahlil qilinadi: leksik-statistik usullar (TF-IDF, BM25, LSA), bilimga asoslangan ontologik usullar (WordNet, bilimlar grafi), distributiv so‘z vektorlari (Word2Vec, GloVe, FastText) va kontekstual neyron modellar (BERT, Sentence-BERT, zich qidiruv modellari). Kross-enkoder, bi-enkoder va kech o‘zaro ta’sir arxitekturalari aniqlik, so‘rov kechikishi va indeks hajmi bo‘yicha qiyoslanadi; qidiruvning olti bosqichli etalon modeli quriladi. Taqribiy eng yaqin qo‘shni indekslash va gibrid qidiruv yoritiladi, baholash korpuslari va metrikalari tizimlashtiriladi, usullarning semantik qidiruv, RAG, tavsiya tizimlari va ma’lumotlar integratsiyasidagi qo‘llanilishi bayon etiladi. Alohida bo‘limda o‘zbek tili kabi morfologik boy va kam resursli tillar uchun yuzaga keladigan muammolar muhokama qilinadi.

Kalit so‘zlar: semantik moslik, semantik o‘xshashlik, vektorli ifodalash, zich qidiruv, BERT, Sentence-BERT, ontologiya, bilimlar grafi, intellektual axborot tizimlari, kengaytirilgan generatsiya, taqribiy eng yaqin qo‘shni izlash, kam resursli tillar.

Introduction. The fundamental defect of classical information retrieval is that it compares words as sequences of characters. A user who asks for the price of an automobile is not shown a document that speaks of the cost of a car, although the two expressions denote the same thing. This phenomenon is known in the literature as the vocabulary mismatch problem, and it arises from synonymy, polysemy, paronymy and morphological variation. Its magnitude is not marginal: in the classical study of Furnas and colleagues two people selected the same term for the same familiar object in fewer than one case in five.

The problem is aggravated in morphologically rich languages. In Uzbek a single stem generates hundreds of surface forms through the successive concatenation of affixes, so that the probability of a purely lexical coincidence between a query and a relevant document falls further still. A retrieval system that matches strings therefore fails precisely where the user is least able to guess the wording of the collection.

Semantic matching is the body of methods that estimate, as a numerical quantity, the degree of correspondence between two or more information objects — a word, a sentence, a document, a query, a database schema or an element of an ontology — on the basis of their meaning. It constitutes the computational core of semantic search engines, question-answering systems, recommender systems, dialogue agents and corporate knowledge bases.

The past decade has seen a change of kind rather than of degree. Approaches founded on rules and dictionaries have been displaced by learned vector representations produced by deep neural networks. Accuracy has risen substantially, but new difficulties have appeared: demand for computational resources, the opacity of the resulting models, and the poor support of languages for which large corpora do not exist.

The contribution of this article is threefold. First, the methods are classified into four generations and compared on a single set of criteria — mathematical basis, computational cost, treatment of synonymy and interpretability. Second, a six-stage reference model of semantic retrieval is constructed in which the three architectural families are assigned to the stages for which their cost profile is appropriate. Third, the specific obstacles presented by morphologically rich low-resource languages are stated, and a practical order of work is proposed for the Uzbek case.

Literature review and methodology. The vector space model of Salton and colleagues established the representation of a document as a weighted vector over the vocabulary and remains the foundation of sparse retrieval. The probabilistic framework was consolidated by Robertson and Zaragoza in the BM25 ranking function, which is still the standard baseline. The first attempt to pass beyond the lexical barrier was latent semantic analysis, which applied singular value decomposition to the term–document matrix.

The distributional turn began with Word2Vec, which produced dense word vectors from corpus statistics alone; GloVe added global co-occurrence information and FastText added character n-grams, which is of direct consequence for agglutinative languages. The Transformer architecture of Vaswani and colleagues, and the BERT model built upon it, made the vector of a word dependent on its context and thereby dissolved the problem of polysemy that static vectors

could not address. Reimers and Gurevych then showed, in Sentence-BERT, how a model of that family can be restructured so that retrieval over large collections becomes tractable.

Two gaps are apparent in the literature. The first is that the generations of methods are usually presented in isolation and compared on accuracy alone, without the latency, index size and resource requirements that in fact determine the choice in a deployed system. The second is that the treatment of low-resource and morphologically rich languages is confined to remarks on multilingual model coverage, without a stated order of practical work.

The methodology of this article is analytical and comparative. A formal statement of the matching problem is given; the four generations are examined in sequence against a common set of criteria; the architectures are compared on measured operational characteristics reported in the primary literature; and a reference model is constructed in which the results are assembled. Benchmark figures are cited from the MTEB and BEIR evaluations as published at the time of writing.

Formal statement of the problem. Let X and Y be two sets of textual objects, for instance queries and documents. The task is to construct a function

$$s : X \times Y \to [0, 1], \quad (x, y) \mapsto s(x, y),$$

in which s(x, y) expresses the degree of semantic proximity of x and y. In practice the function is realised in two steps: the objects are mapped into a d-dimensional vector space and a distance is then computed between the images,

$$s(x, y) = sim(\varphi(x), \psi(y)), \quad \varphi, \psi : X \cup Y \to \mathbb{R}^d.$$

The measure of proximity in almost universal use is the cosine of the angle between the vectors,

$$cos(u, v) = (u \cdot v) / (\|u\| \cdot \|v\|).$$

Its property of depending on direction and not on magnitude makes it appropriate to the comparison of texts of unequal length. Where $\varphi \neq \psi$ the task is asymmetric and separate encoders are trained for the two sides; where the two sides are of the same nature a single encoder suffices.

A distinction must be drawn that is frequently elided. Semantic similarity denotes membership of the same conceptual class, as between automobile and car. Semantic relatedness denotes any associative connection, as between automobile and petrol, which are not similar but strongly related. Distributional models measure relatedness more faithfully than similarity; ontological measures separate the two. The distinction is operative in design: a recommender requires relatedness, whereas duplicate detection requires similarity.

Table 1. Classes of the semantic matching problem

Class	First object	Second object	Distinguishing property
Query–document	Short query, 2–8 words	Long document	Asymmetric; large disparity of length
Text–text (STS)	Sentence	Sentence	Symmetric; comparable lengths

Question–answer	Question	Answer passage	Low lexical overlap
Schema–schema	Table or column name	Table or column name	Very little context available
Ontology alignment	Concept node	Concept node	Graph structure is informative
Entity linking	Named entity in text	Knowledge-base entity	Requires disambiguation of homonyms

Lexical and statistical methods. The first generation treats a text as a bag of words and does not model meaning at all. It nevertheless remains an essential component of hybrid systems, because it is computationally cheap, its output is fully explainable, and it locates proper names, product codes and rare technical terms with a reliability that neural models do not attain.

At the lowest level the comparison is of character sequences. The Levenshtein distance gives the minimum number of elementary edits required to transform one string into another, and the Jaccard coefficient measures the ratio of the intersection to the union of two sets of words or n-grams. These measures handle misspelling, transliteration variants and abbreviation, but they cannot detect synonymy at all: for automobile and car the Jaccard coefficient is zero.

In the vector space model each document is a sparse vector over the vocabulary whose components are weighted by the TF-IDF rule,

$$w(t, d) = tf(t, d) \cdot \log (N / df(t)),$$

where tf(t, d) is the frequency of term t in document d, df(t) the number of documents containing t and N the size of the collection. The BM25 function refines this in two respects, by saturating the contribution of term frequency and by normalising for document length,

$$BM25(q, d) = \Sigma\, IDF(t) \cdot f(t,d)(k_1+1) / [\, f(t,d) + k_1(1 - b + b \cdot |d|/avgdl) \,],$$

with $k_1 \in [1.2, 2.0]$ and $b = 0.75$ in the usual parameterisation. BM25 requires no training and adapts immediately to a new domain, which is why it continues to serve as the permanent baseline against which neural retrievers are measured.

Latent semantic analysis was the first method to pass beyond the lexical barrier. It applies singular value decomposition to the term–document matrix and truncates the representation to k principal components, so that words occurring in the same contexts acquire neighbouring vectors. Its limitations are that the decomposition is costly, that the model is static and must be recomputed when documents are added, that the components admit no interpretation, and that word order is lost entirely. Topic models of the LDA family represent a document as a distribution over latent topics and are interpretable, but they require more text than a query or a short passage provides.

Knowledge-based methods. The second generation derives meaning not from text but from a curated structure of knowledge, so that semantic matching becomes a question of distance in a graph. WordNet groups words into synonym sets connected by hyponymy, meronymy and

antonymy. Knowledge graphs such as Wikidata, DBpedia and ConceptNet store factual as well as linguistic relations, which is what entity linking and factual question answering require.

Path-based measures compute distance in the hierarchy. The Wu–Palmer measure is the most widely used,

$$sim_WP(c_1, c_2) = 2 \cdot depth(LCS(c_1, c_2)) / [\, depth(c_1) + depth(c_2) \,],$$

where LCS denotes the least common subsumer of the two concepts. Measures based on information content add corpus statistics to the graph structure, the information content of a concept being IC(c) = −log P(c). The Resnik measure takes the information content of the common ancestor and the Lin measure normalises it,

$$sim_Lin(c_1, c_2) = 2 \cdot IC(LCS(c_1, c_2)) / [\, IC(c_1) + IC(c_2) \,].$$

In enterprise systems a related problem arises: the alignment of schemas drawn from different sources, where one database records a customer name under one identifier and another under a different one. Alignment combines evidence at three levels — the similarity of names and comments, the position of the node in the structure, and the distribution of actual values in the columns — fused by a weighted sum or a trained classifier. Symbolic graphs may also be embedded in a vector space: the TransE model interprets a relation as a translation, h + r ≈ t, and later models such as DistMult, ComplEx and RotatE extend the construction to symmetric and transitive relations.

The strength of this generation is that its output is fully explainable, since the path by which a match was established can be displayed; that it requires no large training corpus; and that it represents precise domain knowledge, as in medicine or law, with fidelity. Its limitations are that the construction and maintenance of an ontology are expensive, that out-of-vocabulary terms are not handled, and that high-quality ontologies do not exist for low-resource languages. In practice these methods therefore function not as an independent solution but as an additional signal that refines the output of a neural model.

Distributional word vectors. The theoretical basis of the third generation is the distributional hypothesis of Harris and Firth: words occurring in similar contexts have similar meanings. The hypothesis made it possible to extract meaning automatically from corpus statistics rather than from a dictionary.

The Word2Vec model represents each word by a dense vector of 100 to 300 dimensions and is trained in one of two configurations: the continuous bag-of-words, which predicts the central word from its context, and skip-gram, which predicts the context from the central word and yields better vectors for infrequent words. The skip-gram objective is

$$\mathcal{L} = (1/T) \sum_t \sum_{j \neq 0} \log p(w_{t+j} \mid w_t),$$

with negative sampling replacing the full softmax so that the cost ceases to scale with the size of the vocabulary. The arithmetic of the resulting space encodes semantic relations, in the well-known instance king − man + woman ≈ queen.

GloVe constructs the global co-occurrence matrix of the corpus and factorises it, thereby combining global statistics with the local window of Word2Vec. FastText represents a word as the

sum of its character n-grams, so that a vector can be produced for a form never seen in training. This property gives it a substantial advantage in morphologically rich languages such as Uzbek, Turkish and Finnish, and it is the reason the model remains relevant for the Uzbek case discussed below.

Text vectors are obtained from word vectors by averaging, by IDF-weighted averaging, by the SIF scheme of frequency-inverse weighting with removal of the first principal component, by training a document identifier as in Doc2Vec, or by solving an optimal transport problem as in Word Mover's Distance. The intrinsic limitation of the generation is polysemy: each word has exactly one vector, so that all senses of an ambiguous word are collapsed into a single point. In Uzbek the form yuz denotes both a hundred and a face, and a static model cannot separate them. It was this limitation that motivated contextual models.

Contextual neural models. The Transformer architecture produces the vector of a word dynamically from the whole sentence by means of multi-head self-attention,

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k})\, V.$$

Every token is connected directly to every other, so that long-range syntactic and semantic dependencies are preserved and the problem of polysemy is resolved.

BERT learns bidirectional context through the masked language modelling objective. For semantic matching it was first applied as a cross-encoder: query and document are concatenated into a single sequence and the model emits a single score. This yields the highest accuracy obtainable, because the interaction between the tokens of the two sides is modelled directly. It is, however, unusable for retrieval at scale: ranking a million documents for one query requires a million separate forward passes.

The bi-encoder architecture resolves the difficulty. Query and document are encoded independently and the match is the cosine of the angle between precomputed vectors; document vectors are written to the index once, and at query time a single encoding is performed. Sentence-BERT retrained BERT in a Siamese configuration for exactly this purpose. The effect reported by its authors is decisive: finding the most similar pair among ten thousand sentences takes approximately sixty-five hours with a cross-encoder and a few seconds with the bi-encoder, at nearly unchanged accuracy.

A third position lies between the two. ColBERT stores not one vector per document but one vector per token, and defers the interaction to a late stage through the MaxSim operator, $s(q, d) = \Sigma_i \max_j E(q_i) \cdot E(d_j)^T$. This attains accuracy close to that of the cross-encoder at a latency close to that of the bi-encoder, at the cost of an index an order of magnitude larger.

Table 2. Comparison of the three matching architectures

Criterion	Cross-encoder	Bi-encoder	Late interaction
Accuracy	Highest	Medium to high	High
Query latency, 10^6 documents	Hours	Milliseconds	Tens of milliseconds

Precomputation of the index	Impossible	Complete	Complete
Index size	None	One vector per document	One vector per token
Appropriate role	Re-ranking	First-stage retrieval	Single stage or re-ranking

The quality of a modern encoder depends as much on the training regime as on the architecture. The established method is contrastive learning, in which matching pairs are drawn together in the vector space and non-matching pairs pushed apart under the InfoNCE objective,

$$\mathcal{L} = -\log\left[\exp(\mathrm{sim}(q, d^{+})/\tau) / \left(\exp(\mathrm{sim}(q, d^{+})/\tau) + \Sigma_j \exp(\mathrm{sim}(q, d_j^{-})/\tau)\right)\right],$$

where τ is a temperature parameter. The decisive factor in quality is the selection of hard negatives: documents superficially similar to the query that do not in fact answer it. The principal models developed in stages — DPR introduced separate encoders for question answering, SimCSE used dropout as the augmentation in an unsupervised setting, and the E5, BGE and GTE families combined multi-stage training with multilingual coverage.

At the time of writing contrastive training remains the standard. The accepted basis of evaluation is the Massive Text Embedding Benchmark, which scores models across eight task types and reduces the result to a single average; more than five thousand submissions have been recorded and the leader changes almost weekly. Among open models Qwen3-Embedding-8B leads the multilingual and code tracks, reaching 70.58 points over 131 tasks on MTEB Multilingual v2 as of October 2026, while BGE-M3 supports more than one hundred languages and unites dense, sparse and multi-vector modes in a single model, which makes it a convenient choice for hybrid systems. Two cautions are practical. The retrieval section of the benchmark, and not the overall average, is the figure relevant to a search system; and in specialised domains such as biomedicine, financial filings or source code, a model adapted to the domain corpus almost always outperforms a larger general-purpose one.

Vector indexing and the reference model of retrieval. Beyond accuracy there is a second question of equal practical weight: how the nearest vectors are found among a million or a billion in a few milliseconds. Exhaustive search computes a distance against every vector, which is of order O(N·d) and inadmissible at scale. The solution is the family of approximate nearest-neighbour algorithms, which surrender a small fraction of accuracy for a speed-up of two orders of magnitude.

HNSW builds a hierarchical navigable small-world graph and descends from the upper layers, attaining high accuracy and speed at the cost of memory. IVF partitions the vectors into clusters and examines only the nearest, which is cheap and tunable but errs at cluster boundaries. Product quantisation compresses each vector into sub-vector codes and reduces memory by a factor of ten to fifty with a slight loss of accuracy; the combination IVF-PQ serves collections of billion scale. The implementations in use are FAISS, ScaNN and Annoy, and the vector databases Milvus, Qdrant, Weaviate and pgvector.

Dense and sparse retrieval are complementary: the dense side understands meaning, the sparse side locates exact terms, article numbers and proper names. The most robust method of combining their outputs is reciprocal rank fusion,

$$RRF(d) = \Sigma_r \, 1 / (k + rank_r(d)), \quad k = 60,$$

whose advantage is that it requires no normalisation of the scores produced by the different systems and operates on ranks alone.

These components assemble into the six-stage reference model shown in Figure 1. The principle of the model is that each successive stage is more expensive than the last while examining proportionally fewer documents, so that the cost of the accurate method is paid only on a short candidate list.

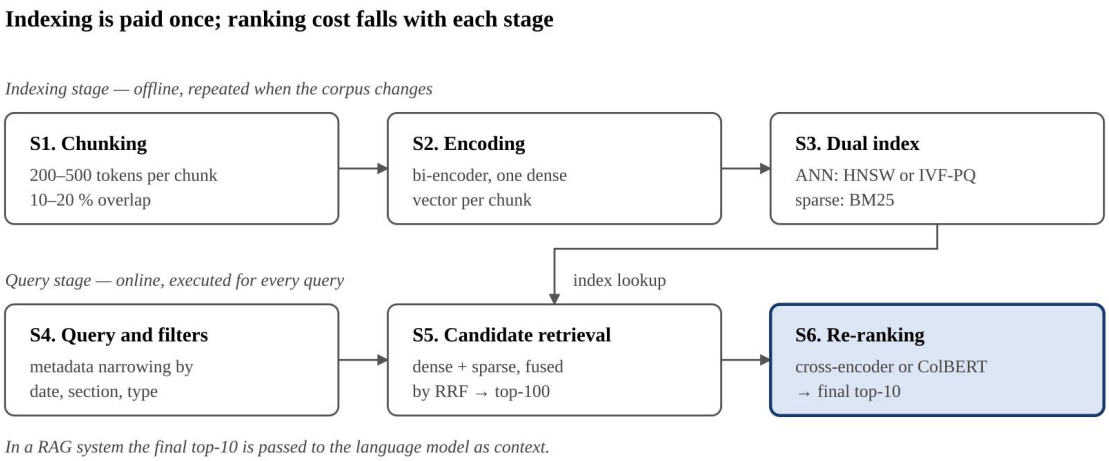


Figure 1. Reference model of semantic retrieval: the indexing stage is paid once, and the cost of ranking falls with each successive stage

The operative rule is that recall governs the first stage and precision the last. A relevant document that does not enter the candidate list cannot be recovered by any subsequent stage, however accurate; conversely, a re-ranker applied to a hundred candidates costs a hundred forward passes rather than a million, which is the whole economy of the design.

Evaluation methodology. Similarity tasks are scored by the correlation of the model output with human judgement, the Spearman rank correlation being preferred to the Pearson coefficient because order and not absolute value is at issue. The reference corpora are STS-Benchmark and SICK for sentence pairs, Quora Question Pairs for duplicate detection, MS MARCO and Natural Questions for retrieval, and BEIR for zero-shot generalisation across eighteen heterogeneous collections.

Retrieval quality is measured by ranking metrics. The mean reciprocal rank,

$$MRR = (1/|Q|) \, \Sigma_i \, 1 / rank_i,$$

reports the average inverse position of the first correct answer and is the principal figure for question answering. The normalised discounted cumulative gain,

$$DCG@k = \Sigma_i (2^{rel_i} - 1) / \log_2(i + 1), \quad nDCG@k = DCG@k / IDCG@k,$$

is the most widely reported metric, since it accounts for the position of a result and the degree of its relevance; nDCG@10 is the conventional operating point. Precision@k, recall@k and mean average precision complete the set.

BEIR deserves separate mention, because it tests a model on domains it was not trained on and therefore measures practical robustness rather than fit to a benchmark. Many models that substantially outperform BM25 on MS MARCO fall behind it on several BEIR collections, which is the empirical justification for retaining the sparse component in a deployed system.

Table 3. Comparative characteristics of the four generations of methods

Group of methods	Correlation on STS	nDCG@10 in retrieval	Latency	Resource requirement
TF-IDF and BM25	Low	Medium; a stable baseline	Very low	Minimal
LSA and LDA	Low to medium	Low	Low	Moderate
Ontological (WordNet)	Medium	Low	Medium	An ontology is required
Static vectors	Medium	Low to medium	Low	Small
Bi-encoder (SBERT)	High	High	Low	GPU advisable
Cross-encoder	Highest	Highest	Very high	GPU required
Hybrid with re-ranking	—	Highest	Medium	High

Note. The figures express the relative position of the groups rather than absolute values, which depend on the corpus and on the particular model; the retrieval column refers to the retrieval section of the benchmark and not to the overall average.

Four errors recur in evaluation practice. The first is reliance on a single average score, when the benchmark aggregates eight unlike tasks. The second is the omission of a trial on the proprietary corpus, where the leader of a public board frequently performs worse than a mid-ranking model. The third is the measurement of accuracy alone, when latency, index size and cost per million tokens weigh equally in the decision. The fourth is the omission of BM25 from the comparison, which conceals the true marginal benefit of the neural model.

Application in intelligent information systems. The most direct application is semantic search. Documents are divided into passages, each passage is encoded and written to a vector index, and at query time the user text is mapped into the same space so that the nearest passages are returned. A query for the procedure of terminating an employment contract then retrieves a document that speaks of the dismissal procedure, a result lexical retrieval would not produce at all.

Retrieval-augmented generation is at present the most widespread application. The process has three stages: the question is encoded and the most relevant passages are selected from the knowledge base; the passages are passed to a large language model as context; and the model composes its answer from that context. The construction mitigates the two principal defects of a language model used alone, the staleness of its knowledge and the fabrication of facts. The quality of semantic matching determines the quality of the whole system, since no correct answer can be generated from an incorrectly retrieved context.

The remaining applications follow the same pattern with different choices of architecture. Recommender systems use content vectors with an ANN index to find similar items and to mitigate the cold-start problem. Document classification and clustering operate on bi-encoder vectors, the latter with k-means or HDBSCAN. Plagiarism and near-duplicate detection, where paraphrase must be recognised, justify the cost of a cross-encoder or of ColBERT. Corporate knowledge bases over internal regulations and instructions require the hybrid scheme with domain adaptation, since they are dense in proper names and institutional terminology. Data integration combines ontological and vector evidence. Dialogue agents map an utterance onto an existing intent, and service desks route an enquiry to the correct department, both by means of a bi-encoder over a base of labelled examples.

Three decisions govern the outcome in deployment. Chunking exerts a stronger influence than is generally expected: passages that are too short lose context and passages that are too long dilute the vector, with 200 to 500 tokens and an overlap of 10 to 20 per cent giving good results, and chunk boundaries best aligned to paragraph or section boundaries. Metadata filtering applied before the vector search improves both latency and accuracy by narrowing the space. Domain adaptation of a general model on a thousand or more labelled pairs is usually cheaper and more effective than migration to a larger general model.

The case of the Uzbek language. The methods surveyed above were developed and validated on English corpora, and their application to Uzbek encounters a specific set of obstacles. Uzbek is agglutinative: successive affixation of a single stem generates hundreds of surface forms, as in kitob, kitobim, kitobimdagi, kitoblarimizdagilardan. The vocabulary expands sharply and the frequency of each individual form falls, which starves the training of static vectors.

Three further difficulties compound this. Texts exist in parallel in Latin and Cyrillic script, and the Latin orthography itself admits several renderings of the same character, so that without a normalisation stage one word is split across several tokens. Open Uzbek corpora are smaller than their English counterparts by orders of magnitude, and the labelled pairs that semantic matching requires — STS sets and query–document pairs — are almost entirely absent. The sub-word vocabularies of multilingual models are fitted to the major languages, so that Uzbek words fragment excessively, wasting the context window and degrading vector quality. To these is added code-switching, since Uzbek, Russian and English terms are mixed in technical and medical text.

The practical order of work follows from the diagnosis. A single normalisation layer, performing script conversion and unifying apostrophe variants, is the cheapest and most consequential step. Sub-word models of the FastText family, or lemmatisation, cover the forms absent from the vocabulary. The corpus deficit is addressed by beginning from a multilingual model — LaBSE, XLM-RoBERTa or BGE-M3 — as a zero-shot starting point rather than by

training from scratch. Labelled data can be bootstrapped by machine translation of the English STS and MS MARCO sets for use in adaptation, and a contrastive fine-tuning on as few as one to ten thousand domain pairs yields a substantial improvement. The hybrid scheme retains BM25, which handles proper names and terminology that the neural model mishandles.

Four directions of further work may be identified for the language. An open STS corpus of human-rated sentence pairs is the precondition of any objective comparison of models. Domain-adapted encoders are required for law, medicine and education. A national lexical semantic network of the WordNet type would supply the interpretable component. Cross-lingual matching, by which an Uzbek query retrieves an English or Russian document, would widen access to the scientific and technical literature.

Discussion. Three conclusions of practical consequence follow from the analysis.

The first is that no generation of methods has wholly displaced its predecessor. BM25 still outperforms neural models on proper names, codes and rare terms; ontological methods retain their place wherever the result must be explained. The dominant direction of practice is therefore combination rather than replacement, and the hybrid scheme with reciprocal rank fusion is the concrete form this takes.

The second is that the choice of architecture is a statement about the balance of accuracy and latency rather than about accuracy alone. At scale the only rational design is the multi-stage one: a cheap method separates the candidates and an expensive model re-ranks them. The reference model of Figure 1 makes the economy explicit.

The third is that position on a public leaderboard is not the operative criterion in model selection; the result on the proprietary corpus is. A mid-ranking model adapted to the domain generally outperforms an unadapted larger one, and the retrieval section of the benchmark rather than the overall average is the figure to compare.

Several limitations of this study should be stated. The comparison of generations is drawn from figures reported in the primary literature rather than from a single controlled experiment on one corpus, so the relative positions are robust but the absolute values are not transferable. The treatment of the Uzbek case is analytical and proposes an order of work that has not been validated experimentally, for want of the open evaluation corpus whose construction is itself identified as the first task. The benchmark figures cited reflect the state of the leaderboard at the time of writing and will change.

These limitations indicate the direction of further work: the construction of an open Uzbek evaluation set, and a controlled comparison of the normalisation, adaptation and hybrid measures proposed here on a single Uzbek corpus with the metrics set out above.

Conclusion. This article has examined semantic matching as the computational core of intelligent information systems and has classified the methods into four generations: lexical and statistical, knowledge-based, distributional and contextual neural. For each the mathematical basis, the computational cost, the treatment of synonymy and the interpretability of the output were stated, and the generations were compared on a single set of criteria.

The three architectural families were compared on accuracy, query latency and index size, and it was shown that the cross-encoder attains the highest accuracy at a latency that excludes it from first-stage retrieval, that the bi-encoder admits full precomputation, and that late interaction occupies an intermediate position at the cost of a larger index. A six-stage reference model was constructed in which each family is assigned to the stage whose cost profile it fits, and in which recall governs the first stage and precision the last.

It was further shown that the generations are complementary rather than successive, that hybrid retrieval with reciprocal rank fusion is the robust form of their combination, and that the evaluation of a model must be conducted on the retrieval section of the benchmark and on the proprietary corpus rather than on a single aggregate score. For Uzbek and comparable morphologically rich low-resource languages, an order of practical work was proposed that begins with a normalisation layer and the adaptation of an existing multilingual model, and the construction of an open evaluation corpus was identified as the precondition of further progress.

References

1. Salton G., Wong A., Yang C.S. A Vector Space Model for Automatic Indexing // Communications of the ACM. – 1975. – Vol. 18, No. 11. – P. 613–620.
2. Furnas G.W., Landauer T.K., Gomez L.M., Dumais S.T. The vocabulary problem in human-system communication // Communications of the ACM. – 1987. – Vol. 30, No. 11. – P. 964–971.
3. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond // Foundations and Trends in Information Retrieval. – 2009. – Vol. 3, No. 4. – P. 333–389.
4. Deerwester S., Dumais S.T., Furnas G.W. et al. Indexing by Latent Semantic Analysis // Journal of the American Society for Information Science. – 1990. – Vol. 41, No. 6. – P. 391–407.
5. Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet Allocation // Journal of Machine Learning Research. – 2003. – Vol. 3. – P. 993–1022.
6. Miller G.A. WordNet: A Lexical Database for English // Communications of the ACM. – 1995. – Vol. 38, No. 11. – P. 39–41.
7. Wu Z., Palmer M. Verb Semantics and Lexical Selection // Proceedings of the 32nd Annual Meeting of the ACL. – 1994. – P. 133–138.
8. Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy // Proceedings of IJCAI. – 1995. – P. 448–453.
9. Lin D. An Information-Theoretic Definition of Similarity // Proceedings of ICML. – 1998. – P. 296–304.
10. Euzenat J., Shvaiko P. Ontology Matching. 2nd ed. – Berlin: Springer, 2013. – 511 p.
11. Bordes A., Usunier N., Garcia-Durán A. et al. Translating Embeddings for Modeling Multi-relational Data // Advances in NeurIPS. – 2013. – P. 2787–2795.

12. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // arXiv:1301.3781. – 2013.

13. Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation // Proceedings of EMNLP. – 2014. – P. 1532–1543.

14. Bojanowski P., Grave E., Joulin A., Mikolov T. Enriching Word Vectors with Subword Information // Transactions of the ACL. – 2017. – Vol. 5. – P. 135–146.

15. Kusner M., Sun Y., Kolkin N., Weinberger K. From Word Embeddings to Document Distances // Proceedings of ICML. – 2015. – P. 957–966.

16. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need // Advances in NeurIPS. – 2017. – P. 5998–6008.

17. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT. – 2019. – P. 4171–4186.

18. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of EMNLP-IJCNLP. – 2019. – P. 3982–3992.

19. Karpukhin V., Oğuz B., Min S. et al. Dense Passage Retrieval for Open-Domain Question Answering // Proceedings of EMNLP. – 2020. – P. 6769–6781.

20. Khattab O., Zaharia M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT // Proceedings of SIGIR. – 2020. – P. 39–48.

21. Gao T., Yao X., Chen D. SimCSE: Simple Contrastive Learning of Sentence Embeddings // Proceedings of EMNLP. – 2021. – P. 6894–6910.

22. Malkov Yu.A., Yashunin D.A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2020. – Vol. 42, No. 4. – P. 824–836.

23. Johnson J., Douze M., Jégou H. Billion-Scale Similarity Search with GPUs // IEEE Transactions on Big Data. – 2021. – Vol. 7, No. 3. – P. 535–547.

24. Thakur N., Reimers N., Rücklé A. et al. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models // NeurIPS Datasets and Benchmarks Track. – 2021.

25. Muennighoff N., Tazi N., Magne L., Reimers N. MTEB: Massive Text Embedding Benchmark // Proceedings of EACL. – 2023. – P. 2014–2037.

26. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in NeurIPS. – 2020. – P. 9459–9474.

27. Feng F., Yang Y., Cer D. et al. Language-agnostic BERT Sentence Embedding // Proceedings of ACL. – 2022. – P. 878–891.

28. Chen J., Xiao S., Zhang P. et al. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings // arXiv:2402.03216. – 2024.

29. Conneau A., Khandelwal K., Goyal N. et al. Unsupervised Cross-lingual Representation Learning at Scale // Proceedings of ACL. – 2020. – P. 8440–8451.

30. MTEB Multilingual v2 leaderboard results. – URL: https://www.llmreference.com/benchmark/mteb-multilingual-v2 (accessed 4 November 2026).

31. Reading the MTEB score for retrieval workloads. – URL: https://techsy.io/en/blog/mteb-score-explained-rag (accessed 4 November 2026).

32. Comparative analysis of embedding models, 2026. – URL: https://futureagi.com/blogs/best-embedding-models-2025 (accessed 4 November 2026).

33. Leading and domain-specific models on the MTEB leaderboard. – URL: https://modal.com/blog/mteb-leaderboard-article (accessed 4 November 2026).