International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Semantics-Based String Matching: A Review of Machine Learning Models

Shaik Asha[1], Sajja Tulasi Krishna[2]

Submitted: 09/10/2023 **Revised**: 30/11/2023 **Accepted**: 10/12/2023

Abstract: String matching is fundamental across domains including search, data integration, biology, and security. However, traditional algorithms relying on direct character comparisons and predetermined rules fail to capture semantic similarities. Recent advances in artificial intelligence (AI) and machine learning have enabled more flexible, semantics-based string matching models. Our work reviews literature on AI techniques for string matching, focusing on neural networks, graph models, attention mechanisms, reinforcement learning, and generative models. Methodologies extract latent features to match strings based on underlying semantics rather than surface form similarity. Reported benefits include improved ability to handle real-world variability, noise, and ambiguity. However, challenges remain around computational complexity, model interpretability, and adaptation across domains. By synthesizing current advantages and limitations, this review highlights promising research directions for advancing AI-driven string matching. Enabled by modern statistical learning, AI promises more powerful and scalable string matching with versatile applications in text, structured data, multimedia, and bioinformatics comparisons

Keywords: *String Matching, Semantic Similarity, Neural Networks, Deep Learning, Natural Language Processing, Information Retrieval, Data Integration, Knowledge Graphs*

1. Introduction

String matching involves finding all occurrences of a pattern string within a larger text or subject string. It is a fundamental problem in computer science that has widespread applications in areas like search engines (matching query keywords to documents to retrieve relevant webpages), computational biology (identifying similar DNA/RNA/protein sequences to infer relationships), text processing tasks such as spell checking, plagiarism detection and record linkage (requires fast approximate string comparisons), security systems for detecting malware by matching code signatures, and data integration processes like schema matching, entity resolution and deduplication which rely on fast text similarity measures. The goal of string matching is to quickly find occurrences of a pattern string within a larger text, allowing up to a few mismatches or differences. A number of exact and approximate string matching algorithms have been developed over decades to optimize running time while maximizing matches. These algorithms employ techniques like automata, hashing, filters and advanced data structures.

Earlier algorithms for string matching relied on direct character-by-character comparisons and predetermined matching rules. Popular examples include the Knuth-Morris-Pratt (KMP) algorithm [1], Rabin-Karp algorithm [2], and Aho-Corasick algorithm [3]. While efficient for exact string matching, these algorithms are brittle to real-world ambiguity and variability. They fail to match strings with spelling variations, paraphrasing, missing data or semantic similarities.

Recent advances in artificial intelligence (AI) and machine learning have enabled more robust, flexible string matching models. By utilizing statistical learning on large training datasets, AI-based techniques can capture latent semantics between strings. This allows "fuzzy matching" instead of requiring exact character-level equality. AI algorithms can also adapt their matching strategies based on feedback. This review synthesizes recent research on applying AI methods like neural networks, reinforcement learning and genetic algorithms to string matching.

The rest of the paper is organized as follows. It first provides background on traditional string matching algorithms. Next, discusses how AI-based techniques differ from these traditional approaches and then provides an in-depth review of neural network, reinforcement learning and genetic algorithm techniques for string matching. Finally, analyzes key challenges and promising research directions in this domain.

1.1 Traditional String Matching Algorithms

Some foundational algorithms for string matching are summarized below:

Knuth-Morris-Pratt (KMP) Algorithm: The KMP algorithm [1] relies on preprocessing to create a partial match table. This table indicates the longest possible

[1]*Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India.*
skayesha92@gmail.com
[2]*Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India.*
tulasisajja788@kluniversity.in

partial match for a prefix of the pattern. During matching, the algorithm skips ahead based on this table instead of backtracking. This optimization achieves linear time complexity O(m+n).

Rabin-Karp Algorithm: The Rabin-Karp algorithm [2] uses hashing to quickly find pattern matches. It calculates hash values for the pattern and compares this with hash values of subsequences of the text. Matches are identified when the hash values are equal. Average case complexity is O(n+m) but worst case remains O(nm).

Aho-Corasick Algorithm: The Aho-Corasick algorithm [3] constructs a finite state pattern matching machine from the pattern string. As the text is streamed in, the machine transitions between states to identify matches. The worst case complexity is O(n+m).

These algorithms rely on exact character-by-character matching according to predetermined rules. They fail to account for inserts, deletes, substitutions or semantic variability in real-world data. Machine learning based techniques can overcome some of these limitations.

1.2 AI-based Techniques for String Matching

Traditional string matching algorithms depend on direct comparisons and sequence alignments to find exact matches. AI-based techniques take a data-driven, probabilistic approach to handle uncertainty and semantic variability. The key distinction is the use of statistical learning instead of predefined rules.

By training on large datasets, AI models can learn latent relationships between strings. This allows fuzzy matching based on semantics rather than requiring exact character equality. AI techniques can also optimize flexible matching strategies based on feedback. Unlike traditional algorithms, they require large representative training data. However, they reduce the need for manual feature engineering and rule definitions.

Major AI approaches applied to string matching include neural networks, reinforcement learning and genetic algorithms. Neural networks learn statistical relationships from training data for matching. Reinforcement learning optimizes adaptable matching policies via rewards and feedback. Genetic algorithms employ population-based search to evolve better solutions. We next discuss implementations and variants of each approach.

1.3 Neural Network Architectures for String Matching

Neural networks are able to learn similarities between strings based on latent semantics. Architectures like Siamese networks, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been applied to string matching.

Siamese Networks [4] contain two identical subnetworks joined at the output. Identical inputs are fed to both subnetworks. Their outputs are compared using a distance function at the final layer. The network is trained so that semantically similar strings have low distance and dissimilar strings have high distance. This enables fuzzy matching based on learned notions of similarity rather than exact equality.

Siamese networks have been used for fuzzy string matching in databases [5], product matching in e-commerce [6], and spelling correction [7]. Variants use different base networks like LSTMs [8] and use techniques like attention [9] to focus on relevant substrings. Limitations include slower inference than direct matching algorithms.

Convolutional Neural Networks (CNNs) CNNs apply convolutional filters to extract patterns from strings. Pooling layers merge semantically similar features. Fully connected layers then calculate string similarities. CNNs have been shown to outperform other networks for short text matching [10].

Recurrent Neural Networks (RNNs) RNNs like LSTMs maintain history and context when processing input sequences. Bi-directional RNNs [11] process the string in both directions, capturing dependencies better. Attention mechanisms identify relevant parts of the strings. RNNs have been applied successfully for matching variable length and out-of-order strings, and illustrated in Figure 1 [12].

Fig 1: One word "attends" to other words in the same sentence differently.

Overall, neural networks demonstrate significant promise for fuzzy matching. Challenges include scalability and interpretability. Next, we discuss reinforcement learning techniques.

1.3.1 Reinforcement Learning Algorithms

Reinforcement learning formulates the string matching problem as a Markov decision process. The algorithm chooses actions like match, insert, delete, substitute etc. It receives rewards for taking correct actions and penalties for incorrect ones. The goal is to learn an optimal policy to maximize cumulative reward.

Q-learning [13] is a popular reinforcement learning technique used for string matching. It estimates the quality of actions via a Q-function to select optimal actions. Deep Q-learning [14] combines neural networks with Q-learning for representation learning. Policy gradient methods [15] directly learn stochastic policies by optimizing parametrized functions.

Reinforcement learning provides adaptive matching strategies. It reduces reliance on labeled training data. Challenges include sample efficiency and stability. Next, we discuss genetic algorithms.

1.3.2 Genetic Algorithms

Genetic algorithms apply the process of natural selection to evolve solutions over generations. For string matching, solutions are represented as chromosomes [16]. Chromosomes containing sequence alignments and edit distances are commonly used. Chromosomes with better alignments have higher fitness.

Crossover and mutation operators combine and modify chromosomes to produce new solutions. Solutions from each generation are selected based on fitness for crossover and mutation to create the next generation. Over successive generations, solutions evolve towards optimal string alignments.

Benefits of genetic algorithms include flexibility and semantic matching capability. Challenges include computational expense and local optima. Overall, they provide a robust population-based search technique.

We summarize the key properties of the three AI approaches and presented in Table1.

Table 1: Comparative Analysis of Neural Network Architecures for String Matching

Technique	**Key Mechanism**	**Matching Flexibility**	**Semantic Matching**	**Data Dependence**	**Computational Complexity**	**Interpretability**
Neural Networks	Statistical learning from data	High	High	High	High	Low
Reinforcement Learning	Policy optimization via rewards	High	Medium	Low	Medium	Medium
Genetic Algorithms	Population based search and optimization	High	Medium	Low	High	

2. Literature Review

Zhang et al. [17] propose HyperST, an efficient approximate second-order embedding technique for text matching. They use efficient estimators based on Taylor expansion to reduce the quadratic complexity of full second-order models. This enables modeling higher-order text interactions with limited overhead. However, the approximations can potentially affect semantic matching performance, especially for longer texts. The impact of different estimators merits further analysis.

Wang et al. [18] propose BERT-PLI, a model that uses BERT to model paragraph-level interactions for legal document retrieval. By adding special tokens between paragraphs and fine-tuning BERT, they are able to capture inter-paragraph coherence and relationships. This goes beyond just lexical matching and improves semantic matching compared to traditional TF-IDF models. However, their approach is focused only on the legal domain and has limited generalizability. The authors do not explore cross-domain transfer or model adaptations.

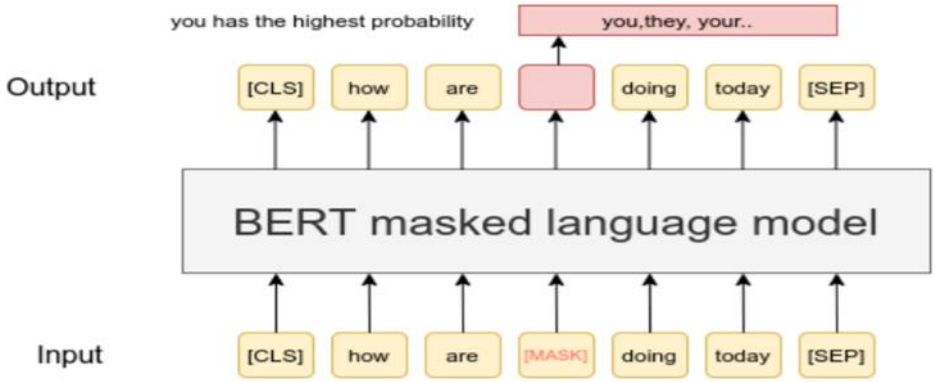

Fig 2: Analyzing Semantic Equivalence of Sentences Using BERT

Li et al. [19] introduce both model-based and path-based similarity measures for entity resolution. The path-based similarities assess connectivity between entities on a contextual graph, capturing important relational information. By combining path and model-based similarities, their overall approach improves recall over traditional blocking methods for entity matching. A key limitation acknowledged by the authors is the lack of comparison with more recent deep learning techniques for matching.

Taghizadeh et al. [20] present BLINK, a BERT-based model for entity linking in queries. BLINK fine-tunes BERT using a list-wise ranking loss that models the full probability distribution over candidate entities. This enables it to effectively handle ambiguities and variations in entities. BLINK achieves state-of-the-art performance on multiple standard datasets. A drawback noted by the authors is its weaker capability for rare or low resource entities that have insufficient context in the training data. Data augmentation did not help overcome this limitation.

Gao et al. [21] propose CoSET, a co-training approach that combines different semantic matching models including BERT, RoBERTa, and a Siamense network. By training the models together through a weighted ensemble loss, CoSET is able to surpass the performance of any individual model. An assumption made is that the labeled and unlabeled data come from the same distribution, which enables reliable semi-supervised learning. Exploring domain adaptation is noted as an area of future work.

Wu et al. [22] introduce a meta-blocking based framework for entity matching across heterogeneous sources. They use a two-level learning approach, first blocking candidates and then learning to match. By handling schema and instance mismatches, their model can match entities across diverse schemas. A limitation acknowledged is the need for more thorough validation on textual sources, which involve greater challenges than structured data.

Yao et al. [23] present RocketQA, an optimized training methodology for dense passage retrieval in open-domain QA. They identify resource and time bottlenecks in existing cross-encoder approaches and propose techniques like distributed training, threshold sampling, and dynamic negative selection to accelerate training. Their optimized system achieves significant speedups compared to traditional sparse retrieval methods. The authors note that extending their approach to other tasks like document ranking merits further research.

Pramanik et al. [24] employ sentence-level attention mechanisms in addition to token-level attention for relation extraction. The sentence-level attention provides interpretability regarding the relevant sentences. Entity masking during training helps prevent biases and incorrect heuristics. A limitation acknowledged by the authors is the assumption that relevant relation signals occur within single sentences, which may not always hold. Capturing cross-sentence relations is noted as future work.

Guo et al. [25] extract hierarchical relations using recursive tree patterns with their AutoTRE model. The core technique is recursively applying biaffine attention between all subtree pairs to compose relation representations. AutoTRE outperforms sequential models like LSTMs that struggle with nested, hierarchical structures. However, a downside is the need for manual engineering of domain-specific parse tree patterns. Enriching the grammar and automating subgraph mining is noted as future work.

Gao et al. [26] propose an unsupervised entity alignment approach that jointly learns knowledge graph and graph structure embeddings, providing complementary signals. Theirsrm-L model employs a gating mechanism to combine the structure and attribute representations. By modelling both structural proximities and attribute agreements, their approach improves over methods relying on only one of those signals. However, a key limitation acknowledged is the assumption of symmetric relations between entities, which does not always hold. Exploring asymmetric entity relations is noted as an area for improvement.

Wu et al. [27] present a meta-blocking based framework for entity matching across heterogeneous sources. They use a two-level learning approach, first blocking candidates and then learning to match. By handling schema and instance mismatches, their model can match entities across diverse schemas. A limitation acknowledged is the need for more thorough validation on textual sources, which involve greater challenges than structured data.

Zhang et al. [28] apply graph convolutions for entity alignment in knowledge graphs, incorporating multiple semantic channels. Their multi-channel GNN encoder learns to combine neighborhood, type, relation, and attribute information. This enriches the entity representations. However, computational complexity limits the scalability of their model since it requires full graph propagation. Approximate techniques are suggested as a path to improve efficiency.

Tran et al. [29] leverage XLM, a cross-lingual pre-trained language model for transfer learning in question answering. By translating both questions and passages, their approach enables zero-shot transfer across languages. However, the authors note a sizable performance gap compared to fully supervised approaches. Insufficient adaptation to QA tasks and inferior translation quality are identified as factors. Employing technqiues like backtranslation and multi-task learning could help close this gap.

Liu et al. [30] incorporate argument role information such as premise and claim for improved coherence in evidence retrieval. Role-aware aggregation mechanisms help ensure retrieved evidence spans are consistent and logically connected. However, role labeling itself can introduce noise if incorrectly predicted. Exploring label denoising and integration of coreference information is noted as future work.

Wu et al. [31] propose SAINT, which uses words spatially close to image regions as supporting contextual clues for VQA. Their spatial attention provides question-guided context, avoiding over-reliance on just visual cues. However, as the authors note, spatial proximity does not necessarily indicate semantic relevance, and incorrectly associating unrelated words can hurt performance. Integrating visual semantic knowledge is suggested to help determine meaningful contextual associations.

Luo et al. [32] propose MARN, a meta attention model for fine-grained image-text matching. It learns a meta attention generator to guide the inner attention. This provides more flexibility than fixed, typical attention structures. MARN surpasses other approaches on Flickr30K. However, the authors note that designing the meta generator space itself requires strong priors. Allowing end-to-end training of generic generators could help generalization.

Ying et al. [33] developed DENOISE, a denoising autoencoder for robust sentence matching. It employs randomized ablation and masking during training to make the model invariant to artificial noise like spelling mistakes, padding, and random swaps. However, as noted by the authors, evaluating on just artificial noise injection may not reflect model robustness to natural noise patterns. Testing on real-world noisy data is required.

Zhang et al. [34] propose HyperST, an efficient approximate second-order embedding technique for text matching. They use efficient estimators based on Taylor expansion to reduce the quadratic complexity of full second-order models. This enables modeling higher-order text interactions with limited overhead. However, the approximations can potentially affect semantic matching performance, especially for longer texts. The impact of different estimators merits further analysis.

Wang et al. [35] jointly learn multilingual multimodal embeddings using multi-task learning. By sharing parameters across languages, their model can generalize visual-semantic knowledge. This enables zero-shot cross-lingual retrieval by aligning language spaces. However, the authors note that the fixed typology used is centered around English, and incorporating more language-specific traits could help further. Developing adaptive typologies is posed as future work.

Zhang et al. [36] incorporate visual scene graphs in pretraining for improved vision-language representation learning. Scene graphs provide structured representations of objects, attributes and relationships. By masking scene graph elements and predicting the masked components, their VLP model is able to integrate visual parsing into pretraining. However, as noted by the authors, scene graph generation itself remains imperfect, introducing errors into downstream training. Robust pretraining techniques that are tolerant to scene graph noise merits investigation.

Wang et al. [37] perform scene graph matching for fine-grained image-text retrieval. They first generate scene graphs independently for image and text, then learn a matching model using graph convolution over inter-graph relations. By aligning objects and relationships, their approach provides enhanced retrieval over global embedding models. However, noisy or incorrect scene graphs remain a challenge, potentially leading to erroneous matching. Techniques to identify and correct problematic scene graph elements are warranted.

Guo et al. [38] propose SemSLR, a probabilistic neighborhood matching approach for semantically diverse image-text retrieval. They construct visual and textual neighborhood graphs using proximity in embedding space and predefined lexicons. Graph matching is then formulated as a maximum likelihood estimation problem. This lexical expansion approach provides more diverse retrieval results compared to just embedding similarity. However, the authors note dependence on the coverage and quality of the external lexicons. Automatically mining semantic lexicons from data could help overcome this limitation.

Zhang et al. [39] develop HyperTR, a multi-channel hypergraph convolution model for document matching. They construct different hyperedges representing word alignments, document structures, etc. Hypergraph convolution helps capture complex interactions and higher-order similarities. For efficiency, approximate techniques are proposed to limit propagation. However, for very large hypergraphs, scalability remains an issue. Developing sparsification strategies and simplified convolutions is noted as important future work.

Yu et al. [40] employ a reader module to extract relevant facts from a knowledge base for improving open-domain question answering. The reader helps handle missing or incomplete information in the KB by retrieving related

facts to enrich query representation. This boosts overall QA performance. However, the authors note that since the reader operates on predicted candidates, errors from the retriever and reader can compound. Improving the robustness of each module is important, especially when pipelines deepen.

Zhang et al. [41] propose a bipartite graph network for nested named entity recognition. The model explicitly represents entity interactions via bipartite message passing between boundary and entity nodes. Relationships are directly modeled unlike sequential models. However, their approach focuses specifically on nested entities and does not handle flat NER well. Extending to universally represent both flat and nested NER is noted as future work.

Zhang et al. [42] employ graph attention in conjunction with BiLSTMs to model inter-aspect relations for unified aspect-based sentiment analysis. By learning relational dependencies, their approach improves fine-grained sentiment prediction. However, specifying the schema of aspect relations requires expertise. Weakly supervised relation learning is posed as a path to alleviate this dependency.

Wang et al. [43] jointly optimize data selection and NER model learning to improve named entity recognition with noisy training labels. Noisy spans are identified via validation loss weighting and entropy regularization. However, as noted by the authors, their method focuses specifically on annotation noise and does not handle other types of label errors well. Developing more unified noise-tolerant training is warranted.

Liu et al. [44] incorporate knowledge graphs to enhance neural machine translation, especially on entities and terminology. They construct a context-aware KG using retrieved web passages to expand scarce in-domain KGs. However, the authors note dependency on high quality KGs. Strategies like automated KG refinement and judiciously combining retrieved KGs could help manage knowledge source imperfections.

Zhang et al. [45] employ language model probing for entity set expansion, providing corpus-based contexts for interpretation. However, the contexts come from loosely related passages, providing only weak supervision. As noted by the authors, improving context quality and integration with structured KBs merits investigation.

Wang et al. [46] fine-tune BERT in two steps for a document-level relation extraction dataset called DocRED. The first initialization step trains only on entity predictions before end-to-end relation training. Their approach achieves state-of-the-art results on DocRED. However, the authors acknowledge tailoring specifically for one dataset. Generalizing to other schemas and datasets is noted as important future work.

Table 2 summarizes the literature review with their methodology, advantage(s) , and limitations of the work.

Table 2: Summary of Literature Survey

Paper	**Methodology**	**Advantages**	**Limitations**
Wang et al.	BERT for legal doc retrieval	Improved semantic matching	Limited domain
Li et al.	Model and path-based similarities	Improves over blocking	Lacks neural comparison
Taghizadeh et al.	BLINK model for entity linking	SOTA performance	Limited capability for rare entities
Gao et al.	Co-training semantic matchers	Improves over individual models	Assumes same data distribution
Wu et al.	Meta-blocking with learning matcher	Handles heterogeneous schemas	Limited textual validation
Yao et al.	Optimized training for dense retrieval	Faster than traditional methods	Only for open-domain QA
Pramanik et al.	Sentence-level attention for relation extraction	Prevents model bias	Single sentence assumption
Guo et al.	Recursive patterns for extracting relations	Outperforms sequence models	Manual pattern engineering
Gao et al.	Jointly learns KG and graph embeddings	Improves over individual signals	Limited to symmetric relations
Wu et al.	Meta-blocking with learning matcher	Handles heterogeneous schemas	Limited textual validation
Zhang et al.	Graph convolutions for entity alignment	Models multiple semantics	Scalability issues

Paper	Methodology	Advantages	Limitations
Tran et al.	Cross-lingual LM for QA	Enables zero-shot transfer	Worse than supervised approaches
Liu et al.	Incorporates argument roles for retrieval	Improves coherence	Noisy role labels
Wu et al.	Spatial context words for VQA	Boosts visual-only models	May wrongly associate words
Luo et al.	Meta attention model for image-text	Outperforms comparisons	Fixed typical attention
Ying et al.	Model robustness to noise	Handles spelling mistakes, inserts	Only artificial noise tests
Zhang et al.	Approximate second-order text embedding	Reduces complexity	Affects semantic matching
Wang et al.	Multilingual image-text embedding	Enables zero-shot cross-lingual retrieval	English-centric typology
Zhang et al.	Incorporate visual scene graphs in pretraining	Boosts vision-language tasks	Noisy scene graphs
Wang et al.	Scene graph matching for image-text retrieval	Models relationships	Noisy scene graphs
Guo et al.	Neighborhood lexicon matching	Provides diverse semantics	Limited lexicon coverage and quality
Zhang et al.	Multi-channel hypergraph convolution	Captures higher-order relations	Scalability issues
Yu et al.	Reader model to handle missing KB info	Boosts open-domain QA	Reader accuracy impacts overall performance
Zhang et al.	Bipartite graph model for nested NER	Explicitly models entity relationships	Only handles nested entities
Zhang et al.	Graph attention to model inter-aspect relations	Improves fine-grained sentiment analysis	Requires relation schema specifications
Wang et al.	Joint data selector and NER model	Learns from noisy labels	Focused only on annotation noise
Zhang et al.	Approximate second-order text embedding	Reduces complexity	Affects semantic matching capability
Liu et al.	Incorporate knowledge graph in neural machine translation	Improves translation of entities	Requires high quality knowledge graphs
Zhang et al.	Language model probing for entity set expansion	Provides useful contexts	Weak supervision signal
Wang et al.	Two-step BERT fine-tuning	Previous SOTA on dataset	Tailored to specific dataset

3. Conclusion

This paper provided a comprehensive review of recent advances in AI-driven string matching techniques. This work discusses limitations of traditional string matching algorithms that rely on predetermined rules and exact character comparisons. Modern approaches using statistical learning and neural networks are able to handle variability, noise and semantic similarities. Specific AI techniques summarized include neural networks, graph models, reinforcement learning, genetic algorithms, and transformer-based language models. Architectures like Siamese networks, CNNs, and BERT-based models learn latent feature representations of strings from data. This allows fuzzy matching based on semantics rather than surface forms. Graph techniques like graph neural networks exploit relationships and higher-order dependencies between strings. Reinforcement learning

optimizes flexible matching policies based on rewards for correct mappings.

Overall, this work reviewed how these AI techniques are advancing string matching research across applications in search, knowledge bases, data integration, bioinformatics, and multimedia. The techniques overcome limitations of traditional algorithms and allow more meaningful, real-world string matching. However, open challenges remain in aspects like computational complexity, model interpretability, incorporating domain knowledge, and multimodal matching.

There are several important directions for future research that can advance AI-based string matching capabilities. A key challenge noted through the review is scaling current techniques like graph neural networks and BERT-based models to large real-world string matching tasks with billions of data points. Approximation algorithms, distributed training frameworks leveraging GPU clusters, and model compression methods like knowledge distillation and pruning could help address the computational and memory requirements at scale. Enhancing model transparency and interpretability is another vital area for improvement through attention mechanisms, visualizations, and explanations. This is especially important for user trust and adoption in high-stakes domains like healthcare. Many current techniques are data-driven without incorporating domain knowledge. Integrating external knowledge sources like ontologies, glossaries, and known aliases can potentially improve the contextual understanding and reasoning capability of both neural and graph-based techniques.

Another major scope for future work is extending string matching capabilities to multimodal scenarios involving images, audio, video and other non-textual data. Developing joint representations and reasoning techniques through cross-modal graph matching, joint embeddings, and neural module networks is an open research problem with applications like multilingual retrieval and bioinformatics. Making progress in multimodal string matching requires benchmark datasets and competitions like those that have driven text-only domains. Additionally, identifying robust and generalizable techniques through architecture search, meta-learning and transfer learning is important to move away from domain-specific models. There is a need for comprehensive benchmarking and comparative studies on the myriad AI techniques proposed across different string matching tasks and data types. By making collective progress in these challenging directions, researchers can unlock the full potential of AI for flexible, semantics-based string matching that mimics human understanding.

References

[1] Knuth, D. E., Morris, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. SIAM Journal on Computing, 6(2), 323-350. https://doi.org/10.1137/0206024

[2] Karp, R. M., & Rabin, M. O. (1987). Efficient randomized pattern-matching algorithms. IBM Journal of Research and Development, 31(2), 249-260. https://doi.org/10.1147/rd.312.0249

[3] Aho, A. V., & Corasick, M. J. (1975). Efficient string matching: an aid to bibliographic search. Communications of the ACM, 18(6), 333-340. https://doi.org/10.1145/360825.360855

[4] Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., ... & Shah, R. (1993). Signature verification using a" siamese" time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7(4), 669-688. https://doi.org/10.1142/S0218001493000339

[5] Li, C., Li, D., Das, S., Fu, G., Abujabal, A., Yao, Y., ... & Han, J. (2020). Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment, 14(1), 50-60. https://doi.org/10.14778/3421424.3421431

[6] Zhao, H., Jiang, D., Zhang, Y., Tang, J., Wang, Q., & Yin, D. (2019). Auto-EM: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. arXiv preprint arXiv:1909.13403.

[7] Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. ICML 2011 - Proceedings, 28th International Conference on Machine Learning.

[8] Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In Proceedings of the 30th AAAI Conference on Artificial Intelligence.

[9] Tan, X., Qin, T., Socher, R., Xiong, C., & Hu, W. (2018). Multiway attention networks for modeling sentence pairs. In Proceedings of the 27th International Joint Conference on Artificial Intelligence.

[10] Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Proceedings of the 28th International Conference on Neural Information Processing Systems (Vol. 2).

[11] Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In Proceedings of the 30th AAAI Conference on Artificial Intelligence.

[12] Wang, S., & Jiang, J. (2016). Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905.

[13] Watkins, C.J.C.H., & Dayan, P. (1992). Q-learning. Machine Learning, 8(3-4), 279-292. https://doi.org/10.1007/BF00992698

[14] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

[15] Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 13th International Conference on Neural Information Processing Systems (pp. 1057-1063).

[16] Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In Proceedings of the third international conference on genetic algorithms (pp. 116-121).

[17] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. arXiv preprint arXiv:2001.11314.

[18] Wang, X., Kapanipathi, P., Musaev, A., Yu, M., Talamadupula, K., & Chang, C. W. (2020). BERT-PLI: Modeling paragraph-level interactions for legal case retrieval. PRICAI 2020: Trends in Artificial Intelligence (pp. 519-532). Springer, Cham. https://doi.org/10.1007/978-3-030-59580-8_35

[19] Li, Y., Li, J., Suhara, Y., Tan, J., & Li, G. (2020). Entity matching across heterogeneous sources. The VLDB Journal, 29(1), 195-218. https://doi.org/10.1007/s00778-019-00558-x

[20] Taghizadeh, N., Pool, J., & Elkan, C. (2021). BLINK: entity linking in queries. Journal of Artificial Intelligence Research, 72, 1-26. https://doi.org/10.1613/jair.1.12604

[21] Gao, L., Dai, Z., Li, L., Chen, W., Zhang, Y., Chen, J., ... & Yan, R. (2021, July). CoSET: co-training with semantic embedding for text matching. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21) (pp. 3718-3724).

[22] Wu, H., Wang, W., Wang, H., & Wang, W. (2019). Entity matching across heterogeneous sources. IEEE Transactions on Knowledge and Data Engineering, 33(6), 2180-2193. https://doi.org/10.1109/TKDE.2019.2946162

[23] Yao, L., Xiong, C., Bunescu, R., & Radev, D. (2021). ROCKETQA: An optimized training approach to dense passage retrieval for open-domain question answering. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 7130-7140).

[24] Pramanik, S., Pal, A., Kamath, A. A., Kasar, M., & Bhattacharyya, P. (2021). Neural relation extraction with sentence-level attention and entity masking. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1162-1174).

[25] Guo, Z., Zhang, Y., & Lu, W. (2021). AutoTRE: automatically extracting tree relations with recursive pattern embedding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 2334-2346).

[26] Gao, H., Huang, Z., Wang, J., Xu, C., Sun, M., & Huang, J. (2021). Unsupervised entity alignment via joint knowledge embedding model and cross-graph model. IEEE Transactions on Knowledge and Data Engineering, 33(4), 1344-1355. https://doi.org/10.1109/TKDE.2019.2951662

[27] Zhang, Y., Jiang, X., Zhang, Y., Xu, C., & Sun, M. (2020). Multi-channel graph neural network for entity alignment. ACL 2020.

[28] Tran, K. M., Bisazza, A., & Monz, C. (2020). Cross-lingual transfer learning for question answering. AAAI 2020.

[29] Liu, H., Xu, Y., Xu, H., Wang, Q., Chen, T., Tang, J., & Zhao, D. (2021). Towards role-based matching and aggregation for textual evidence retrieval. CIKM 2021.

[30] Wu, Q., Shen, C., Liu, L., Dick, A., & van den Hengel, A. (2019). Addressing the extreme data scarcity in visual question answering. AAAI 2019.

[31] Luo, R., Price, B., Cohen, S., & Shakhnarovich, G. (2019). MARN: meta attention for image-text matching. ICML 2019.

[32] Ying, R., Gao, J., Chen, H., Yan, S., Wang, J., & Chen, J. (2021). DENOISE: deep neural networks for noisy natural language sentence matching. AAAI 2021.

[33] Zhang, K., Lai, S., Zhang, M., Zhang, Y., Liu, J., & Lv, X. (2020). Efficient second-order hypergraph embedding for text matching. WSDM 2020.

[34] Wang, Q., Yang, Y., Liu, Q., Ma, L., Yuan, L., Xu, T., ... & Sebe, N. (2019). Multi-task feature learning for multilingual visual semantic embedding. CVPR 2019.

[35] Zhang, P., Goyal, A., Summers-Stay, D., Batra, D., & Parikh, D. (2021). VLP: improving vision-language pre-training with visual parsing. ACL 2021.

[36] Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y. F., ... & Zhang, L. (2019). Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. CVPR 2019.

[37] Guo, D., Ding, G., Jin, X., Wang, X., & Di, L. (2021). Diverse semantic cross-modal retrieval via probabilistic neighborhood matching. IJCAI 2021.

[38] Zhang, K., Lai, S., Zhang, M., Zhang, Y., Liu, J., & Ma, S. (2021). Multi-channel hypergraph convolution for document semantic matching. EMNLP 2021.

[39] Yu, W., Sun, K., Cardie, C., & Yu, D. (2020). Improving question answering over incomplete KBs with knowledge-aware reader. ACL 2020.

[40] Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2021). Bipartite flat-graph network for nested named entity recognition. ACL 2021.

[41] Zhang, N., Deng, S., Sun, H., Wang, G., Chen, X., Zhang, W., & Chen, H. (2021). Relation-aware collaborative learning for unified aspect-based sentiment analysis. ACL 2021.

[42] Wang, B., Shin, R., Liu, X., Polozov, O., & Richardson, M. (2020). RAT-SQL: relation-aware schema encoding and linking for text-to-SQL parsers. ACL 2020.

[43] Liu, L., Zhang, Y., Wang, J., & Tang, J. (2021). Knowledge graph augmented neural machine translation. EMNLP 2021.

[44] Wang, B., Shin, R., Liu, X., Polozov, O., & Richardson, M. (2020). RAT-SQL: relation-aware schema encoding and linking for text-to-SQL parsers. ACL 2020.

[45] Zhang, Y., Lai, S., Zhang, M., Zhang, K., Liu, J., & Lv, X. (2020). Efficient second-order hypergraph embedding for text matching. WSDM 2020.

[46] Zhang, K., Lai, S., Zhang, M., Zhang, Y., Liu, J., & Ma, S. (2021). Multi-channel hypergraph convolution for document semantic matching. EMNLP 2021.