Computers and Artificial Intelligence; Vol.3, No.5; 2026
E-ISSN: 3006-3728 P-ISSN: 3006-371X
DOI: https://doi.org/10.70267/cai.v3n5.6376
Published by: Zeus Press

A Review of Multi-Label and Multi-Document Research in Community Question Answering

Yahan Tan*

School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, Jiangxi, China

**Corresponding author: Yahan Tan.*

Abstract

Community Question Answering (CQA) platforms function as continuously evolving repositories of user-generated knowledge, where questions are often informal, intents may be multiple, evidence is distributed across sources, and knowledge changes over time. These characteristics limit the applicability of conventional single-label and single-document approaches to complex information needs. This review traces the evolution of CQA from retrieval and ranking through neural semantic matching and pretrained Transformer models to retrieval-augmented generation and large language models, with particular emphasis on multi-label question understanding and multi-document evidence integration. Existing studies are compared from the perspectives of semantic representation, intent modeling, evidence retrieval and fusion, knowledge grounding, and answer reliability. The literature shows that substantial progress has been made in individual capabilities, but a unified mechanism that coordinates multi-label intent interpretation, dynamic retrieval, evidence verification, and grounded generation remains insufficiently developed. Future research should therefore move toward closed-loop architectures and evaluation frameworks that jointly measure label identification, evidence coverage, source attribution, and final answer quality in realistic community settings.

Keywords

community question answering, multi-label text classification, multi-document question answering, large language models, retrieval-augmented generation, knowledge grounding

1. Introduction

Community Question Answering sites are more than just regular question-answering tools. They actually work as collections of user-made knowledge that changes all the time. People who use these sites keep adding new content, fixing old stuff, disagreeing with points, and adding extra details, which means that any single question and answer pair can change after it gets posted. So how useful an answer is doesn't just rely on how well it matches the meaning of the question. It also needs to fit the exact information people are looking for, its claims have to be backed up by evidence that people can actually check, and the basic info in it has to still be right when someone reads it later. These demands get harder when a single question asks for several different things, or when evidence you need is split across many different posts. Earlier studies looking at CQA also note these systems match up to complex, personal, context-specific needs for information, which sets them apart from traditional fact-based QA [1].

© 2026 The Author(s). Authors retain copyright in their articles.

Community question-answering (CQA) questions are very different from the standard question-answering tasks you usually see. Questions from online communities don't usually follow a set format, and the lines between what's being asked and the words used to ask it can be unclear. The space where answers live is open, and it often has different suggestions that don't agree with each other, instead of just one right answer. Useful proof can also show up in comments, linked files, or older conversations, instead of just sitting in the first answer people find. Because of all that, methods that work well when you're only classifying into one label or pulling text from a single document don't always work straight away in real-world CQA situations.

Early work on CQA mostly described the problem as something about retrieval and ranking. Jeon and his team looked into how questions people just submitted could be matched with similar existing questions from big question and answer collections. This meant finding candidate questions become one of the most important basic steps [2]. Surdeanu and others added learning-to-rank methods that put together text features and context features to make answer ranking work better [3]. SemEval-2015 Task 3 gave a standard shared evaluation framework for people working on answer selection, and this helped make answer selection in CQA become a separate research problem on its own [4]. All these early studies point out a basic starting point: before any system can analyze and use community knowledge, it first has to find content that might be related to the question.

Frameworks that focus on retrieval still have long-lasting problems. Lexical overlap does its job well when people talk about the same problem using similar words. But technical groups often have abbreviations, terms that change between versions, and informal descriptions for the same thing. Because of this, answers you get back might be on the right topic, but they still miss what the user actually needs to work with. Retrieval methods also struggle when you want to put together bits of different answers, or explain why one piece of evidence from the community is better than another. Deep learning changed CQA by moving from hand-built similarity features to learned versions of question and answer pairs: some encoders that use convolution and recurrence made the system more okay with different word choices, and interactive models made the comparison between questions and answers clearer [5 – 7]. These new systems got better at matching meaning, but their main goal was still just sorting out whether a candidate answer fits a question. They don't directly deal with what to do when no single candidate has the full solution you need.

Transformer models that are pretrained have gotten even better at representation learning. BERT showed that contextual pretraining is really useful for lots of different natural language processing tasks [8], and Sentence-BERT made it possible to use sentence-level embeddings for doing large-scale similarity searches [9]. When it comes to CQA, getting better results does not just come from improved word representations. It also comes from context encoders that can tell different meanings based on context, and pass knowledge between similar tasks. Still, there are built-in limitations to these models: just having a stronger encoder doesn't mean it will automatically do multi-document synthesis, check evidence clearly, or have ways to keep knowledge up to date. Big language models have opened up more options for designing CQA systems, because they let systems understand combined requests, compare passages that got pulled up, and make answers that are easier to read. But this generation ability also created a new problem with reliability: answers that sound smooth might have claims that don't have any evidence to back them up [10,11]. Retrieval-Augmented Generation, or RAG, deals with part of this problem by letting the generator use outside evidence instead of only depending on what it memorized in its parameters [12].

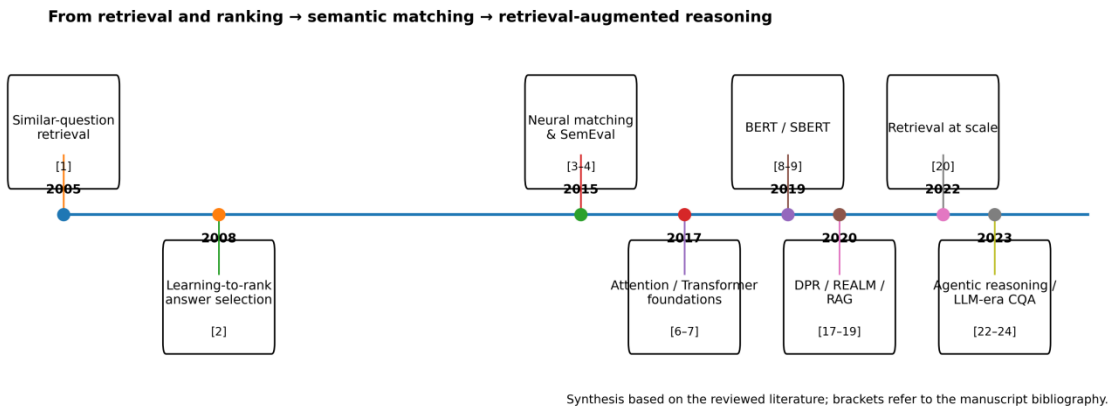
This review has three main points that it adds to the existing research. One key point is that we group related studies by following the development from tech changes over to the dimensions of capability, the datasets people use and how people evaluate the work, then finally talk about open problems and what might happen next. This is different from just putting all existing model architectures in a plain list. Another point is that we talk about how to understand multi-label questions and how to do multi-document evidence reasoning all together in one common CQA work flow. By doing this, we make it clear how these two parts work with each other when you need to find what the user wants, get the right evidence, and then build the final answer. The third point is that this review sums up common problems that exist at the whole system level. These problems have to do with how up-to-date the knowledge is, how trustworthy the sources are, whether evidence is checked properly, and how humans and AI work together. On top of that, we also point out that coordinated optimization that's based on solid evidence is a really good direction to keep working on.

© 2026 The Author(s). Authors retain copyright in their articles.

2. The Way Community Question Answering Has Changed with Technology

If we look at how CQA has changed, it's more like a shift in how problems are set up instead of just one new model architecture after another. The earliest methods put their focus on finding questions that are similar and existing answers of good quality. Neural-based methods worked on how to show and compare the semantic links between questions and their answers. Transformers plus pretrained language models made contextual understanding and transfer learning much better, and at the same time retrieval augmentation and large language models started to put together outside knowledge, complicated reasoning and natural language generation all in one single workflow. None of these steps rule each other out: modern systems usually keep retrieval, ranking, semantic encoding and generation components working together at the same time. As you can see in Fig. 1, this whole development is a step-by-step move that starts with retrieval and ranking and ends up with retrieval-augmented reasoning.

Figure 1: How CQA technology has changed: starting from retrieval and ranking, then moving toward retrieval-augmented reasoning



2.1 Getting and Ordering

In the earliest days, most CQA systems treated new questions just like a retrieval and ranking problem. Online communities ended up building up huge collections of old questions and answers over time. Because of this, finding similar existing questions and reusing their answers became really useful in practice. Jeon and their team looked into how to find similar questions in these big question-and-answer collections, and they showed that finding candidate questions is a really basic part of the whole process [2]. After that, Surdeanu et al. Used learning-to-rank methods to put together text features and context features, which worked better for ranking candidate answers [3]. Then SemEval-2015 Task 3 came up with a more standard way to test how well CQA systems pick answers, and it also split answer selection apart from general information retrieval [4]. The main goal of this early stage was just to find an answer that already existed. On the good side, this approach was efficient for computing power, and how it worked when retrieving answers was pretty easy to see. But the bad side is that it depended way too much on matching words exactly and using hand-made features. It didn't work well with reworded questions, hidden requirements, or questions that needed to put multiple answers together. When the amount of content in these communities got bigger, these problems pushed people to move away from matching surface-level words and start using learned representations of meaning instead.

2.2 Neural Semantic Matching

Instead of needing a lot of manually built similarity features, neural semantic matching instead uses learnable continuous representations to get the job done. A common way to do this is to first turn a question and an answer into their own codes separately, or sometimes create more detailed connections between them, and then sort the possible candidates by how well they match on a score. Severyn and Moschitti used convolutional neural networks for ranking pairs of short texts, and showed that deep representations can actually learn good matching connections between pairs of text [5]. After that, Tan et al. Added attention mechanisms to make the question-answer matching representations work better [6]. If you compare these

© 2026 The Author(s). Authors retain copyright in their articles.

methods to older ones like TF-IDF and BM25, they don't need as much lexical overlap and can learn more meaningful semantic features by training the whole system from start to finish all at once.

Later, interactive matching models started to pay more attention to matching individual tokens between questions and answers. Zhang and their team put forward an attentive interactive neural network. This model clearly describes how questions and answers interact with each other, so that it can do a better job in picking out correct answers for CQA [7]. This kind of research brings out a key point: picking the right answer isn't the same thing as finding similar topics. A working system has to check whether what's in the answer actually answers the constraints, conditions and requirements that the question brings up. Dual encoders are good for quickly searching through lots of candidate answers, since you can calculate the representations of answers ahead of time. On the other hand, models that are built on interactions are better for doing detailed sorting again, even though this costs more computing power. Because of this, the two-stage approach — first quick screening, then detailed matching — became widely used, and later it was an important base for QA systems that use pre-trained language models.

Even so, neural semantic matching still has three big problems to deal with. For one thing, how well it works usually relies on having quite a lot of labeled data, which means these models struggle when there is not enough data in low-resource areas or new topics that just popped up. For another, the models mostly learn things from their training datasets, and they cannot directly get new community knowledge that appears after the training step is done. The last thing is, the main thing a matching model gives you is still a relevance score, and that's not enough for combining evidence across documents, comparing different sources, and doing multi-step reasoning. These problems have pushed people to start using large-scale pretrained models, which have better context representations, better ability to transfer knowledge and also better generation ability.

2.3 Transformer and Pretrained Language Models

Transformer models are really different from CNNs and RNNs at the root, because the main thing they rely on is self-attention instead of local convolution or repeating updates to the state. One big plus is that they can directly connect different positions in a sequence through this attention, and at the same time they allow a lot of parallel computing to happen. That makes them work better for tracking long-range connections between parts of a sequence [13]. Right now, most big language models are made with Transformers or their changed versions. You shouldn't just put Transformer in the same category as older feed-forward neural networks, even though it does have feed-forward sublayers inside it. The thing that makes Transformer what it is still its attention-based design, and how all the different modules work together to model sequences.

In its original version, the Transformer is made up of an encoder and a decoder. The encoder takes an input sequence and changes it into contextualized internal representations. BERT is one well-known encoder-only Transformer, and it's especially good at language understanding, classification, and matching [8]. When you use it for CQA, BERT can put questions and candidate answers into context when it encodes them, and after that it uses a classification or ranking layer to work out how relevant they are. The usefulness of BERT isn't just that it turns text into numbers for the computer; it actually uses bidirectional context links to build representations that include words all around the current one, which in turn helps with getting the right word meaning, matching things semantically, and picking the correct answer.

The decoder generates an output sequence autoregressively based on the available context. GPT is a representative decoder-only Transformer: its input is first transformed through token embeddings and positional information, after which causal attention is used to predict subsequent tokens. It therefore does not contain a separate encoder module. GPT-3 further demonstrated that large language models can perform a wide range of language tasks through few-shot learning [14]. Meanwhile, Sentence-BERT adopts a Siamese-network structure to obtain vector representations suitable for sentence-level similarity computation and large-scale semantic retrieval [9]. Borgeaud et al. further explored retrieving information from extremely large external corpora to augment language models [15]. Thus, the Transformer stage should not be understood as a simple replacement of retrieval by generation; rather, it gradually established a functional division in which encoders support understanding and matching, vector models support large-scale retrieval, and generative models support language organization and reasoning.

© 2026 The Author(s). Authors retain copyright in their articles.

2.4 Retrieval Augmentation and Large Language Models

Big language models have pushed CQA (Community Question Answering) past just getting answers, into a whole combined process that includes understanding questions, finding evidence, thinking through problems, and making new responses. The fact that they can understand mixed requests, compare lots of different possible evidence passages, and put together natural language answers opens up new ways for people to get access to community knowledge. But the knowledge that gets stored inside the model parameters has limits when it comes to time and specific topic areas, and it's really hard to keep it updated with fast changes in technical communities. Retrieval-Augmented Generation (which people call RAG for short) fixes part of this problem by letting generative models get information from outside sources [12,16]. REALM added retrieval into the pretraining step for language models [17], and Lewis and his team came up with RAG and showed that using retrieval helps a lot with knowledge-heavy content generation [12]. Other work has looked into making or expanding on context information instead of just depending on traditional retrieval methods [18]. All these projects together have changed CQA: instead of just using static knowledge, we now get dynamic access to evidence, but how well it works still depends on how good your retrieval is, how complete your evidence is, and how trustworthy your sources are.

Building on RAG, agentic approaches break planning, retrieval, reasoning, verification, and generation into multi-step processes to organize them better. ReAct mixes reasoning and action together, so that language models can call on outside tools and keep looking for more information based on the results they got halfway [19]. Reflexion uses feedback from the model itself to help change its answer later on [20]. When you work with CQA, these kinds of methods can make the model go get more information if the first evidence it found isn't enough, and then compare the facts from all kinds of different sources. We also know from hallucination research that even outputs that sound smooth and fluent can have statements that aren't backed up by good enough evidence [10,11]. Liévin et al. Looked at how large language models reason when they answer medical questions [21], and Singhal et al. Did work with Med-PaLM to find out what large language models can and can't do in clinical knowledge situations [22]. Because of this, right now researchers care more and more about not just whether a system can spit out an answer, but also whether that answer has good support, whether the evidence adds up, and whether the model can tell when it isn't sure about something.

3. Capability-Oriented Classification of Methods

Existing CQA reviews often classify methods according to their dominant architecture or learning paradigm, such as retrieval models, neural encoders, pretrained Transformers, and generative systems. This organization is useful for reconstructing technological change, but it can obscure functional differences within the same architectural family. Two Transformer-based systems may share an encoder while serving very different roles in an end-to-end CQA pipeline: a lightweight Transformer retriever may narrow the candidate pool, whereas a large language model may compare evidence and construct the final response. Treating both simply as “Transformer-based CQA” loses distinctions that are more relevant to system design.

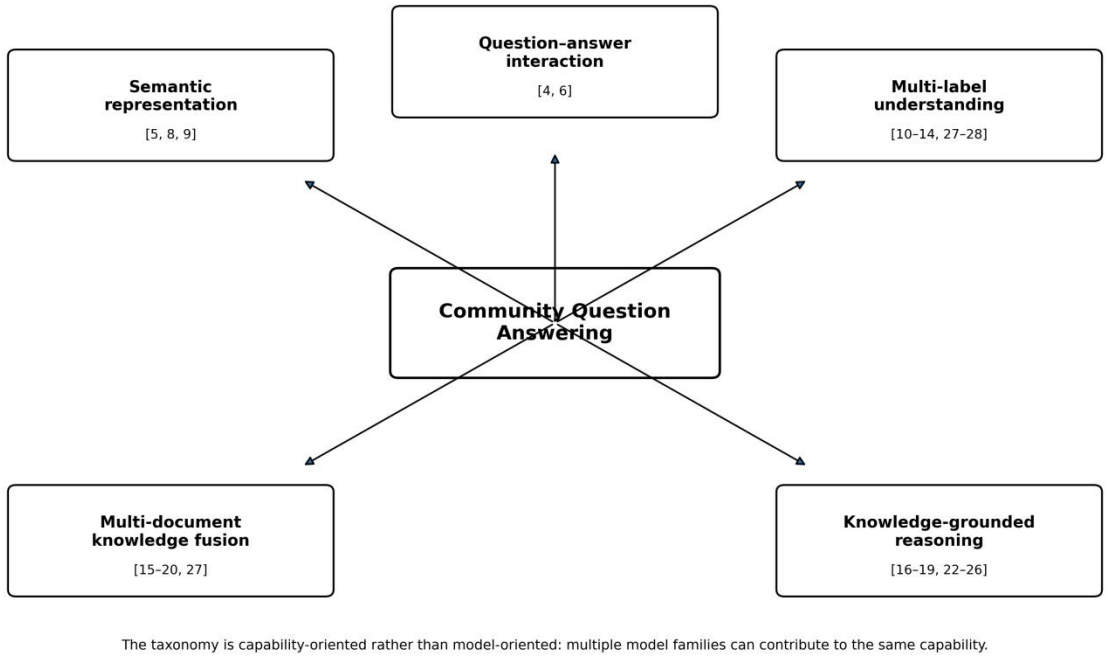
So this review uses overall capability, not just a single model structure, as the second analysis perspective. We split the related studies into five connected parts: semantic representation and answer picking; multi-label question understanding; knowledge integration and evidence reasoning across multiple documents; how multi-label and multi-document processing work together; and knowledge grounding along with reliable generation. In this context, the term “multi-label” covers two things: labels that come directly from community platforms, and multi-dimensional semantic or intent labels that researchers build to better understand questions. These two different kinds of labels don't have the same sources, they don't have the same level of semantic detail, and they're evaluated in different ways. Figure 2 gives an overview of this capability-focused sorting that the review uses, and it also shows how all five capabilities link together in a common CQA work process.

Semantic representation and answer selection Semantic representation and answer selection are two core skills that CQA systems depend on. Traditional methods like TF-IDF and BM25 use lexical statistics, which give you fast retrieval and work with mature codes. But these methods don't do so well when you need to handle semantic equivalence between different ways of saying the same thing. Neural models cut down how much they rely on lexical overlap by using continuous vector representations, and Transformers make contextual representations even better [5,6,7,13,8,9]. Answer selection is more than just finding general semantic similarity: the system needs to figure out if a candidate answer actually solves the user's question,

© 2026 The Author(s). Authors retain copyright in their articles.

not just that it talks about the same topic. Zhang and their team's attentive interactive network shows that detailed question-answer interactions can tell apart candidates that are close in meaning but not equally good at answering the question [7]. Studies that use pretrained models also show that better contextual representations help pick the right candidate answer better [23]. So for most large-scale CQA systems, a staged architecture works well: first BM25 or dual encoders do quick recall, then interaction-based models or large language models take over for reranking, checking the evidence, and putting the final answer together.

Figure 2: Classification of community question answering methods based on capability



Multi-label question understanding Understanding multi-label questions looks at what one single question can talk about all at once. In real life, questions asked in online communities usually cover more than one topic, purpose or requirement. For example, a technical question could talk about setting up the environment, checking if versions work together, and finding out what went wrong, all in one go. In general, multi-label learning connects an input X to a group of labels Y. In other words, it learns f(X) → Y [24,25]. It's different from single-label classification because multi-label learning has to figure out which labels are correct, and also deal with things like how labels connect to each other, imbalanced classes, and changes in the label space. Tsoumakas and Katakis wrote a well-organized introduction to multi-label learning [24], and Zhang and Zhou talked about why connections between labels matter a lot [25]. XMLC, sequence-generation models, and AttentionXML all work on solving calculation and statistics problems that come with big label spaces. Each approach uses a different angle: extreme multi-label modeling, creating label sequences, and organizing labels into a tree [26 – 28]. When looking at Chinese multi-label research, Xiao et al. Added meaning information from labels into the classification process [29], and Li et al. Looked at how to automatically classify multi-label text for ICD-9-CM3 [30]. Putting all these studies together, we can see that labels shouldn't just be seen as separate categories. Instead, they need to be treated as information structures that have meaning, a hierarchy, and relationships between each other.

If we use multi-label learning on CQA, the main problem changes to handling open label spaces and tricky combinations of different intents. Labels from the community often have a really clear long-tail pattern: popular labels have lots of training data, but rarer technical labels don't have much data even though they're really good at telling things apart. Also, new frameworks, tools and terms keep popping up all the time, which makes it really hard to keep a fixed set of labels up to date. Large language models can lower the cost of adjusting for new labels by using prompts and context info, but if you don't put any limits on how they generate output, they might mix up label boundaries and give outputs that don't have a consistent format. One design that gives more control is to let a large language model work out complex or unclear intent, but at the same time use a controlled label system, classifier, or post-processing step to keep the final set of labels from getting out of hand. The main goal isn't just to get the highest possible accuracy for label predictions. Instead, we want

© 2026 The Author(s). Authors retain copyright in their articles.

to build a structured way to show what information people need, which can directly help with later searching steps. Newer research on open-world multi-label learning also points out that building the label space can actually become part of the learning problem itself, especially when you don't have enough or any labels, and this is even more true for long-tail categories [31].

Multi-document knowledge integration and evidence reasoning. Multi-document knowledge integration and evidence reasoning are all about figuring out where you can find the evidence you need for a question and how you put that evidence together. Most knowledge that people share in communities is spread out over many different answers, comments, linked documents and talks from the past. It also often has repeated info, missing bits or even things that contradict each other. Because of that, answering questions that use multiple documents isn't just about pulling more material together. It needs a bunch of different decisions: you have to recall possible answers, filter out bad evidence, compare different sources, put stuff from across documents together and finally draw a conclusion. Open-domain question answering gives some really important basic tech for getting this process done. Chen and his team looked at open-domain questions by pulling info from huge knowledge sources [32]. HotpotQA made a point to include multi-paragraph putting together of evidence and multi-step thinking right into how it sets up its tasks [33]. DPR made large-scale finding of passages better by using dense semantic representations [34], and REALM and RAG went a step further by combining finding info with language models [17,12]. All these approaches give CQA parts that you can reuse, but you can't say they're exactly the same as real community situations. That's because evidence in communities has a lot more noise, changes over time, and people disagree with each other more than in these controlled setups.

Studies about multi-document QA also show that how much you can get from retrieval by itself is a really important condition for good reasoning. Ni et al. Found that multi-document QA can have problems with false-negative retrieval checks: contexts that are useful for supporting answers might get left out of the set you retrieve, and this limits how good the later reasoning can be [35]. Even a really good generative model can't get back information that was never included in its context. What's more, answers from community sources have non-text clues like how experienced the author is, when they posted it, and what other people in the community think. When picking which evidence to use, we need to think not just about how related the meaning is, but also how trustworthy the source is, how up-to-date it is, and whether different sources agree with each other. Self-RAG puts together retrieval decisions, content creation, and self-checking, so the model can use and check outside evidence in a flexible way [36]. CRAG checks how good the retrieval is and will start a new corrected retrieval and filter information when there isn't enough evidence [37]. All these studies show that research has moved from “getting more content” to “figuring out which content is actually useful”, but even so, doing good multi-document reasoning in CQA still has to deal with different kinds of sources and changing knowledge.

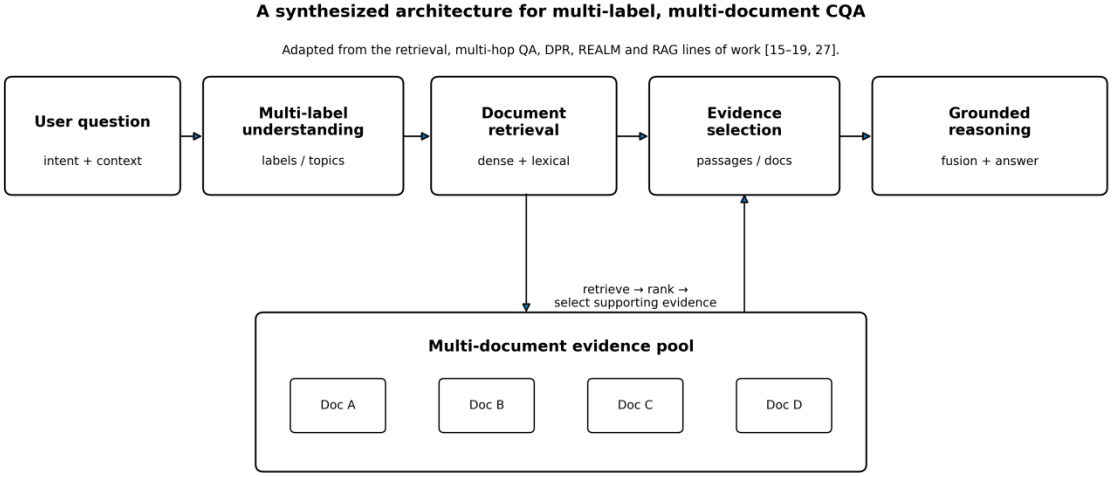
Coordination between multi-label and multi-document processing. The key to coordinating multi-label and multi-document processing is not simply to place the two tasks in sequence, but to allow what the question requires to directly influence what is retrieved and how evidence is organized, while also allowing retrieved evidence to revise the initial label interpretation. For community questions containing multiple topics, conditions, or objectives, multi-label representations can function as structured queries that decompose complex needs into several retrieval dimensions and thereby improve evidence coverage. If evidence associated with a particular label remains consistently absent, the system can use evidence feedback to reconsider whether that label is necessary, too broad, or whether an implicit intent has been missed. Multi-label prediction therefore becomes an intermediate representation for retrieval and reasoning rather than merely the endpoint of classification.

Direct work on this coordination remains limited and task-specific. Li and Li proposed a multi-label multi-document classification model using real Zhihu data for supplementary question-answer recommendation [38]. The study reports gains in precision, recall, and F1 on three meta-tasks, but its objective is supplementary-answer recommendation rather than open-ended evidence-grounded generation [38]. This distinction matters: identifying complementary answers is narrower than jointly interpreting multiple intents, retrieving heterogeneous evidence, resolving conflicts, and generating a cited response. The current literature therefore suggests a clear complementarity between multi-label understanding and multi-document evidence reasoning, while a mature end-to-end optimization paradigm has yet to emerge. Promising directions include label-guided retrieval, evidence-feedback-based label revision, dynamic retrieval based on evidence coverage, and joint

© 2026 The Author(s). Authors retain copyright in their articles.

objectives that explicitly couple intent completeness with evidence completeness. Figure 3 illustrates the proposed coordination between multi-label question understanding and multi-document evidence processing.

Figure 3: An architecture for coordinated multi-label and multi-document community question answering



Knowledge grounding and reliable generation. The main point of knowledge grounding is to make sure that any important claims you make in a generated answer can be tracked back to some clear, checkable evidence from outside the model. Traditional CQA mostly focuses on getting a good existing answer that matches the question, but knowledge-grounded systems go further — they need all the facts, opinions, and suggestions in the response to be clearly linked to evidence that the system pulled up earlier. RAG gives a basic way to do this kind of linking [12,16], but just adding retrieval step doesn't automatically make sure the answer is reliable. If the evidence you pull up is missing some parts, old, or even contradicts itself, the generator might still put together wrong conclusions or make the weaknesses in the original sources even bigger.

To get good knowledge grounding needs to look at three things together: how good the evidence is, who the source is, and checking each claim separately. ReAct and Reflexion add planning, looking up information, and thinking again when creating answers [19,20]. Self-RAG makes better use of evidence and controls how things are generated by changing retrieval steps and checking its own work [36], while CRAG checks how good the retrieved information is and gets better information when there's not enough evidence [37]. KEPR makes answering general knowledge questions better by adding knowledge and ranking how likely things are [39]. In areas where mistakes cause big problems, Liévin and other researchers showed that even when a large language model gets the right final answer, the steps it took to get there might still have parts that aren't reliable [21]. Similar work done in medicine also shows that putting large language models together with special knowledge can make answering hard questions better, but the reliability and safety still needs to be looked at carefully [22]. Future CQA systems need to check not just if the final answer is similar to a correct one, but also if all the main claims come from sources that you can find, are up to date, and are properly supported.

4. Datasets and Evaluation

Datasets determine which capabilities can be observed and the extent to which experimental conclusions can be generalized. Early CQA datasets are well suited to studying similar-question retrieval and answer ranking, whereas open-domain QA benchmarks such as MS MARCO [40], Natural Questions [41], and HotpotQA [33] have strengthened the evaluation of passage retrieval, information localization, and multi-hop reasoning. These benchmarks cannot fully replace real community archives, however, because community content exhibits greater linguistic informality, user interaction, competing answers, viewpoint conflict, and temporal change. A more appropriate research design is therefore to use real CQA data to assess scenario adaptation while employing standard open-domain benchmarks for comparable evaluation of retrieval, matching, and multi-document reasoning components.

Traditional community question-answering datasets. Yahoo! Answers is a representative early large-scale general-purpose CQA source, while Stack Overflow has a distinctive technical-community structure; both illustrate the informal language, topic diversity, and community interaction that differentiate CQA from

© 2026 The Author(s). Authors retain copyright in their articles.

curated QA benchmarks [1]. Stack Overflow tags can also provide a useful empirical signal for multi-label research because a single post may carry multiple technical topics, although native tags reflect community practice rather than a strictly controlled semantic ontology. The SemEval-2015, SemEval-2016, and SemEval-2017 community question-answering tasks further provided standardized settings for answer selection, question similarity, and related CQA subtasks [4,42,43].

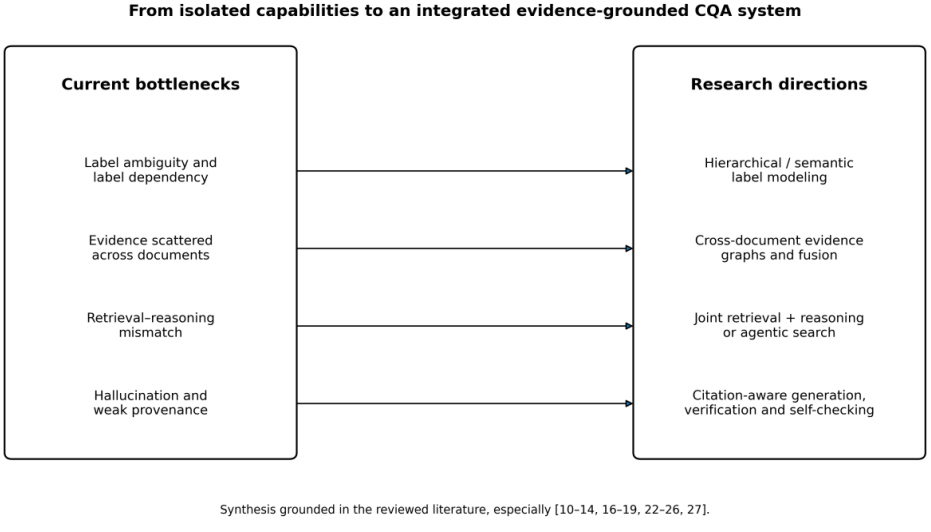
Question-answering benchmarks. MS MARCO was constructed from real search queries and passages and has become an important benchmark for passage retrieval and open-domain question answering. It is useful for evaluating retrieval components within a CQA pipeline, but its search-oriented construction does not fully capture the social and conversational characteristics of community answers. Natural Questions uses search-engine questions and Wikipedia as major answer sources and is suitable for studying information localization and answer generation, but its evidence structure is more controlled than that of community archives. HotpotQA emphasizes obtaining evidence from multiple passages and performing multi-hop reasoning, making it particularly relevant to multi-document CQA [33]. However, its evidence has already been curated by task design. These datasets should therefore be treated as technical benchmarks for community-oriented multi-document QA rather than as pure CQA datasets.

Evaluation metrics. Evaluation metrics should be aligned with the specific task. For ranking tasks, MAP measures overall ranking quality across multiple relevant answers, MRR emphasizes the rank position of the first correct answer, and NDCG is appropriate when candidates have graded levels of relevance. For multi-label classification, Precision, Recall, Micro-F1, and Macro-F1 should be reported together. Micro-F1 is more strongly influenced by frequent labels, whereas Macro-F1 is more sensitive to performance on long-tail labels. For generative question answering, lexical-overlap metrics alone cannot fully capture quality; evaluation should distinguish answer correctness, relevance to the question, and evidence-grounded faithfulness. In real community settings, human evaluation is also needed to assess usefulness, clarity, credibility, and whether different viewpoints are presented appropriately. For future multi-label–multi-document research, a more informative evaluation framework should jointly report label identification, evidence recall, evidence coverage, claim-level attribution, and final answer quality so that errors can be localized to specific stages.

5. Challenges and Future Directions

The challenges identified in the literature are not independent; together they point to a systemic issue: the inputs, evidence, and knowledge states in CQA are all dynamic. Multi-label interpretation determines retrieval directions, retrieval completeness affects evidence reasoning, source credibility influences generated conclusions, and evidence insufficiency discovered during generation may in turn trigger renewed retrieval. Future research should therefore move from isolated capability optimization toward cross-module evaluation and coordination. Figure 4 summarizes the resulting research framework, moving from isolated capability optimization toward an integrated evidence-grounded CQA system.

Figure 4: From isolated capabilities to an integrated evidence-grounded CQA system



© 2026 The Author(s). Authors retain copyright in their articles.

Dynamic knowledge and knowledge freshness. Community knowledge changes substantially over time, particularly in technical communities where software versions, APIs, toolchains, and best practices are continuously updated. Large language models that rely on fixed training data may produce information that was once correct but is no longer valid. Future systems therefore need an explicit knowledge-refresh pathway: continuously retrieve recent community content and external knowledge sources while using timestamps, version numbers, update histories, and community feedback to distinguish historical information from currently valid knowledge. Compared with retraining an entire language model, updating a retriever or external knowledge base can often be achieved at lower cost, making dynamic retrieval and knowledge-base maintenance important foundations for long-term CQA usability.

Long-tail and dynamic multi-label spaces. Community labels typically exhibit both pronounced long-tail distributions and dynamic evolution. Popular topics have abundant examples, whereas niche technologies, emerging concepts, and new tools may have few examples despite high discriminative value. Conventional classification methods tend to favor frequent labels, resulting in insufficient recall for long-tail labels, while the continual emergence of new labels gradually reduces the adequacy of a fixed label space. Future research should jointly consider label dependencies, long-tail learning, hierarchical organization, and dynamic label expansion. Recent open-world multi-label work suggests that label-space discovery can itself be learned under weak supervision [31]. Large language models can reduce the cost of interpreting new labels, but unconstrained generation may cause boundary drift and inconsistent formatting. A promising direction is therefore to combine open-ended semantic understanding with controlled label systems. Recent open-world multi-label research further suggests that when label-space boundaries are unclear, a system must jointly discover the label space and perform classification rather than merely select from a fixed set [31]. This is highly relevant to community platforms where new concepts continuously emerge and suggests that evaluation should include new-label discovery, few-shot label recognition, and semantic drift of existing labels rather than relying solely on Micro-F1 over a fixed label set.

Multi-document evidence conflicts and source credibility. In real online communities, different people can put forward solutions that don't agree with each other for the same question. Apart from that, answers also vary in when they were posted, how much experience the author has, what feedback the community gave, and how well they've been tested in real use. Because of this, pulling together multiple answers doesn't always give you knowledge that's more trustworthy. The multi-document QA systems we'll build in the future have to account for both how relevant the content is and how reliable the source is. They also have to spot when evidence is in conflict, compare different sources, and be clear about where there's uncertainty. What's more, Ni et al. Found that when retrieving documents, you might not even get all the supporting evidence you need [35]. This shows that when you do multi-document reasoning, you can't just assume the evidence you get is all you need.

Joint optimization of retrieval, reasoning, and generation Current systems often use a step-by-step structure of retriever plus generator, but mistakes can get passed from one part to another. Wrong multi-label guesses can make retrieval less accurate, missing pieces from retrieval keep key evidence out of the context, and contradicting bits of information can mess up both reasoning and creating the final output. Future systems need to treat understanding multi-labels, getting information, checking evidence, and creating output as processes that depend on each other. Big language models are good at working through complex intent and putting different pieces of evidence together, while more specialized retrieval tools, classifiers and checkers can add stable structure and rules. Evidence should also be able to change label predictions; if evidence is missing or conflicts, the system should go find more; and checking individual claims should happen while the output is being generated. All this would create a looped structure instead of just one-way flow. Some recent methods like Self-RAG and CRAG have already started linking retrieval choices, checking evidence, and output generation [36,37], which gives good ideas for how to build closed-loop CQA. But when it comes to real community situations, we still need more work to integrate label feedback, source reliability, and knowledge that changes over time.

Reliability and human – AI collaboration Large language models are really good at cutting down the costs of pulling out information, summarizing things, and making new responses from community knowledge. Even so, they cannot completely take the place of what a human user can judge. Community knowledge often comes from personal experience, and it includes context-specific details as well as contested points that static datasets cannot always capture very well. Putting humans and AI to work together can be a really useful way to deploy these systems: the AI does fast searches, puts candidate evidence together, and writes a first draft of an answer.

© 2026 The Author(s). Authors retain copyright in their articles.

But then human users check the big conclusions for themselves, add extra context that the AI might have missed, and sort out disagreements that the system just can't handle reliably enough. In areas like medicine, law, and other work where mistakes are really dangerous, people need to check more carefully over what the AI produces. This should go along with citing where the information comes from, checking each claim one by one, and pointing out when the AI isn't totally sure about something — all this helps stop wrong information from spreading. When there isn't enough evidence to support an answer, it's also a good thing for reliable systems to either not give an answer at all, or ask a human to double-check it. A research done by Wen et al. Looked into when large language models should not answer questions, and they found that having the right rules in place for when to abstain is one way that could help cut down on hallucinations and make the systems safer to use [44].

6. Research Synthesis and Outlook

The literature suggests that multi-label question understanding and multi-document evidence reasoning should not be treated as independent research directions. Rather, they are interdependent stages in the transition of CQA from answer matching toward complex problem solving. Multi-label prediction addresses what the user needs, multi-document reasoning addresses where the evidence corresponding to those needs can be found, and knowledge grounding addresses why the resulting conclusions should be trusted. At the same time, retrieved evidence may reveal omissions or over-broad interpretations in the initial label prediction. Classification, retrieval, evidence verification, and generation are therefore better understood as a feedback-connected system.

If we look at how technology has changed, CQA has moved on from early lexical retrieval and answer ranking steps to newer stages like neural semantic matching, pretrained Transformer models, and retrieval-augmented generation. Early work in the field mainly looked for ways to find similar questions, while neural models dealt with the problem of how to represent and compare questions and answers. Pretrained language models made contextual understanding and knowledge transfer better, and RAG with large language models went further to solve the problem of how to bring in external knowledge and build new responses from scratch. What matters here is that these different stages worked out different bottlenecks, they didn't just replace one after another. Retrieval makes it easier to get access to candidate evidence, semantic matching makes it easier to tell candidates apart from each other, pretrained encoders make contextual understanding stronger, and generative systems give a way to put all the evidence together. Current CQA systems get more benefits when they combine all of these abilities, instead of just thinking that the latest generative model can completely replace older retrieval or ranking methods.

The biggest gap in current research isn't just smaller model size or weaker parameter ability. It's that there isn't a single consistent optimization method that connects multi-label intent identification, dynamic multi-source retrieval, multi-document evidence checking, and reliable content generation. What most existing studies do is improve one single part at a time: multi-label work centers on structured ways to show what users need, multi-document question answering centers on getting and putting together evidence from different sources, RAG centers on getting access to outside knowledge, and big language model research makes complex reasoning and language generation even better. But even with that, mistakes keep passing between different parts: wrong labels can pull retrieval off the right track, if retrieval misses things that limits how good reasoning can be, and conflicting evidence can change how good the final generation turns out. Future studies need to build real-world common benchmarks that cover multi-label, multi-document and open interactive settings, plus also do layered evaluation starting from label identification and evidence retrieval all the way to source attribution and the final generation. These kinds of benchmarks would help different parts work together in a closed loop: label interpretation, retrieval, evidence checking, and generation. Some recent research also looks at RAG-reasoning systems as a new direction that people can study more [45].

7. Conclusion

Over almost twenty years, research on CQA has changed from just finding words and ranking answers to working on context clues, pulling info from outside sources, reasoning about multiple documents and making new answers yourself. Even though the focus of the research has shifted over time, the main features of community knowledge has stayed pretty much the same: users keep adding new content all the time, the

© 2026 The Author(s). Authors retain copyright in their articles.

language they use is usually casual, info is spread out and might even contradict each other, and knowledge changes as time goes on. What's more important is that the goal of the work has changed too: systems these days don't just need to find an answer that's similar to what someone asked, they more and more have to figure out what a person really wants from combined information, put together evidence from a bunch of different places, check how reliable the evidence is, and then write answers based on that evidence. In this review paper, we looked at these changes from two angles: how technology evolved and what capabilities different systems have. We paid special attention to how understanding questions with multiple labels connects to reasoning about evidence across multiple documents. The biggest problem for CQA research going forward isn't just about getting better at individual models. Instead, we need to build systems that work in closed loops, they can keep updating their knowledge, change how they get info, check evidence from multiple sources, and help make answers that are actually reliable.

References

[1] I. Srba and M. Bielikova, “A Comprehensive Survey and Classification of Approaches for Community Question Answering,” ACM Transactions on the Web, vol. 10, no. 3, Art. 18, pp. 1–63, 2016, doi:10.1145/2934687.

[2] J. Jeon, W. B. Croft, and J. H. Lee, “Finding similar questions in large question and answer archives,” in Proc. 14th ACM CIKM, 2005, pp. 84–90, doi:10.1145/1099554.1099572.

[3] M. Surdeanu, M. Ciaramita, and H. Zaragoza, “Learning to Rank Answers on Large Online QA Collections,” in Proc. ACL-08: HLT, 2008, pp. 719–727.

[4] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree, “SemEval-2015 Task 3: Answer Selection in Community Question Answering,” in Proc. SemEval 2015, 2015, pp. 269–281, doi:10.18653/v1/S15-2047.

[5] A. Severyn and A. Moschitti, “Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks,” in Proc. 38th ACM SIGIR, 2015, pp. 373–382, doi:10.1145/2766462.2767738.

[6] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, “Improved Representation Learning for Question Answer Matching,” in Proc. ACL, 2016, pp. 464–473, doi:10.18653/v1/P16-1044.

[7] X. Zhang, S. Li, L. Sha, and H. Wang, “Attentive Interactive Neural Networks for Answer Selection in Community Question Answering,” in Proc. AAAI, 2017, pp. 3525–3531, doi:10.1609/AAAI.V31I1.11006.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186, doi:10.18653/v1/N19-1423.

[9] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992, doi:10.18653/v1/D19-1410.

[10] Z. Ji et al., “Survey of Hallucination in Natural Language Generation,” ACM Computing Surveys, vol. 55, no. 12, Art. 248, 2023, doi:10.1145/3571730.

[11] P. Manakul, A. Liusie, and M. Gales, “SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models,” in Proc. EMNLP, 2023, pp. 9004–9017, doi:10.18653/v1/2023.emnlp-main.557.

[12] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems 33, 2020.

[13] A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems 30, 2017.

[14] T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems 33, 2020.

© 2026 The Author(s). Authors retain copyright in their articles.

[15] S. Borgeaud et al., “Improving Language Models by Retrieving from Trillions of Tokens,” in Proc. 39th ICML, PMLR 162, 2022, pp. 2206–2240.

[16] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv:2312.10997, 2023.

[17] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, “REALM: Retrieval-Augmented Language Model Pre-Training,” in Proc. ICML, 2020, pp. 3929–3938.

[18] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang, “Generate Rather Than Retrieve: Large Language Models are Strong Context Generators,” in Proc. ICLR, 2023.

[19] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “ReAct: Synergizing Reasoning and Acting in Language Models,” in Proc. ICLR, 2023.

[20] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language Agents with Verbal Reinforcement Learning,” in Advances in Neural Information Processing Systems 36, 2023.

[21] V. Liévin, C. E. Hother, A. G. Motzfeldt, and O. Winther, “Can Large Language Models Reason about Medical Questions?” Patterns, vol. 5, no. 3, Art. 100943, 2024, doi:10.1016/j.patter.2024.100943.

[22] K. Singhal et al., “Large Language Models Encode Clinical Knowledge,” Nature, vol. 620, pp. 172–180, 2023, doi:10.1038/s41586-023-06291-2.

[23] X. Hu, “Enhancing Answer Selection in Community Question Answering with Pre-trained and Large Language Models,” arXiv:2311.17502, 2023.

[24] G. Tsoumakas and I. Katakis, “Multi-Label Classification: An Overview,” International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 1–13, 2007, doi:10.4018/jdwm.2007070101.

[25] M.-L. Zhang and Z.-H. Zhou, “A Review on Multi-Label Learning Algorithms,” IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819–1837, 2014, doi:10.1109/TKDE.2013.39.

[26] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, “Deep Learning for Extreme Multi-label Text Classification,” in Proc. 40th ACM SIGIR, 2017, pp. 115–124, doi:10.1145/3077136.3080834.

[27] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, “SGM: Sequence Generation Model for Multi-label Classification,” in Proc. COLING, 2018, pp. 3915–3926.

[28] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu, “AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification,” in Advances in Neural Information Processing Systems 32, 2019.

[29] L. Xiao, B.-L. Chen, X. Huang, H.-F. Liu, L.-P. Jing, and J. Yu, “Multi-label Text Classification Method Based on Label Semantic Information,” Journal of Software, vol. 31, no. 4, pp. 1079–1089, 2020, doi:10.13328/j.cnki.jos.005923.

[30] B. Li, J. Zhou, X. Jing, and N. Li, “Research on Automatic Multi-Label Text Classification for ICD-9-CM3,” Chinese Journal of Medicine, vol. 22, no. 17, pp. 51–56, 2025 [in Chinese].

[31] X. Li, J. Jiang, R. Dharmani, J. Srinivasa, G. Liu, and J. Shang, “Open-world Multi-label Text Classification with Extremely Weak Supervision,” in Proc. EMNLP, 2024, pp. 15084–15096, doi:10.18653/v1/2024.emnlp-main.841.

[32] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading Wikipedia to Answer Open-Domain Questions,” in Proc. ACL, 2017, pp. 1870–1879, doi:10.18653/v1/P17-1171.

[33] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. Manning, “HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering,” in Proc. EMNLP, 2018, pp. 2369–2380, doi:10.18653/v1/D18-1259.

[34] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense Passage Retrieval for Open-Domain Question Answering,” in Proc. EMNLP, 2020, pp. 6769–6781, doi:10.18653/v1/2020.emnlp-main.550.

© 2026 The Author(s). Authors retain copyright in their articles.

[35] A. Ni, M. Gardner, and P. Dasigi, “Mitigating False-Negative Contexts in Multi-document Question Answering with Retrieval Marginalization,” in Proc. EMNLP, 2021, pp. 6149–6161, doi:10.18653/v1/2021.emnlp-main.497.

[36] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,” in Proc. ICLR, 2024.

[37] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, “Corrective Retrieval Augmented Generation,” arXiv preprint arXiv:2401.15884, 2024.

[38] Y. Li and M. Li, “Supplementary Q&A Recommendation Based on Transfer Learning Enhanced Multi-Label Multi-Document Classifier,” Data Analysis and Knowledge Discovery, vol. 8, no. 10, pp. 89–99, 2024, doi:10.11925/infotech.2096-3467.2023.0683.

[39] Z. Li, B. Zou, Y. Fan, and Y. Hong, “KEPR: Knowledge Enhancement and Plausibility Ranking for Generative Commonsense Question Answering,” in Proc. IEEE IJCNN, 2023, pp. 5163–5170.

[40] P. Bajaj et al., “MS MARCO: A Human Generated MAchine Reading COmprehension Dataset,” arXiv:1611.09268, 2016.

[41] T. Kwiatkowski et al., “Natural Questions: A Benchmark for Question Answering Research,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 452–466, 2019, doi:10.1162/tacl_a_00276.

[42] P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A. A. Freihat, J. Glass, and B. Randeree, “SemEval-2016 Task 3: Community Question Answering,” in Proc. SemEval-2016, 2016, pp. 525–545, doi:10.18653/v1/S16-1083.

[43] P. Nakov et al., “SemEval-2017 Task 3: Community Question Answering,” in Proc. SemEval-2017, 2017, pp. 27–48, doi:10.18653/v1/S17-2003.

[44] B. Wen, J. Yao, S. Feng, C. Xu, Y. Tsvetkov, B. Howe, and L. L. Wang, “Know Your Limits: A Survey of Abstention in Large Language Models,” Transactions of the Association for Computational Linguistics, vol. 13, pp. 529–556, 2025, doi:10.1162/tacl_a_00754.

[45] Y. Li et al., “A Survey of RAG-Reasoning Systems in Large Language Models,” in Findings of EMNLP, 2025, pp. 12120–12145, doi:10.18653/v1/2025.findings-emnlp.648.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights

Copyright for this article is retained by the author (s), with first publication rights granted to the journal. This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).

© 2026 The Author(s). Authors retain copyright in their articles.