International Journal of Agricultural and Environmental Information Systems
Volume 16 • Issue 1 • January-December 2025

Search Engine for E-Books Containing Questions and Answers About Agriculture

Glauber José Vaz
https://orcid.org/0000-0002-4527-5150
Embrapa Digital Agriculture, Brazil

ABSTRACT

This article presents a search engine application programming interface (API) for e-books structured in questions and answers about agriculture and related topics. It can answer questions made in natural language or with keywords. Its methodology includes the treatment of e-books, parsing with different analyzers, indexing standard files, and exploring search templates using Elasticsearch and the BM25 algorithm. It also involves providing the search engine API through an API platform. All API resources are presented in detail, along with a user interface that uses them for a specific technical assistance and rural extension application. The dataset with treated text extracted from the agricultural books considered in the present work is freely accessible, and the API is available to partners. Based on qualified information, this digital solution is relevant to disseminating technical information, especially to small farmers and extension workers, and contributes to increasing knowledge, productivity, and sustainability in agriculture.

KEYWORDS

API, Brazil, Digital Agriculture, Elasticsearch, Information Dissemination, Information Retrieval, Question-Answering Systems, Technical Assistance and Rural Extension, Vertical Search Engine

INTRODUCTION

Digitalization has been a driving force in the evolution of agricultural knowledge and innovation systems. Despite its impact on culture and farmers' identities and work, digitalization in agriculture is expected to provide technical optimization of production systems, value chains, and food systems, as well as enhance knowledge exchange and learning (Klerkx et al., 2019). Digital technologies have a high potential impact on food and agriculture, empowering rural households and inspiring entrepreneurship, and they are already improving smallholders' access to information and productivity (Food and Agriculture Organization of the United Nations [FAO], 2022).

Digital agriculture involves information and communication technologies (ICTs) that support rural producers in planning, monitoring, and managing the activities of the production system (Bolfe et al., 2020). Janc et al. (2019) argued that the diffusion of ICTs has lowered the costs of producing and disseminating information and that the internet promotes better access to information, making it possible to accumulate, exchange, and share farming innovations to be introduced on farms, especially given the limited mobility of the farming population. However, limited digital competencies represent obstacles to smart agriculture.

Inwood and Dale (2019) noted that the lack of mobile devices or web-based applications for providing or exchanging information represents a barrier to using digital tools by agriculture actors

DOI: 10.4018/IJAEIS.376171

seeking to learn about practices that improve sustainability. Kenny and Regan (2021) demonstrated the potential for increasing the adoption and usage of smartphones by Irish farmers. Daum et al. (2018) also discussed the potential of using smartphone apps in rural areas of low-income countries, focusing on using smartphones as a research tool in rural areas.

According to Rotz et al. (2019), there is an emerging trend for technologies in digital agriculture to exclude small farmers from participating in agri-food production. The authors argue that many technological solutions empower corporate actors rather than supporting independent farmers to make informed decisions. However, big data technologies could support large-scale, agroecological, and small-scale farmers.

Indeed, Van Campenhout (2017) noted that ICTs can facilitate access to agricultural knowledge by presenting information online and increasing yields due to technology adoption. The author explored the potential for ICTs to provide agricultural information and extension services to smallholder farmers in Uganda. In a later study in the country, Van Campenhout et al. (2021) evaluated the effectiveness of delivering agricultural information to small-scale maize farmers using an ICT-mediated extension approach. This involved short videos, SMS messages, and an interactive voice response system in which a farmer listens to recorded information through phone calls. Videos proved to be the most effective at raising awareness of advances such as improved maize cultivation. The author encourages the development of new approaches using ICTs in agricultural extension systems to overcome some problems, such as the low cost-effectiveness and limited scalability of extension efforts.

In Brazil, Bolfe et al. (2020) used an online consultation to interview hundreds of farmers about digital technologies. They found that 84% used at least one digital technology in their production system, and more than 60% used it to obtain general information. The main perceived benefit was increased productivity, and connectivity was one of the main concerns. Based on 11 case studies carried out in different countries, Šūmane et al. (2018) examined the diversity of knowledge sources and learning forms that farmers use to meet their knowledge needs. The authors cited the knowledge accumulated from their practical experience, other farms, market actors, public administrations and regulatory institutions, and formal agricultural institutions. They show the importance of integrating different knowledge sources into modern agriculture.

Books are good sources of information, but they have limited reach. Digital solutions based on book content can provide greater access to information in a contextualized way. The Brazilian Agricultural Research Corporation (Embrapa) is the leading research company in the world on tropical agriculture. It was established to develop technologies for tropical agriculture and animal farming and produces a considerable amount of technical and scientific documentation, including digital books (e-books). One collection of these books is called "500 Questions, 500 Answers". It consists of dozens of books written in Portuguese that contain Embrapa's answers to questions made by farmers, cooperatives, and other stakeholders. The collection is available online at no cost (https://mais500p500r.sct.embrapa.br/view/index.php). Each book deals with a different topic, such as soy, corn, dairy cattle, beef cattle, and integrated crop–livestock–forest (ICLF; in Portuguese, *integração lavoura-pecuária-floresta*) systems. This collection is one of the most accessed documents in Embrapa's online library by small-scale farmers.

Expanding the reach of digital technologies, then, is a promising avenue for increasing the cost-effectiveness and scalability of agricultural extension. For example, given the hurdles of connectivity and digital competencies, digital solutions can be packaged in smartphone applications and web systems to benefit both large-scale and small-scale farmers. In particular, Brazil offers many opportunities to explore solutions of this kind, given the existing interest of farmers and farmers' increasing connectivity and use of smartphone applications. Search engines and question-answering (QA) systems are good examples of applications that offer excellent potential.

Therefore, access to specialized information for farmers, especially small ones, is challenging in many countries, including Brazil. In this regard, digital agriculture allows for cost-effective and scalable dissemination of information, providing tools such as QA systems and search engines. One of the

main challenges faced by small farmers and extension workers is finding organized and reliable information on specific topics. Less than 20% of Brazilian family farmers report having access to Technical Assistance and Rural Extension (Ater) services, which aim to improve the income and quality of life of rural families. To address this gap, the Ministry of Agriculture and Livestock created Ater Digital, a public policy designed to promote the use of information and communication technologies in Ater initiatives.

The Ater+ Digital platform (https://www.atermaisdigital.cnptia.embrapa.br/) is one of these initiatives. It provides information organized into hubs dedicated to topics such as climate change, cashew, beans, cowpea, poultry, swine, and ICLF systems, among others. Some of these hubs are related to the books in the "500 Questions, 500 Answers" collection.

This work aims to provide a search engine capable of answering questions made in natural language or with keywords about specific agriculture topics related to the books available in Embrapa's "500 Questions, 500 Answers" collection. The tool uses an application programming interface (API), an essential component in enabling digitalization (Pillai et al., 2021). This makes it possible to embed the search engine in solutions developed by various institutions and to facilitate the integration of this rich content into other knowledge sources. In this study, the use of the API is demonstrated through Ater+ Digital.

SEARCH ENGINES AND QUESTION-ANSWERING SYSTEMS IN AGRICULTURE

While search engines return relevant documents for particular keywords, QA systems return answers (Allam & Haggag, 2012). QA systems can respond to questions asked in natural language, making it easier for people to access information (Uttarwar et al., 2019), and they have been increasingly used to obtain knowledge and solve problems, including in the modern agricultural field (Xiong et al., 2018). Allam and Haggag (2012) provided an overview of the different components of QA system architecture, including question classification, information retrieval, and answer extraction. The present study involves books organized into questions and answers. Each question–answer pair can be considered a document. Therefore, the system that contains these documents is simpler than general QA systems since it does not need to consider question classification or answer extraction phases but can instead focus solely on information retrieval. It is essentially a search engine capable of responding to a set of questions asked in natural language.

Azizan et al. (2018) conducted an experiment to evaluate the effectiveness of popular commercial search engines in searching domain-specific information—in their case, relating to fruits. They concluded that the ability of these search engines to find domain-specific information remains unsatisfactory. This highlights the importance of domain-specific or vertical search engines, which provide search results for a particular field (Yao, 2017). Previous studies have noted the practical value of research into vertical search engines in the agricultural field (Ding, 2016).

There are many examples of systems that allow queries about agricultural information. Al Manir et al. (2018) described a service that helped agricultural consultants identify optimal crop varieties in a use case relating to farmers deciding which variety of eggplant to plant. Ingram and Gaskell (2019) proposed a search engine in the field of agriculture and forestry with an ontology at its core, exploring semantic technologies. Gaikwad et al. (2015) developed a system to answer factoid questions such as *which*, *what*, *who*, and *where* questions about agriculture. Niranjan et al. (2019) surveyed QA systems and argued that very few provide correct and efficient answers, especially in the agriculture domain. Thus, there are good opportunities for research and development in this area.

Closed-domain QA systems consider only a specific topic since they are restricted to a particular set of information sources and limit the questions that can be asked (Uttarwar et al., 2019; Salunkhe, 2020). This study deals with a closed-domain system based on the content of the books in an agricultural collection. Damiano et al. (2016) proposed a QA framework for closed domains to provide factual and self-contained information extracted from documents as answers to users'

questions. They described a generic pipeline with three phases: question analysis, answer extraction, and answer selection. The first phase produces a set of keywords from the question, the second uses it in a search engine to find relevant documents, and the third returns the final answer from these documents. They proposed a framework following this pipeline that consists of four modules: indexing, question processing, information retrieval, and answer processing. Compared with this framework, the system presented in this paper is simpler because it focuses on answer extraction and information retrieval since the documents are already arranged as questions and answers.

Devi and Dua (2017) presented a QA system in the agriculture field that combines natural language processing (NLP) and semantic web technologies, exploring ontology and the semantics of the question rather than the syntax. Diefenbach (2018) also considered the semantic web for QA systems using knowledge bases, using a triple-encoding information set. However, in the present study, these technologies are unnecessary for question processing, as the questions are already mapped to precise answers. They also involve obstacles in an operational context, as Ingram and Gaskell (2019) verified using a user-centered ontology in a search engine.

This work can also be considered a way of linking good science to good practice in agriculture. According to Top and Wigham (2015), this can be done through advice given by extension workers but also with products, services, and smart tools, such as web-based QA systems, because these can facilitate the dissemination of information in a manner more suited to the modern agricultural sector. As examples, the authors cite the QA system ask-Valerie and API-AGRO, a platform provided by several agricultural institutes to give access to agricultural data and services. They also highlight the importance of jumping from research prototypes to actual products, representing a step from the world of research to agricultural practice.

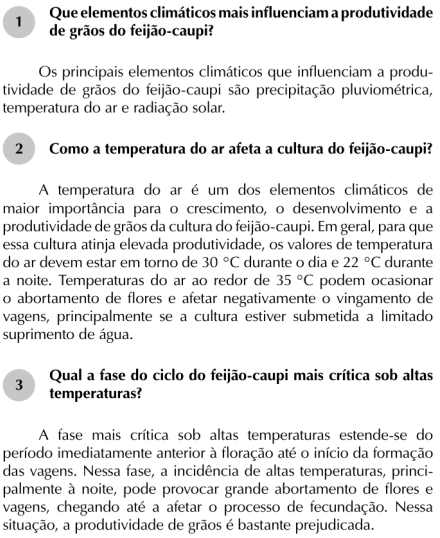
Tools like the digital expert-assistant ask-Valerie (Willems et al., 2015) promise to make scientific results accessible for practical use by end-users, but their success cannot depend on intense work by experts, as in this case. Platforms like API-AGRO (Siné et al., 2015) are also important for facilitating the distribution of datasets and services related to agriculture and software reuse. Such API management solutions are critical for accelerating the creation of dynamic digital ecosystems, achieving operational excellence, and optimizing customer experience (Heffner et al., 2020). These considerations led to the decision to create an API in this study and make it available on an API platform so that different applications, such as web systems, apps, and other APIs, can use the same API.

Therefore, very few QA systems provide correct and efficient answers. None of the cited studies consider the content of e-books with previously structured questions and answers. This simplifies developing the QA system as a domain-specific search engine and enables the sharing of better-quality information since it always returns the correct answer according to the indexed books. This paper presents an information retrieval system for the content of a collection of books containing questions and answers about agriculture and related topics. This API-accessible solution is important for disseminating technical information and can be embedded in different systems.

METHOD

The books used in this study are from Embrapa's "500 Questions, 500 Answers" collection. They were selected based on the needs of the Ater+ Digital initiative. Their structure is the same for all of them, with questions numbered from one to 500, each followed by its own answer. They are available in Portable Document Format and ePub formats, the latter being a distribution and interchange format widely adopted for e-books. It provides a means of representing, packaging, and encoding web content, including hypertext markup language (HTML), cascading style sheets, and other resources, for distribution in a single-file container (Garrish & Cramer, 2019). Figure 1 shows the first three questions and answers from the book about cowpeas (Cardoso et al., 2017).

Figure 1. Questions and answers about cowpeas



1 **Que elementos climáticos mais influenciam a produtividade de grãos do feijão-caupi?**

Os principais elementos climáticos que influenciam a produtividade de grãos do feijão-caupi são precipitação pluviométrica, temperatura do ar e radiação solar.

2 **Como a temperatura do ar afeta a cultura do feijão-caupi?**

A temperatura do ar é um dos elementos climáticos de maior importância para o crescimento, o desenvolvimento e a produtividade de grãos da cultura do feijão-caupi. Em geral, para que essa cultura atinja elevada produtividade, os valores de temperatura do ar devem estar em torno de 30 °C durante o dia e 22 °C durante a noite. Temperaturas do ar ao redor de 35 °C podem ocasionar o abortamento de flores e afetar negativamente o vingamento de vagens, principalmente se a cultura estiver submetida a limitado suprimento de água.

3 **Qual a fase do ciclo do feijão-caupi mais crítica sob altas temperaturas?**

A fase mais crítica sob altas temperaturas estende-se do período imediatamente anterior à floração até o início da formação das vagens. Nessa fase, a incidência de altas temperaturas, principalmente à noite, pode provocar grande abortamento de flores e vagens, chegando até a afetar o processo de fecundação. Nessa situação, a produtividade de grãos é bastante prejudicada.

In order to index the e-books in the system, the treated files must be generated through *digital curation*, defined as the active management and enhancement of digital information assets for current and future use (National Research Council of the National Academies, 2015). In this process, each book is analyzed and edited to be transformed into a single HTML file with an appropriate header and the essential elements of the book, including the images, tables, and references. The set of HTML files can be used by other applications in the future.

The e-book is first transformed to HTML format, and all the cascading style sheets elements are removed when unnecessary or replaced by HTML tags. Tables are converted to base64 format to make it possible for their direct inclusion in text format in the file. The file's header has metadata providing the title of the book and an identifier for it, links to digital versions of the book in Portable Document Format and ePub formats, the year of publication, and additional information about the authors, editors, and curators. Other kinds of data can be included. In the body of the HTML file, the questions are followed by their answers and grouped by chapters, which are identified by *<h1>* tags. Each question is embraced by *<p>* tags with class *pergunta* and starts with its number followed by a parenthesis. The answers include all the text between the question and the paragraph assigned the *separador* class, the content of which is ●●●. Table 1 shows an HTML file involving the header and one question–answer pair from the book on cowpeas (Cardoso et al., 2017). This was prepared from the original e-book.

Table 1. Example hypertext markup language (HTML) file

```
<html>
<head>
<meta charset="UTF-8">
<meta name="identifier" content="feijao-caupi">
<meta name="pdf" content="http://ainfo.cnptia.embrapa.br/digital/bitstream/item/166168/1/500P500R-Feijao-caupi
.pdf">
<meta name="epub" content="http://ainfo.cnptia.embrapa.br/digital/bitstream/item/166086/1/-files-500p500r-feijao
-caupi.epub">
<meta name="year" content="2017">
<meta name="author" content="Embrapa">
<meta name="editor" content="Editor 1">
<meta name="editor" content="Editor 2">
<meta name="editor" content="Editor 3">
<meta name="editor" content="Editor 4">
<meta name="curator" content="Curator">
<title>Feijão-caupi</title>
</head>
<body>
<h1>Ecofisiologia</h1>
<p class="pergunta" id="p1">1) Que elementos climáticos mais influenciam a produtividade de grãos do
feijão-caupi?</p>
<p>Os principais elementos climáticos que influenciam a produtividade de grãos do feijão-caupi são precipitação
pluviométrica, temperatura do ar e radiação solar.</p>
<p class="separador">●●●</p>
</body>
</html>
```

The accepted content for the HTML files can be specified by a grammar. According to Aho et al. (2007), a grammar specifies the syntax of a language, and it has four components: the terminals, elementary symbols of the language defined by the grammar; the nonterminals, which represent a set of strings of terminals; the rules of production; and the initial symbol. Table 2 presents the production rules for the grammar used in this study. The initial symbol is *ebook*. A particular representation was used to specify the grammar, but standard conventions were explored. It also uses the notation from Cameron (1993) for syntactic constructs in which the order of constituent elements is unimportant. The notation $<< e_1 \| e_2 \| \ldots \| e_n >>$ denotes any permutation of the elements e_i. The words in italics are nonterminals and strings in grey delimited by single quotes are terminals. However, in this representation, to make it simpler, some nonterminals, represented by underlined words, are embedded within terminals, and rules are not derived for them. These underlined nonterminals correspond exactly to the content for indexing, but in this work, only the bold ones are associated with indexed content, while the others are not important for the application and are simply used for documenting the data.

Table 2. Grammar for the hypertext markup language (HTML) file

ebook → '<html>' *head body* '</html>'

head → '<head>' '<meta charset="UTF-8">' *metadata title* '</head>'

metadata → << *identifier* || [*pdf*] || [*epub*] || *year* || *author** || *editor** || *curator*$^{+}$ || *others** >>

identifier → '<meta name="identifier" content="***book_id***">'

pdf → '<meta name="pdf" content="***pdf***">'

epub → '<meta name="epub" content="***epub***">'

year → '<meta name="year" content="***year***">'

author → '<meta name="author" content="***author***">'

editor → '<meta name="editor" content="***editor***">'

curator → '<meta name="curator" content="***curator***">'

others → '<meta name="***meta_name***" content="***meta_content***">'

title → '<title>' ***book*** '</title>'

body → '<body>' *qa_chapter*$^{+}$ '</body>'

qa_chapter → '<h1>' ***chapter*** '</h1>' *qa*$^{+}$

qa → *question* ***answer*** ' ••• '

question → ' ' ***question_number*** ')' ***question*** ' '

Parsing is the process of analyzing a string of words to uncover its phase structure according to the rules of a grammar, starting from the initial symbol (Russell & Norvig, 2010). Since the content is prepared according to the grammar in Table 2, as in the example of Table 1, it is necessary to structure its content in an appropriate format for indexing in Elasticsearch. Elasticsearch is a distributed search and analytics engine that provides a fast search for all data types. It includes a bulk API that can perform, in a single request, multiple actions that are specified using a *Newline Delimited JavaScript Object Notation* (NDJSON; Elastic, 2022). Each JavaScript Object Notation (JSON) text must conform to the JSON specification and be followed by the newline character. JSON is a lightweight, text-based, language-independent data interchange format, which defines a small set of formatting rules for the serialization of structured data (Bray, 2014), and NDJSON is a convenient format for storing structured data to be processed one record at a time (NDJSON, 2014). For index operations, for example, the request body must contain a newline-delimited list of index actions and their associated source data.

Table 3 shows the result, in NDJSON format, of parsing the first question of the HTML file from the book on cowpeas. The first object brings the record identifier *feijao-caupi_001*, and the next presents the document. The underlined and bold nonterminals of the grammar in Table 2, such as *question_number*, *question*, and *answer*, correspond to the properties of the object representing the document. The identifier is obtained by concatenating the book's code in *book_id* to the question number formatted in three characters linked by the _ character.

Table 3. Example of input for indexing a record with elasticsearch bulk application programming interface

{**"index"**:{**"_id"**: "feijao-caupi_001"}}
{**"question_number"**: 1, **"question"**: "Que elementos climáticos mais influenciam a produtividade de grãos do feijão-caupi?", **"answer"**: " Os principais elementos climáticos que influenciam a produtividade de grãos do feijão-caupi são precipitação pluviométrica, temperatura do ar e radiação solar. ", **"chapter"**: "Ecofisiologia", **"book"**: "Feijão-caupi", **"book_id"**: "feijao-caupi", **"epub"**: "http://ainfo.cnptia.embrapa.br/digital/bitstream/item/166086/1/-files-500p500r-feijao-caupi.epub",**"pdf"**: "http://ainfo.cnptia.embrapa.br/digital/bitstream/item/166168/1/500P500R-Feijao-caupi.pdf", **"year"**: 2017}

Since the content is structured in an appropriate format, it is indexed for future queries. The results of the queries should return the corresponding text provided by the HTML files, which facilitates the development of user interfaces based on web technologies. However, indexing and searching require text analysis. In Elasticsearch, this is done with analyzers, which are compounded by zero or more character filters, one tokenizer, and zero or more token filters. The first components receive the text and transform, add, remove, or change characters. The second one breaks the text into tokens, transforming a stream of characters into a stream of tokens. Finally, token filters transform the tokens by adding, removing, or changing the tokens (Elastic, 2022).

Table 4 shows an example of a simple analyzer processing a string given by the *text* field. The character filter *html_strip* removes HTML elements such as <*p*> or <*i*> tags. The *standard* tokenizer breaks the text into words, considering punctuation and whitespaces as separators. The *asciifolding* token filter performs character substitution for the ASCII equivalent, such as *ã* for *a*, *é* for *e*, and *ç* for *c*. Finally, the *lowercase* filter uses lowercase for all letters.

Table 4. Example of an analyzer for elasticsearch

```
GET /_analyze
{
"char_filter": [ "html_strip" ],
"tokenizer": "standard",
"filter": [ "asciifolding", "lowercase" ],
"text": "<p class=\"pergunta\" id=\"p113\">113) Atualmente, qual é a forrageira mais utilizada e quais são aquelas que apresentam potencial para que sejam recomendadas para o pasto de safrinha na região Centro-Oeste do Brasil?</p>"
}
```

Thus, the application of the analyzer to the "text" property in Table 4 produces the following list of tokens: ["113", "atualmente", "qual", "e", "a", "forrageira", "mais", "utilizada", "e", "quais", "sao", "aquelas", "que", "apresentam", "potencial", "para", "que", "sejam", "recomendadas", "para", "o", "pasto", "de", "safrinha", "na", "regiao", "centro", "oeste", "do", "brasil"]. If the input was replaced by " Atualmente, a U. ruziziensis é a forrageira mais utilizada para formação do pasto de safrinha no Centro-Oeste brasileiro. ", the output would be ["atualmente", "a", "u",

"ruziziensis", "e", "a", "forrageira", "mais", "utilizada", "para", "formacao", "do", "pasto", "de", "safrinha", "no", "centro", "oeste", "brasileiro"].

This type of text analysis must be conducted in the indexing and search phases. For instance, a query for "forrageira para a safrinha Centro-Oeste" is processed and generates the token list ["forrageira", "para", "a", "safrinha", "centro", "oeste"]. If a document containing the text in Table 4 is indexed, it is returned as a result of this query bringing the complete text, including HTML tags, even though they were ignored in the indexing and searching phases. Therefore, analyzers based on this idea make it possible to properly index the content of the books provided in HTML files. Although it is possible to ignore elements in analysis phases that would cause distortions in the search, such as HTML tags, this organization facilitates the development of user interfaces to present the content in a representation similar to the original e-books.

The same text is processed by different analyzers in addition to that presented in Table 4: *stem_analyzer*, *exact_analyzer*, and *ngram_analyzer*. Each generates variations of the same word or term. The analyzer *stem_analyzer* tries to get the stems of the words, considering the Portuguese language. The output of *exact_analyzer* consists of the words of a text exactly how they are written, and *ngram_analyzer* generates subwords of lengths from three to 30 characters. For instance, the text "Forrageiras utilizadas no Brasil", when processed by *stem_analyzer*, *exact_analyzer*, and *ngram_analyzer*, respectively, generates the outputs ["forrageir", "utiliz", "no", "brasil"], ["Forrageiras", "utilizadas", "no", "Brasil"], and ["for", "forr", "forra", "forrag", "forrage", "forragei", "forrageir", "forrageira", "forrageiras", "uti", "util", "utili", "utiliz", "utiliza", "utilizad", "utilizada", "utilizadas", "bra", "bras", "brasi", "brasil"].

Different weights are assigned to each processing type. The output of *exact_analyzer* has greater weight than the standard analyzer, similar to the one provided by Table 4. Both weigh more than stemmed words, which have greater weight than the generated terms by *ngram_analyzer*. These analyzers and weights help to increase precision for the first positions of the ranking, placing more relevant documents at the top of the search results since they value the similarity with the query more. However, they also allow the retrieval of documents containing similar words, especially with the same stem or subword. Irrelevant documents can be retrieved, but likely later than the relevant ones.

This work uses BM25, the default Elasticsearch algorithm, to rank the results (Robertson & Zaragoza, 2009). Kamphuis et al. (2020) explored the nuances of different variants of this well-known scoring function for document retrieval and concluded that the differences among these variations do not impact retrieval effectiveness. BM25 computes the relevance of the query terms appearing in each document, using simple statistical measures such as term frequencies, document frequencies, and document length. Even so, it provides competitive performance compared to modern approaches to text ranking tasks (Rosa et al., 2021).

The challenge is to get the best-indexed question–answer pair from the user input. After selecting the question, the answer provided is correct because the precise answers are already mapped to the questions. Each part of the text can be analyzed in a specific way. This is determined by a mapping configuration in Elasticsearch, and the indexing analysis can be different from the search. It is important to note that search templates were used. A search template is a stored search that can be run with different values for its variables. This makes it possible to run searches without exposing the query syntax of Elasticsearch, making it simpler. It is also possible to change searches without modifying the application code and to have many templates, each with a different identifier (Elastic, 2022).

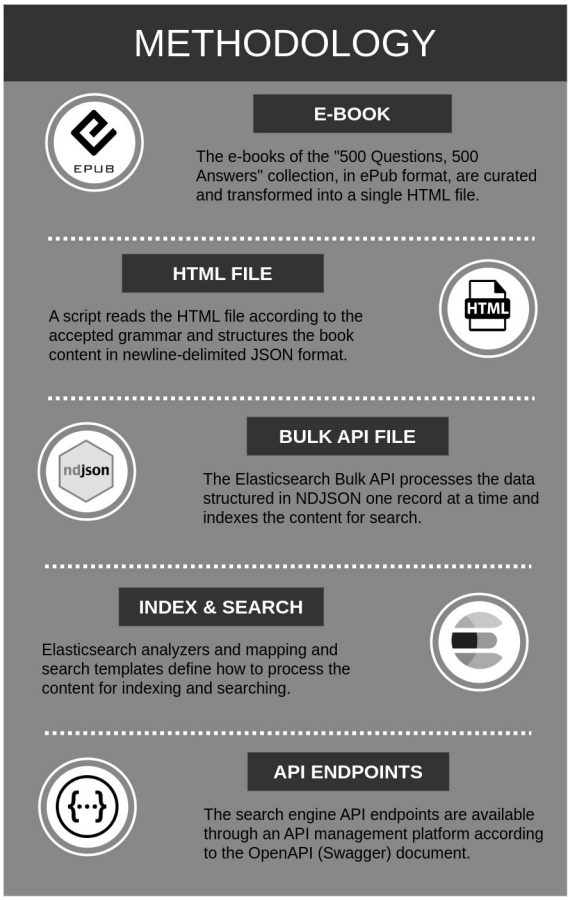
Elasticsearch provides many endpoints through its API, but only a few are helpful for this application. Therefore, the API was made available through an API management technology that supports the different stages of the API life cycle, from the planning to the retirement of APIs, including design, publication, consumption, versioning, and others (Pillai et al., 2021). An API can be described according to the OpenAPI specification, formerly the Swagger specification.

In summary, the methodology of this work is presented in Figure 2. The e-book is treated to be available in a single HTML file with all the relevant content. A script processes it to generate a proper file for the

Elasticsearch Bulk API in NDJSON format. The HTML and NDJSON files for the processed books, along with the script, are available in Embrapa's Research Data Repository (Redape; Campos et al., 2022). The documents are indexed using suitable analyzers and mapping of Elasticsearch technology. Specific templates for search are also built, and an OpenAPI (or Swagger) document specifies the available API resources. This methodology is described in detail by Vaz et al. (2023).

The main challenges encountered were related to the extensive curation activities performed on the books. These activities involved manipulating the files to improve the organization and standardization of HTML elements, as well as adjusting content to enhance its usability in digital solutions. While reading in books typically occurs sequentially, in digital solutions, the content of a question can be accessed independently, requiring it to be complete and not reliant on other questions.

Figure 2. Methodology for building the search engine application programming interface based on e-book content



The Search Engine Application Programming Interface

This paper introduces the Responde Agro API, designed to enable the development of a search engine for the content of books in Embrapa's "500 Questions, 500 Answers" collection. The available API resources are accessed through only two endpoints, represented by the paths */_doc/{id}* and */_search/template*. They enable access to a specific question from a book and to the list of the identifiers of all the books indexed in the search engine, queries of a book or all indexed books, and

autocomplete resources for a specific book or all books. Each is explained in detail, and an example of the user interface is presented.

Access to a Specific Question From a Book

It is possible to access a question from a book using the question number and the identifier for the book. Table 5 shows the request for the fifth question from the book about cowpeas (*feijão-caupi* in Portuguese) and its result. In this example, *respondeagro* is the name of the index, *_doc* is the endpoint for accessing the specific document, and *feijao-caupi_005* is the document identifier. Its content is returned in fields *question_number*, *question*, *answer*, and so on. These are the same properties as in Table 3.

Table 5. Access to a specific question from a book

Request:
GET /respondeagro/v1/_doc/feijao-caupi_005
Output:
{
“_index”: “responde_agro”,
“_id”: “feijao-caupi_005”,
“_version”: 1,
“_seq_no”: 504,
“_primary_term”: 1,
“found”: true,
“_source”: {
“question_number”: 5,
“question”: “É possível maximizar a produtividade de grãos na cultura do feijão-caupi em condições de altas temperaturas?”,
“answer”: “ Sim. Para tanto, devem ser utilizadas variedades tolerantes a altas temperaturas. As plantas devem ser distribuídas adequadamente na área e supridas com nutrientes e água, em quantidade adequada para atender a maiores taxas de crescimento. Em geral, deve-se reduzir o número de plantas por unidade de área, para diminuir o autossombreamento na cultura. “,
“chapter”: “Ecofisiologia”,
“book”: “Feijão-caupi”,
“book_id”: “feijao-caupi”,
“epub”: “https://ainfo.cnptia.embrapa.br/digital/bitstream/item/166086/1/-files-500p500r-feijao-caupi.epub”,
“pdf”: “https://ainfo.cnptia.embrapa.br/digital/bitstream/item/166168/1/500P500R-Feijao-caupi.pdf”,
“year”: 2017
}
}

Access to the List of Indexed Book Identifiers

It is possible to access the list of identifiers of all the books indexed in the search engine. Table 6 shows the request for the list and its output when only the books about cowpeas and ICLF are indexed. It is shown that all the 500 documents for the indexed books are available. This resource is provided by a search template whose identifier is *book_ids*.

Table 6. Access to the list of book identifiers

```
Request:
POST /respondeagro/v1/_search/template
{
“id”: “book_ids”
}
Output:
{
“aggregations”: {
“book_ids”: {
“buckets”: [
{
“key”: “feijao-caupi”,
“doc_count”: 500
},
{
“key”: “ilpf”,
“doc_count”: 500
}
]}}
}
```

Queries in the Books

Queries can be done for just one book, as in Table 7, or all indexed books, as in Table 8. The difference is that in the first case, it is necessary to provide the book identifier and the template identifier *query_one_book*. To search in all the books, the template identifier must be *query_all*. In both cases, the query string is mandatory, and there are two fields for pagination: *size*, to determine how many results can be returned at most, and *from*, which sets the first document to be returned in the ranking of the results.

Table 7. Query a specific book

```
POST /respondeagro/v1/_search/template
{
“id”: “query_one_book”,
“params”: {
“query_string”: “produção”,
“book_id”: “feijao-caupi”,
“from”: 0,
“size”: 2
}
}
```

Table 8. Query all books

```
POST /respondeagro/v1/_search/template
{
“id”: “query_all”,
“params”: {
“query_string”: “produção”,
“from”: 0,
“size”: 5
}
}
```

Table 9 shows the output for the request in Table 7. It brings information about the number of hits, i.e., documents meeting the query, the maximum score of the hits, and a list of documents matching the query. Each document has an identifier, a score that measures its relevance to the query, and a *_source* field with the document data similar to the one shown in Table 5. In addition, it contains a "highlight" field with two properties, one for the question and one for the answer, but in these cases, the snippets of text that match the query string are placed between *span* tags of class *highlight*, so the user interface can be developed to highlight these snippets related to the keywords entered as input to the query. The output format for the request in Table 8 is similar to this.

Table 9. Output of a query


```
{
"hits": {
"total": {
"value": 85
},
"max_score": 38.080486,
"hits": [
{
"_index": "responde_agro",
"_id": "feijao-caupi_049",
"_score": 38.080486,
"_source": {...},
"highlight": {
"question": ["Quantas vistorias devem ser feitas nos campos de <span
class='highlight'>produção</span> de semente?"],
"answer": ["<p>Devem ser feitas, no mínimo, duas vistorias(obrigatórias)
no campo de <span class='highlight'>produção</span>. Elas
deverão ser feitas pelo responsável técnico do produtor ou do
certificador, nas fases de floração e de pré-colheita.</p>"]
}
},
{
"_index": "responde_agro",
"_id": "feijao-caupi_060",
"_score": 37.76433,
"_source": {...},
"highlight": {
"question": ["Além dos parâmetros de campo, há outros indicados para a
<span class='highlight'>produção</span> de sementes?"],
"answer": ["<p>Sim. A pureza varietal e a germinação das sementes são parâmetros que devem ser considerados para o
estabelecimento dos campos de <span class='highlight'>produção</span> de sementes.</p>"] } }
] } }
```


Autocomplete and Suggestions

The autocomplete functionalities work as queries but with some particularities. Table 10 shows the autocomplete request when a user enters the string "cultiv" in a query involving the ICLF book (Cordeiro et al., 2015). The template identifier is *autocomplete_one_book*. To use this resource involving all the books, the field *book_id* is not used, and the template identifier must be *autocomplete_all*. The pagination parameters are unnecessary for autocomplete because only the first results are normally used.

Table 10. Autocomplete for one book

```
POST /respondeagro/v1/_search/template
{
“id”: “autocomplete_one_book”,
“params”: {
“book_id”: “ilpf”,
“query_string”: “cultiv”
}
}
```

Table 11 shows the output for the request in Table 10. It provides information about the total documents retrieved, the maximum score obtained, a list of documents that satisfy the query, and suggestions of terms similar to the query being entered. Snippets are omitted and replaced by … for simplification. The *_source* property has the same structure as indicated in Table 5. The question texts are the most relevant for autocomplete since they should be exhibited as alternatives for the query input field. In some cases, no question may be returned. Thus, it might be interesting to show term suggestions for helping the users with their queries. That is why, depending on the input text and the content of both question and answer fields, terms are suggested and returned, respectively, in the properties *sug-question-exact* and *sug-answer-exact*. In Table 11, the terms *cultivo*, *cultiva*, and *cultivá* are suggested for the input “*cultiv*.” They are ranked according to their frequencies in the index. Up to three changes are accepted between the term in the query and the suggested ones.

Table 11. Autocomplete result for input “cultiv”

```
{ “hits”: {
 “total”: { “value”: 160 },
 “max_score”: 13.913353,
 “hits”: [ ...,
 { “_index”: “responde_agro”,
 “_id”: “ilpf_400”,
 “_score”: 13.913353,
 “_source”: {
 “question_number”: 400,
 “question”: “A recuperação de uma pastagem degradada por meio do uso cultivos anuais possibilita a substituição da mesma por outra espécie ou cultivar?”, ... } }, ... ] },
 “suggest”: {
 “sug-answer-exact”: [ ... ],
 “sug-question-exact”: [
 { “text”: “cultiv”,
 “offset”: 0,
 “length”: 6,
 “options”: [
 { “text”: “cultivo”,
 “score”: 0.8333333,
 “freq”: 354 },
 { “text”: “cultiva”,
 “score”: 0.8333333,
 “freq”: 2 },
 { “text”: “cultivá”,
 “score”: 0.8333333,
 “freq”: 1 } ] } ] }
}
```

User Interface

Figure 3 shows a user interface that explores the resources of the Responde Agro API. It is embedded on the ICLF hub site of Ater+ Digital (https://www.atermaisdigital.cnptia.embrapa.br/web/ilpf/perguntas-e-respostas). It contains an input field where the user has entered "sistema agrossilvipastoril." This query returned 13 documents, but Figure 3 presents only the first two. Both are included in the chapter about "Concepts and Modalities of the ICLF Strategy" ("Conceitos e Modalidades da Estratégia de Integração Lavoura-Pecuária-Floresta," in Portuguese), and they are the 11th and sixth questions of the book, whose publication year (*ano* in Portuguese) is 2015. A link to the ICLF book is also provided on the right. The same tool is provided by other hubs, such as beans, cashew, and cowpea. Therefore, the Responde Agro API assists users in finding organized and reliable information on specific topics.

Figure 3. User interface based on the application programming interface

DISCUSSION

Many examples of QA systems have been provided, but none specifically work with the content of e-books that have been structured as a series of questions and answers. This organization of the documents makes it possible to develop the system as a search engine. The system developed in this study attempts to retrieve the question the user wants and shows the

corresponding answer. It includes resources for finding the complete question and is based on book content, so the answers are precise for the given questions. Additional work by domain experts beyond what has gone into writing the books is not necessary. Since the set of books in this study is limited to a few dozen, scalability is not a concern. As the system does not involve user data collection and is accessed through an API manager, security and privacy are ensured. Finally, the proper use of Elasticsearch with a relatively small dataset ensures the system's performance.

The Responde Agro API is in an operational environment and can be accessed through Embrapa's API platform AgroAPI (https://www.agroapi.cnptia.embrapa.br; Romani et al., 2023). It is used in the Ater+ Digital platform, related to the Ater Digital program, which aims to strengthen and expand the Brazilian System of Technical Assistance and Rural Extension by promoting the wide use of ICTs for increasing the farmers' competitiveness. Other applications can explore this approach for information structured in pairs of questions and answers. These include other e-books but also frequently asked questions documents or customer service databases, which can be information sources of services provided by similar APIs. It can also be used to determine the most relevant user queries so new content can be produced according to user demand. According to Caballero (2021), future research for QA systems should combine text-based and knowledge-based systems to maximize efficiency and accuracy. As a future step, this work can be built upon to explore knowledge bases, as well as related videos, to improve the user experience.

CONCLUSION

This paper presents a detailed description of a search engine capable of answering questions made in natural language or with keywords based on the content of Embrapa's "500 Questions, 500 Answers" collection. It is accessible via API and can be explored by digital solutions, such as Ater+ Digital, to provide an easier way to access information and increase knowledge and productivity, especially for small-scale farmers and extension services. This represents a contribution to disseminating technical information on agriculture and related topics. This paper shows how tools can be developed based on content structured in a question-and-answer format using a methodology that can be replicated in other applications, which includes the treatment of the text, especially from an e-book, and the creation of an API providing a search engine for its content.

CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

FUNDING STATEMENT

This work was supported by Embrapa, the Ministry of Agriculture and Livestock (Department of Family Farms and Cooperatives; SEG Project: 20.22.10.019.00.02; 359/2022), and the ILPF Network Association (SEG Project: 20.22.06.012.00.02; AgroTag ILPF).

PROCESS DATES

04, 2025

This manuscript was initially received for consideration for the journal on 04/23/2024, revisions were received for the manuscript following the double-anonymized peer review on 04/02/2025, the manuscript was formally accepted on 03/05/2025, and the manuscript was finalized for publication on 04/18/2025

CORRESPONDING AUTHOR

Correspondence should be addressed to Glauber Vaz; glauber.vaz@embrapa.br

REFERENCES

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). *Compilers: Principles, techniques and tools* (2nd ed.). Pearson Education.

Al Manir, M. S., Spencer, B., & Baker, C. J. (2018). Decision support for agricultural consultants with semantic data federation. *International Journal of Agricultural and Environmental Information Systems*, *9*(3), 87–99. DOI: 10.4018/IJAEIS.2018070106

Allam, A. M. N., & Haggag, M. H. (2012). The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences*, *2*(3).

Azizan, A., Bakar, Z. A., Rahman, N. A., Masrom, S., & Khairuddin, N. (2018). A comparative evaluation of search engines on finding specific domain information on the web. *IACSIT International Journal of Engineering and Technology*, *7*(4), 1–4. DOI: 10.14419/ijet.v7i4.33.23471

Bolfe, E. L., Jorge, L. A. de C., Sanches, I. D., Luchiari, A.Jr, da Costa, C. C., Victoria, D. de C., Inamasu, R. Y., Grego, C. R., Ferreira, V. R., & Ramirez, A. R. (2020). Precision and digital agriculture: Adoption of technologies and perception of Brazilian farmers. *Agriculture*, *10*(12), 1–16. DOI: 10.3390/agriculture10120653

Bray, T. (2014). *The Javascript object notation (JSON) data interchange format*. https://www.ietf.org/rfc/rfc7159.txt

Caballero, M. (2021). A brief survey of question answering systems. *International Journal of Artificial Intelligence & Applications*, *12*(5), 1–7. https://ssrn.com/abstract=3996229. DOI: 10.5121/ijaia.2021.12501

Cameron, R. D. (1993). Extending context-free grammars with permutation phrases. *ACM Letters on Programming Languages and Systems*, *2*(1–4), 85–94. DOI: 10.1145/176454.176490

Campos, F. R., Romanini, R. P., Rodrigues, M. E. N., Moura, M. F., & Vaz, G. J. (2022) Content from the books of Embrapa's 500 Questions 500 Answers Collection (Coleção 500 Perguntas 500 Respostas) treated to be used in digital solutions [Data set], Redape, V3. DOI: 10.48432/YIGNPF

Cardoso, M. J., Bastos, E. A., de Andrade, A. S., Jr., & Athayde Sobrinho, C. (Eds.; 2017). Cowpea beans: The producer asks, Embrapa answers [*Feijão-caupi: O produtor pergunta, a Embrapa responde*]. Embrapa. https://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1075578

Cordeiro, L. A. M., Vilela, L., Kluthcouski, J., & Marchão, R. L. (Eds.). (2015). Crop-livestock-forest systems: The producer asks, Embrapa answers [*Integração lavoura-pecuária-floresta: O produtor pergunta, a Embrapa responde*]. Embrapa. https://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1023335

Damiano, E., Spinelli, R., Esposito, M., & De Pietro, G. (2016). Towards a framework for closed-domain question answering in Italian. *Proceedings of the Twelfth International Conference on Signal-Image Technology and Internet-Based Systems*. IEEE. DOI: 10.1109/SITIS.2016.100

Daum, T., Buchwald, H., Gerlicher, A., & Birner, R. (2018). Smartphone apps as a new method to collect data on smallholder farming systems in the digital age: A case study from Zambia. *Computers and Electronics in Agriculture*, *153*, 144–150. DOI: 10.1016/j.compag.2018.08.017

Devi, M., & Dua, M. (2017). ADANS: An agriculture domain question answering system using ontologies. *Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE. DOI: 10.1109/CCAA.2017.8229784

Diefenbach, D. (2018). *Question answering over knowledge bases* (Doctoral dissertation, Université de Lyon). HAL theses.

Ding, E. (2016). Design and implementation of agricultural information resources vertical search engine based on Nutch. *Chemical Engineering Transactions*, *51*, 619–624. DOI: 10.3303/CET1651104

Elastic. (2022). *Elasticsearch guide [8.2]*. https://www.elastic.co/guide/en/elasticsearch/reference/8.2/index.html

Food and Agriculture Organization of the United Nations. (2022). *Digital agriculture.* https://www.fao.org/digital-agriculture/en/

Gaikwad, S., Asodekar, R., Gadia, S., & Attar, V. Z. (2015). AGRI-QAS question-answering system for agriculture domain. *Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. DOI: 10.1109/ICACCI.2015.7275820

Garrish, M., & Cramer, D. (Eds.). (2019). *EPUB 3.2: Final community group specification 08 May 2019.* https://www.w3.org/publishing/epub32/epub-spec.html

Heffner, R., Mines, R., Livingston, A., & Hartig, K. (2020). *The Forrester Wave: API management solutions, Q3 2020: The 15 providers that matter most and how they stack up*. Forrester Research.

Ingram, J., & Gaskell, P. (2019). Searching for meaning: Co-constructing ontologies with stakeholders for smarter search engines in agriculture. *NJAS Wageningen Journal of Life Sciences*, *90–91*(1), 1–13. DOI: 10.1016/j.njas.2019.04.006

Inwood, S. E. E., & Dale, V. H. (2019). State of apps targeting management for sustainability of agricultural landscapes. A review. *Agronomy for Sustainable Development*, *39*(1), 1–15. DOI: 10.1007/s13593-018-0549-8 PMID: 30881486

Janc, K., Czapiewski, K., & Wójcik, M. (2019). In the starting blocks for smart agriculture: The internet as a source of knowledge in transitional agriculture. *NJAS Wageningen Journal of Life Sciences*, *90–91*(1), 1–12. DOI: 10.1016/j.njas.2019.100309

Kamphuis, C., Vries, A. P., Boytsov, L., & Lin, J. (2020). Which BM25 do you mean? A large-scale reproducibility study of scoring variants. *Advances in Information Retrieval:* European Conference on Information Retrieval (ECIR 2020), 12036. Springer. DOI: 10.1007/978-3-030-45442-5_4

Kenny, U., & Regan, A. (2021). Co-designing a smartphone app for and with farmers: Empathising with end-users' values and needs. *Journal of Rural Studies*, *82*, 148–160. DOI: 10.1016/j.jrurstud.2020.12.009

Klerkx, L., Jakku, E., & Labarthe, P. (2019). A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. *NJAS Wageningen Journal of Life Sciences*, *90–91*(1), 1–16. DOI: 10.1016/j.njas.2019.100315

National Research Council of the National Academies. (2015). *Preparing the workforce for digital curation*. The National Academies Press.

NDJSON. (2014). *Newline delimited JSON*. https://ndjson.org/

Niranjan, P. Y., Rajpurohit, V. S., & Malgi, R. (2019). A survey on chat-bot system for agriculture domain. *Proceedings of the First International Conference on Advances in Information Technology (ICAIT)*. IEEE. DOI: 10.1109/ICAIT47043.2019.8987429

Pillai, S., Iijima, K., O'Neill, M., Santoro, J., Jain, A., & Ryan, F. (2021). *Magic quadrant for full life cycle API management*. Gartner Group.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, *3*(4), 333–389. DOI: 10.1561/1500000019

Romani, L. A., Evangelista, S. R., Vacari, I., Apolinário, D. R., Vaz, G. J., Speranza, E. A., Barbosa, L. A. F., Drucker, D. P., & Massruhá, S. M. F. S. (2023). AgroAPI platform: An initiative to support digital solutions for agribusiness ecosystems. *Smart Agricultural Technology*, *5*, 100247. DOI: 10.1016/j.atech.2023.100247

Rosa, G. M., Rodrigues, R. C., Lotufo, R. A., & Nogueira, R. (2021). Yes, BM25 is a strong baseline for legal case retrieval. arXiv preprint. arXiv:2105.05686.

Rotz, S., Duncan, E., Small, M., Botschner, J., Dara, R., Mosby, I., & Fraser, E. D. G. (2019). The politics of digital agricultural technologies: A preliminary review. *Sociologia Ruralis*, *59*(2), 203–229. DOI: 10.1111/soru.12233

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach*. Pearson Education.

Salunkhe, A. (2020). Evolution of techniques for question answering over knowledge base: A survey. *International Journal of Computer Applications*, *177*(34), 9–14. DOI: 10.5120/ijca2020919817

Siné, M., Haezebrouckl, T. P., & Emonet, E. (2015). API-AGRO: An open data and open API platform to promote interoperability standards for farm services and ag web applications. *Agrárinformatika Folyóirat*, *6*(4), 56–64. DOI: 10.17700/jai.2015.6.4.209

Šūmane, S., Kunda, I., Knickel, K., Strauss, A., Tisenkopfs, T., des Ios Rios, I., Rivera, M., Cgebacgm, T., & Ashkenazy, A. (2018). Local and farmers' knowledge matters! How integrating informal and formal knowledge enhances sustainable and resilient agriculture. *Journal of Rural Studies*, *59*, 232–241. DOI: 10.1016/j.jrurstud.2017.01.020

Top, J., & Wigham, M. (2015). The role of e-science in agriculture: How e-science technology assists participation in agricultural research. In *EU SCAR. Agricultural knowledge and innovation systems towards the future: A foresight paper*. European Commission.

Uttarwar, S., Gambani, S., Thakkar, T., & Mulla, N. (2019). Machine learning based review on development and classification of question-answering systems. *Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. DOI: 10.1109/ICCMC.2019.8819667

Van Campenhout, B. (2017). There is an app for that? The impact of community knowledge workers in Uganda. *Information Communication and Society*, *20*(4), 530–550. DOI: 10.1080/1369118X.2016.1200644

Van Campenhout, B., Spielman, D. J., & Lecoutere, E. (2021). Information and communication technologies to provide agricultural advice to smallholder farmers: Experimental evidence from Uganda. *American Journal of Agricultural Economics*, *103*(1), 317–337. DOI: 10.1002/ajae.12089

Vaz, G. J., Veiga, P. H. R. C., Caldas, R. G., Vidal, W. C. L., Assis, C. P., Correa, J. L., & Moura, M. F. (2023). Treatment of text extracted from digital books for search engine indexing [Tratamento de texto extraído de livros digitais para a indexação em mecanismo de busca]. *Revista Ibero-Americana de Ciência da Informação*, *16*(2), 311–328. DOI: 10.26512/rici.v16.n2.2023.42740

Willems, D. J., Koenderink, N. J., & Top, J. L. (2015). From science to practice: Bringing innovations to agronomy and forestry. *Agrárinformatika Folyóirat*, *6*(4), 85–95. DOI: 10.17700/jai.2015.6.4.214

Xiong, M., Li, A., Xie, Z., & Jia, Y. (2018). A practical approach to answer extraction for constructing QA solution. *Proceedings of the Third International Conference on Data Science in Cyberspace (DSC)*. IEEE. DOI: 10.1109/DSC.2018.00064

Yao, Y. (2017). Library resource vertical search engine based on ontology. *Proceedings of the 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*. IEEE. DOI: 10.1109/ICSGEA.2017.159

Glauber José Vaz received his bachelor's degree in computer science from the Federal University of Uberlândia in 2000 and his master's degree from the State University of Campinas (Unicamp) in 2003. From 2003 to 2010, he taught computing courses in three institutions, including at Unicamp. Since 2010, he has worked in research, development, and innovation in computing applied to agriculture, at the Digital Agriculture unit of the Brazilian Agricultural Research Corporation (Embrapa), in Campinas, Brazil. His research interests include information retrieval, data science, artificial intelligence, and digital agriculture.