

FactCheckToolEC: A Cloud-Based System Demonstration for Semiautomated Fact-Checking in Ecuadorian Spanish

Mariuxi del Carmen Toapanta-Bernabé^{1,2,*†}, Miguel Ángel García-Cumbreras^{1,†},
Luis Alfonso Ureña-López^{1,†}, Jordy Jaír Pivaque-Franco^{2,†} and
Gino Polinario-Rodríguez Pesántez^{2,†}

¹Universidad de Jaén, Campus Las Lagunillas s/n, 23071 Jaén, España

²Universidad de Guayaquil, Cda. Universitaria, Av. Delta s/n y Av. Kennedy, Guayaquil, Ecuador

Abstract

This paper presents *FactCheckToolEC*, a cloud-based system for semiautomated fact-checking in Ecuadorian Spanish. The platform integrates a web interface, containerized microservices, MongoDB-based storage, and multiple Natural Language Processing components within a unified environment. Its workflow supports authenticated access, dashboard visualization, news collection and consultation, and a multistage verification pipeline including claim normalization, subjectivity analysis, sentiment analysis, named entity recognition, numerical verification, scientific verification, and final veracity assessment. The demonstration highlights the end-to-end operation of the system in Oracle Cloud Infrastructure. It shows how the platform supports traceable, modular, and scalable fact-checking processes through a single interface.

Keywords

fact-checking, Ecuadorian Spanish, natural language processing

1. Introduction

This system demonstration addresses the growing need for computational support in fact-checking workflows, particularly in contexts where false or misleading information spreads rapidly through digital media. In Ecuador, this challenge is especially relevant due to the increasing circulation of online content and the need to process information efficiently in scenarios related to politics, public health, security, and social events. Manual verification alone is often insufficient to deal with the volume, speed, and linguistic variability of the information that must be reviewed in a timely manner [1, 2].

To contribute to this problem, we present *FactCheckToolEC*, a cloud-based platform designed to support semiautomated fact-checking in Ecuadorian Spanish. The system integrates a web interface, containerized microservices, a MongoDB-based storage layer, and a set of Natural Language Processing (NLP) modules within a unified environment. Its workflow supports authenticated access, dashboard visualization, news collection and consultation, and a multistage verification pipeline that allows the user to inspect the processing status and outputs of each analyzed item. This version of the platform builds on previous deployments by reorganizing the frontend, adapting the regional linguistic model for Ecuadorian Spanish, updating the corpus with verified news from June 2025 to February 2026, and integrating specialized numerical and scientific verification modules.

The main goal of this demonstration paper is to present the platform's operational behavior from both technical and functional perspectives. In particular, the paper describes the system architecture, the

SEPLN 2026: 42nd International Conference of the Spanish Society for Natural Language Processing, León, Spain, 22-25 September 2026.

*Corresponding author.

†These authors contributed equally.

✉ mariuxi.toapantab@ug.edu.ec (M. d. C. Toapanta-Bernabé); magc@ujaen.es (M. : García-Cumbreras); laurena@ujaen.es (L. A. Ureña-López); jordy.pivaquef@ug.edu.ec (J. J. Pivaque-Franco); gino.rodriguezp@ug.edu.ec (G. P. Pesántez)

🌐 <https://www.ujaen.es/centros/ceatic/en/contactos/garcia-cumbreras-miguel-angel> (M. : García-Cumbreras);

<https://www.ujaen.es/centros/ceatic/contactos/urena-lopez-l-alfonso> (L. A. Ureña-López)

🆔 0000-0002-4839-7452 (M. d. C. Toapanta-Bernabé); 0000-0003-1867-9587 (M. : García-Cumbreras); 0000-0001-7540-4059 (L. A. Ureña-López)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

NLP components currently integrated into the verification workflow, and the demonstration scenario used to validate the platform’s end-to-end interaction. The proposed system is intended not only as a practical tool to support fact-checkers but also as a reproducible, research-oriented environment for developing and evaluating semiautomated fact-checking technologies for Ecuadorian Spanish.

2. Related Works

Automated fact-checking has been widely studied as a Natural Language Processing task, with work spanning claim detection, evidence retrieval, textual entailment, and veracity classification [1]. In this context, benchmark resources such as FEVER have played a central role in consolidating fact verification as a relevant research problem and in promoting pipeline-based approaches for evidence-oriented verification [2].

At the modeling level, Transformer-based approaches have become a standard choice for semantic and classification tasks in fact-checking workflows. In particular, BERT-based architectures have provided strong foundations for claim representation, semantic comparison, and downstream verification components [3]. In addition, scientific fact-checking has emerged as a specialized subarea, requiring domain-aware resources and methods for handling evidence in science- and health-related claims [4].

Beyond dataset-centered and model-centered research, recent NLP applications have also highlighted the importance of deployable, user-facing systems that make verification workflows observable, reproducible, and operational in real-world settings. In this sense, demonstration papers play a complementary role by showing how multiple NLP components can be integrated into a single environment for human-centered interaction and semiautomated analysis. In contrast to previous work, which mainly focused on datasets, models, or isolated subtasks, this paper presents a system demonstration centered on the end-to-end deployment of a cloud-based platform for Ecuadorian Spanish. The contribution of *FactCheckToolEC* lies in integrating multiple NLP services, user-facing modules, and traceable data management within a single operational environment designed for semiautomated fact-checking.

3. System Architecture

FactCheckToolEC follows a modular architecture in which the presentation layer, application services, and data layer are clearly separated. The user interacts with the system via a web interface, while the backend services handle requests for administration, data collection, NLP analysis, and verification. The complete platform can be run locally for development or deployed to Oracle Cloud Infrastructure (OCI) for remote access and live demonstrations.

The frontend is implemented as a web application that provides authenticated access to the platform and exposes the main system modules through a navigation menu. These modules include dashboard visualization, administration, news collection, news consultation, verification stages, and consolidated result views. This interface was designed to support the fact-checking workflow through a single access point, reducing the need to switch between disconnected tools.

The backend is organized as a set of containerized microservices, each one responsible for a specific task in the overall workflow. These services support functionalities such as authentication, administration, data ingestion, web scraping, analysis, and specialized NLP-based verification. This design improves modularity, maintainability, and service independence, while also facilitating deployment and updates of individual components without affecting the complete platform.

The system uses MongoDB as the storage layer for users, metadata, sources, categories, collected news items, and processing results. In the cloud-based configuration, the platform is deployed on OCI and exposed through a public web endpoint. Internally, the frontend communicates with backend services, which interact with the database layer to persist the outputs of the verification workflow. This architecture supports traceable interactions between the user interface and the processing pipeline, which is essential for semiautomated fact-checking scenarios.

FactCheckToolEC Deployment Architecture in Oracle Cloud

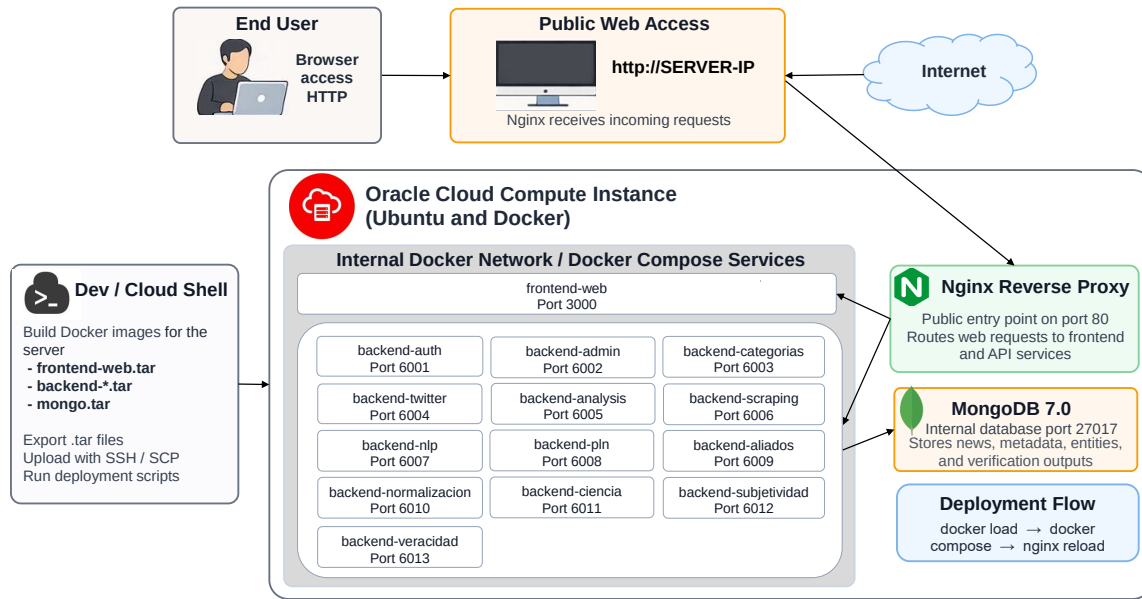


Figure 1: Deployment architecture of *FactCheckToolEC* in Oracle Cloud. The platform is deployed on an Ubuntu-based compute instance using Docker, with public access via Nginx over HTTP: a frontend service on port 3000; core platform microservices on ports 6001–6009; additional AI verification microservices on ports 6010–6013; and MongoDB on port 27017.

Figure 1 illustrates the deployment and service interaction model used in the demonstrated cloud-based configuration.

3.1. Execution Modes and Deployment

The platform supports two main execution modes: local execution for development and testing, and cloud-based deployment for remote access and live demonstrations. In the local setting, the system is started in two stages. First, the backend group is launched, including MongoDB and the available microservices; then, the frontend web application is started separately. After that, a restoration script loads the project backup into the database, allowing the user to access the system through `http://localhost:3000`. The local execution documentation also defines two user profiles: an administrator account with full access to dashboard, news, scraping, and control modules, and an operator account with read-only access to news and charts.

For the cloud-based setting, the system is deployed in Oracle Cloud Infrastructure (OCI). The technical manual describes a deployment process based on Dockerized services, public web access via Nginx, and a two-layer firewall configuration that combines OCI security rules with Ubuntu `iptables`. The deployment workflow is supported by automation scripts for complete deployment, partial service updates, database restoration, and backup generation. In this configuration, the application is exposed through a public IP address, while the backend services and the storage layer remain internally organized as a modular service ecosystem.

This dual execution model increases reproducibility and practicality. The local mode facilitates development, debugging, and controlled testing, whereas the cloud mode supports live system demonstrations and shared browser-based access. From the perspective of a system demonstration paper, this is relevant because it shows that the platform is not only conceptually designed but also operationally documented for both standalone and remote deployment scenarios. This deployment scheme is particularly relevant for demonstration purposes because it supports both reproducible local execution and live remote interaction with the same modular service ecosystem.

4. NLP Components

The NLP layer of *FactCheckToolEC* is implemented as a modular microservice architecture rather than as a single monolithic backend. In the cloud deployment, the platform combines a core set of Docker services with additional AI-oriented verification services. The core platform includes backend-auth, backend-admin, backend-categorias, backend-twitter, backend-analysis, backend-scraping, backend-nlp, backend-pln, and backend-aliados. At the same time, the extended verification layer incorporates independent services for claim normalization, scientific verification, subjectivity analysis, and numerical verification. This design allows each task to be deployed, updated, and maintained separately without interrupting the rest of the platform [5].

From the perspective of information sources, the system does not operate on a single static dataset. Instead, it combines content ingestion and management through multiple services. The backend-twitter component supports Twitter- and news-oriented content handling; backend-scraping is responsible for automated collection from external web sources; and backend-categorias and backend-aliados support organizational metadata used in the verification workflow. At the operational level, this means that the platform continuously manages news items, categories, and source-related information before they are passed to the linguistic and verification modules.

The user-facing workflow exposes several NLP-driven processing stages. These include claim normalization, subjectivity analysis, sentiment analysis, Named Entity Recognition (NER), numerical verification, scientific verification, and final veracity assessment. Functionally, each processed news item can be enriched with normalized text, polarity and subjectivity cues, detected entities, numerical or scientific validation outputs, and verification-oriented labels before reaching the consolidated result view. In this sense, the system implements a pipeline in which linguistic preprocessing and verification tasks are progressively accumulated rather than produced by a single endpoint.

A relevant characteristic of the current architecture is that several advanced verification modules are deployed as independent Docker services. The normalization module is maintained as a standalone microservice that can be updated independently of the rest of the stack. The scientific verification module is deployed as backend-ciencia on port 6011 and includes domain-specific assets, such as pretrained BERT-based resources and embedding components, packaged within the image. The subjectivity module is also deployed independently, with port 6012 exposed externally and port 5005 used internally by the container. Likewise, the numerical verification module is deployed as backend-veracidad, even though functionally it corresponds to the numerical verification stage presented in the interface [5, 6, 7, 8, 9].

From a system-design perspective, this organization is closer to a service-oriented fact-checking pipeline than to a monolithic language model or a Retrieval-Augmented Generation architecture. The available documentation supports a modular ecosystem of Dockerized services connected via a shared network and coordinated by the platform workflow. Still, it does not explicitly describe a complete RAG pipeline. Therefore, the contribution of *FactCheckToolEC* lies in integrating multiple information sources, specialized NLP microservices, and verification modules into a single operational environment for semiautomated fact-checking in Ecuadorian Spanish. Future work may extend this architecture with retrieval-oriented evidence modules or additional domain-adapted language models.

4.1. Linguistic Adaptation and Model Choice

An important aspect of the platform is its orientation toward Ecuadorian Spanish and its regional variability. According to the user documentation, the claim normalization module relies on a fine-tuned BETO-based model to determine semantic equivalence and detect duplicates. In addition, the associated thesis motivates the use of Transformer-based and subword-based models such as BERT, RoBERTuito, and FastText as suitable alternatives for handling dialectal variation, noisy content, and social-media-oriented language in Ecuadorian contexts [9].

This design choice is relevant for fact-checking because misinformation in Ecuadorian digital media often appears in informal, abbreviated, and regionally marked language. The associated thesis reports that RoBERTuito achieved better results than generic BERT and FastText when handling Ecuadorian linguistic variation, especially in social-media-style texts [9]. Therefore, *FactCheckToolEC* is not only a deployment environment for verification services but also a research-oriented platform for adapting NLP models to regional varieties of Spanish.

5. Demonstration Scenario

The demonstration scenario was designed to show the complete operational workflow of *FactCheckToolEC* from the user's perspective. The demo begins with authenticated access to the platform, where the user enters valid credentials and reaches the main dashboard. This initial view summarizes the system's current state using cards, statistical charts, process indicators, and highlighted named entities, providing a general overview of the collected and processed news items. During the live demonstration, the audience will observe how a news item progresses through the platform, from collection and consultation to modular analysis and final verification.

After logging in, the user can navigate to the collection and consultation modules. In the collection stage, the system supports retrieving news items from external sources via web scraping and related ingestion processes. In the consultation stage, the user can browse the collected items, apply filters, inspect metadata, and open the detailed view of each news item. This part of the demonstration shows how the system organizes the available information before the execution of the verification workflow.

The demo then continues with the NLP-based processing stages. Selected news items move through claim normalization, subjectivity analysis, sentiment analysis, named entity recognition, numerical verification, scientific verification, and final veracity assessment. At each stage, the platform updates the news item's processing status and records the outputs produced by the corresponding module. This step illustrates the practical value of the microservice-based design, since multiple analysis services can be orchestrated in a traceable and modular way.

Finally, the demonstration concludes in the consolidated report view, where fully processed news items are displayed alongside their metadata, activated process labels, extracted entities, and verification results. This final stage is particularly important because it makes the complete semiautomated workflow observable within a single interface, from data collection and consultation to analysis and final verification. In this way, the demo validates not only the platform's functional availability but also the interaction among the interface, cloud deployment, NLP pipeline, and storage layer.

5.1. User Workflow and Data Management

The user workflow implemented in *FactCheckToolEC* is explicitly documented in the user manual and follows a structured sequence. The interaction starts with login, where the user accesses the system through authenticated credentials. Once inside, the dashboard provides a visual summary of the available information, including cards with collected news counts, process indicators for verification stages, statistical charts organized by region, source, verifier, and media type, and highlighted named entities grouped into persons, organizations, and locations.

After this entry point, the user can move to the collection and consultation modules. The collection stage involves web scraping and ingestion of external data. At the same time, the consultation module allows users to browse news items using search bars, filters, metadata fields, and detailed record views. The manual specifies that each detailed record may include the title, source, region, publication date, processing status, activated process labels, detected entities, and even the item's JSON representation, thereby improving traceability and transparency during verification.

The data layer also contributes to the system's technical relevance. According to the local execution manual, MongoDB stores not only processed news items but also users and supporting dimensions such as source categories, news types, and verification labels. In the thesis, the data schema is further described through fields such as source, link, likes, retweets, sentiment, entities, normalized text,

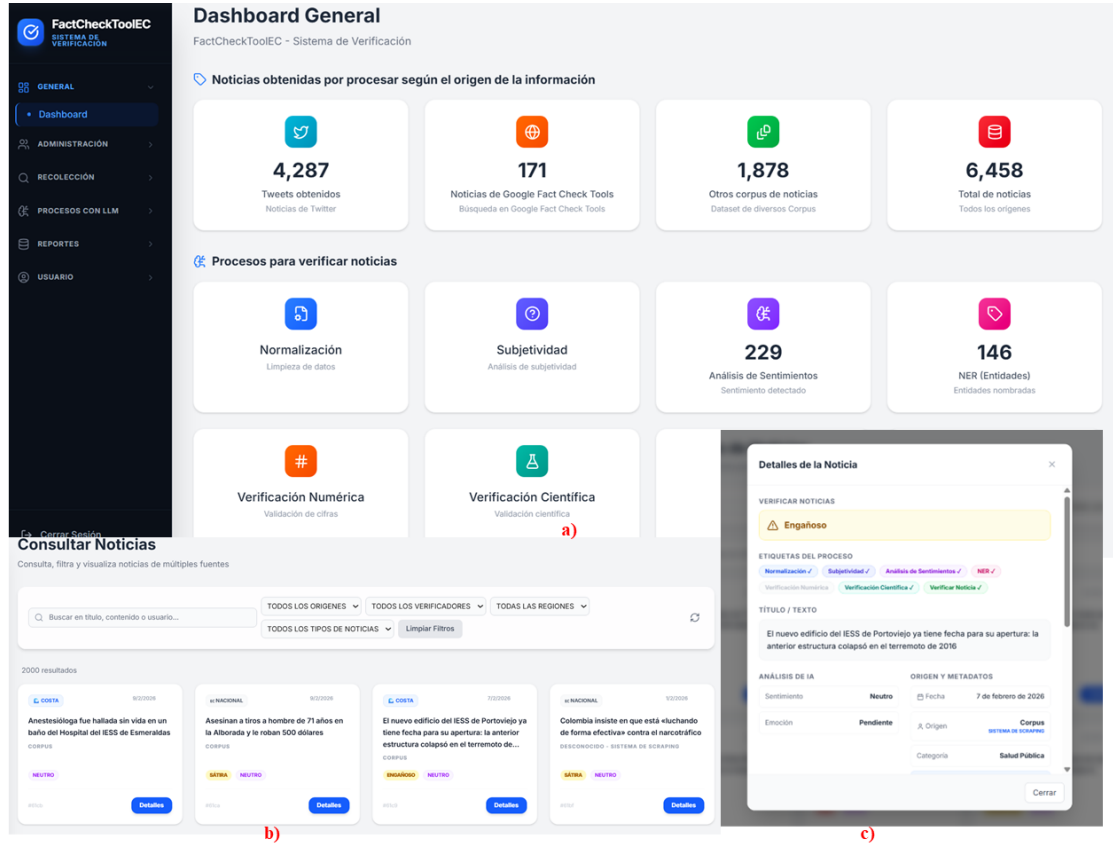


Figure 2: Main interface views of *FactCheckToolEC*: (a) dashboard overview with source statistics and verification stages, (b) news consultation interface with filters and detailed access, and (c) detailed verified record with activated process labels and analysis metadata.

thematic category, verifier, final veracity label, regional variety, and confidence score. This combination of user-facing workflow and structured storage strengthens the platform’s value as a semiautomated fact-checking environment rather than a simple news-browsing interface.

Figure 2 complements this workflow by illustrating the main user-facing views of the platform, including the dashboard, the consultation interface, and the detailed verified record.

Table 1 summarizes the main functional modules of *FactCheckToolEC* and the observable outputs presented during the demonstration.

5.2. Validation

Although this paper primarily focuses on system demonstration, the current version of *FactCheckToolEC* is supported by quantitative evidence from the platform’s main verification modules. For subjectivity detection, the DistilBETO-based component was evaluated on an independent test set of 1,928 Ecuadorian-news sentences and achieved 88.30% accuracy, 91.37% precision, 93.65% recall, and 92.50% F1-score [7].

For numerical verification, the best-performing configuration was RoBERTuito, which obtained 0.8122 accuracy, 0.8106 precision, 0.8122 recall, and 0.8110 F1-score on the test set [8].

For scientific verification, the classification component achieved stable results under stratified five-fold cross-validation, with an average Macro-F1 of 0.8547. In addition, the scientific evidence retrieval module obtained an MRR@5 of 0.6563 and a Recall@5 of 0.7778 using the multilingual-e5-base retrieval approach, outperforming BM25 and SciBERT-NLI on the evaluated corpus [6].

For claim normalization, the currently available evidence should be interpreted as preliminary. For the corpus version available up to July 2025, the lexical baseline achieved an F1-score of 0.17 and a recall

Table 1Main functional modules of *FactCheckToolEC*.

Module	Main function	Observable output in the demo
Dashboard	Global monitoring of the platform	Cards, process indicators, statistical charts, highlighted NER entities
Web Scraping	News collection from external sources	Retrieved news items available for consultation
Consult News	Inspection and filtering of collected news	Detailed view with metadata, status, labels, entities, and JSON record
Claim Normalization	Semantic equivalence and duplicate detection	Updated process status and normalized claim information
Subjectivity Analysis	Detection of opinions and bias	Subjectivity-oriented processing result
Sentiment Analysis	Detection of emotional polarity	Sentiment-oriented output for each processed news item
NER	Entity extraction	Persons, organizations, locations, and numerical expressions
Numerical Verification	Quantitative claim assessment	Evidence-oriented numerical validation output
Scientific Verification	Verification of health, science, and technology claims	Domain-oriented verification result
Veracity Verification	Final truthfulness assessment	True, false, misleading, or imprecise-oriented result
Consolidated Report	Final integrated view of processed items	Verified news with metadata, labels, entities, and outputs

of 0.13, while transformer-based models substantially improved performance, with mBERT reaching 0.80 F1 and 0.85 recall, and BETO reaching 0.81 F1 and 0.83 recall. Since the corresponding study reports these values prior to the final integration of the extended corpus, they are presented here as preliminary support rather than as final benchmark results [9].

Taken together, these results indicate that the demonstrated platform is supported by competitive module-level performance in subjectivity detection, numerical verification, scientific discourse detection, scientific evidence retrieval, and claim normalization. Although a fully unified end-to-end benchmark across all modules is still pending, the available evidence is sufficient to support the practical usefulness of *FactCheckToolEC* as a semiautomated fact-checking environment for Ecuadorian news.

6. Limitations

Despite the usefulness of the proposed demonstration, several limitations should be acknowledged. First, the current implementation is tightly coupled to the technological stack already adopted by *FactCheckToolEC*, particularly its deployment in Oracle Cloud Infrastructure and its microservice-based integration strategy. As a result, the present work demonstrates feasibility and operational integration in the OCI environment, but it does not yet evaluate portability across alternative cloud infrastructures or deployment settings [5].

Second, although the platform is designed for the Ecuadorian context, some linguistic challenges remain open. The system explicitly aims to address regional variation in Ecuadorian Spanish, including expressions and discourse patterns from the Costa, Sierra, and Amazonía regions, but this adaptation is still an ongoing challenge. Informal language, local slang, abbreviated expressions, and context-dependent wording in digital media may still reduce the robustness of some modules, especially when claims are phrased in highly colloquial or ambiguous ways [9].

Third, the validation currently available for the demonstrated platform is mainly modular rather than fully end-to-end. The integrated system is supported by quantitative results from subjectivity detection, numerical verification, scientific verification, scientific evidence retrieval, and preliminary

normalization experiments; however, not all modules have yet been evaluated under a single unified benchmark over the complete pipeline. Therefore, the current evidence supports the practical usefulness of the platform, but a consolidated end-to-end evaluation remains pending [7, 8, 6, 9].

Finally, some components still depend on the quality, availability, and coverage of external or domain-specific sources. This is particularly relevant for scientific evidence retrieval and for modules that require updated corpora or specialized validation resources. Consequently, system performance may vary depending on the topical domain, the availability of reliable evidence, and the maturity of the corresponding module.

7. Conclusions

This paper presented *FactCheckToolEC*, a cloud-based system demonstration for semiautomated fact-checking in Ecuadorian Spanish. The platform combines a web interface, a microservice-based backend, MongoDB-based storage, and multiple NLP components to support the collection, consultation, analysis, and verification of news items within a unified environment.

From the technical perspective, the system demonstrates that a modular deployment in Oracle Cloud Infrastructure can support reproducible and traceable fact-checking workflows. From a functional perspective, integrating claim normalization, subjectivity analysis, sentiment analysis, named entity recognition, numerical verification, scientific verification, and final veracity assessment provides fact-checkers with a richer interaction model than isolated or manually disconnected tools.

In addition, the current version of the platform reflects a practical evolution of previous fact-checking deployments by improving usability, extending the verification pipeline, and adapting the system to Ecuadorian regional variation and updated national data.

As future work, the platform can be extended through broader evaluations with professional fact-checkers, additional explainability mechanisms for NLP outputs, and further adaptation to regional varieties of Ecuadorian Spanish. These directions would strengthen both the system’s practical usefulness and its value as a research-oriented environment for semiautomated fact-checking. In this sense, *FactCheckToolEC* contributes not only as a deployable application but also as a modular experimental environment for studying semiautomated fact-checking in Ecuadorian Spanish.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for grammar and spelling checks and ChatGPT (OpenAI) for improving writing style and formatting. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. URL: <https://aclanthology.org/2022.tacl-1.11/>. doi:10.1162/tacl_a_00454.
- [2] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. URL: <https://aclanthology.org/N18-1074/>. doi:10.18653/v1/N18-1074.
- [3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long

and Short Papers), Association for Computational Linguistics, 2019. URL: <https://aclanthology.org/N19-1423/>.

- [4] J. Vladika, F. Matthes, Scientific fact-checking: A survey of resources and approaches, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 6215–6230. URL: <https://aclanthology.org/2023.findings-acl.387/>. doi:10.18653/v1/2023.findings-acl.387.
- [5] J. J. Pivaque Franco, G. P. Rodríguez Pesántez, Integración de nuevas funcionalidades en el sistema de apoyo a los verificadores de hechos en Oracle Cloud Infrastructure para detectar noticias falsas, 2025. URL: <https://repositorio.ug.edu.ec/handle/redug/87307>, Universidad de Guayaquil, Publicado en DSpace UG.
- [6] C. J. Chusan Zambrano, A. D. Plúas Mero, Implementación de detección de afirmaciones científicas y recuperación de fuentes en el sistema de verificación de noticias falsas en Oracle Cloud Infrastructure, 2025. URL: <https://repositorio.ug.edu.ec/handle/redug/87299>, Universidad de Guayaquil, Publicado en DSpace UG.
- [7] C. T. Rodríguez Bernal, A. D. Triviño León, Implementación de detección de subjetividad en el sistema de apoyo a los verificadores de hechos para noticias falsas en Oracle Cloud Infrastructure, 2025. URL: <https://repositorio.ug.edu.ec/handle/redug/87300>, Universidad de Guayaquil, Publicado en DSpace UG.
- [8] D. D. Mora Intriago, G. D. Morocho Chafla, Implementación de verificación de afirmaciones numéricas en noticias de Ecuador en el sistema de verificación de noticias falsas en Oracle Cloud Infrastructure, 2025. URL: <https://repositorio.ug.edu.ec/handle/redug/87306>, Universidad de Guayaquil, Publicado en DSpace UG.
- [9] J. A. Jipa Andi, J. D. Pérez Navarro, Implementación de normalización de afirmaciones en el sistema de apoyo a los verificadores de hechos basado en Oracle Cloud Infrastructure, 2025. URL: <https://repositorio.ug.edu.ec/handle/redug/87301>, Universidad de Guayaquil, Publicado en DSpace UG.