УДК 004.89:004.51 https://doi.org/10.15407/pp2025.02.077

B. O. Kuzikov, O. A. Shovkoplias, P. O. Tytov, S. R. Shovkoplias

APPLICATION OF SMALL LANGUAGE MODELS FOR SEMANTIC ANALYSIS OF WEB INTERFACE ACCESSIBILITY

Web accessibility remains a critical aspect of ensuring equal opportunities for internet resource usage, especially for people with disabilities. The Web Content Accessibility Guidelines 2.5.3 criterion "Label in Name" requires that the accessible name of an interface component include text that is visually presented. Existing automated verification methods for this criterion are predominantly based on primitive string comparison, which does not account for semantic context. Objective: investigate the possibilities of using small language models with up to 1 billion parameters for automated semantic analysis of compliance with Web Content Accessibility Guidelines 2.5.3, as an alternative to resource-intensive large language models and limited algorithmic methods. Methodology: the research involved creating synthetic datasets (7,200 English-language and 5,615 Ukrainian-language samples) and using real-world datasets (Top500 – 380 samples, UaUniv – 319). Sentence Bidirectional Encoder Representations from Transformers models were tested for computing semantic similarity, and fine-tuning of the google/electra-base-discriminator model was performed for 3-class classification of semantic relationships ("similar", "unrelated", "opposite"). Results: the trained model of 437 MB demonstrated high accuracy on synthetic data (0.96) and sufficient accuracy on real datasets (Top500: 0.77, UaUniv: 0.73). The model effectively identifies all three classes of semantic relationships with an accuracy of 95.1 % for "opposite", 92.7 % for "unrelated", and 97.4 % for "similar" texts in the validation sample. Conclusions: the research confirmed the feasibility of using small language models for automated verification of semantic compliance according to Web Content Accessibility Guidelines 2.5.3. The proposed approach provides acceptable classification accuracy with significantly lower computational costs compared to large language models, allowing for the integration of semantic analysis into standard development and testing processes. Despite certain limitations, the developed solution can significantly improve web accessibility testing.

Keywords: Small Language Models, Semantic Analysis, Text Classification, Web Accessibility, Web Content Accessibility Guidelines, Model Fine-Tuning, Natural Language Processing

Б. О. Кузіков, О. А. Шовкопляс, П. О. Титов, С. Р. Шовкопляс

ЗАСТОСУВАННЯ МАЛИХ МОВНИХ МОДЕЛЕЙ ДЛЯ СЕМАНТИЧНОГО АНАЛІЗУ ДОСТУПНОСТІ ВЕБІНТЕРФЕЙСІВ

Вебдоступність залишається важливим аспектом для забезпечення рівних можливостей користування інтернет-ресурсами, особливо для людей з інвалідністю. Критерій Настанов з доступності вебвмісту 2.5.3 «Мітка в імені» вимагає, щоб доступне ім'я компонента інтерфейсу включало текст, представлений візуально. Існуючі методи автоматизованої перевірки цього критерію базуються переважно на примітивному порівнянні рядків, не враховуючи семантичний контекст. Мета. Дослідити можливості застосування малих мовних моделей із кількістю параметрів до 1 мільярда для автоматизованого семантичного аналізу відповідності критерію Настанов із доступністю вебвмісту 2.5.3 як альтернативи ресурсомістким великим мовним моделям і обмеженим алгоритмічним методам. Методологія. Дослідження передбачало створення синтетичних (англомовних – 7200, україномовних – 5615 прикладів) та використання реальних наборів даних (380 прикладів із 500 найвідвідуваніших вебсайтів, 319 прикладів із вебсайтів українських університетів). Здійснено тестування моделей двоспрямованих кодувальних представлень речень із трансформерів для обчислення семантичної схожості та використано тонке налаштування базової дискримінаторної моделі ELECTRA від Google: для 3-класової класифікації семантичних відношень («схожі», «непов'язані», «протилежні»). Результати. Навчена модель розміром 437 МБ продемонструвала високу точність на синтетичних даних (0,96) та достатню точність на реальних наборах даних (0,77 для найвідвідуваніших вебсайтів та 0,73 для вебсайтів українських університетів). Модель здатна ефективно ідентифікувати всі три класи семантичних відношень із точністю 95,1 % для «протилежних», 92,7% для «непов'язаних» та 97,4% для «схожих» текстів на валідаційній вибірці. Висновки. Дослідження підтвердило доцільність застосування малих мовних моделей для автоматизованої перевірки семантичної відповідності згідно з критерієм Настанов із доступністю вебвмісту 2.5.3. Запропонований підхід забезпечує при-

© Б. О. Кузіков, О. А. Шовкопляс, П. О. Титов, С. Р. Шовкопляс, 2025

йнятну точність класифікації при значно менших обчислювальних витратах порівняно з великими мовними моделями, що дозволяє інтегрувати семантичний аналіз у стандартні процеси розроблення та тестування. Незважаючи на певні обмеження, розроблене рішення може істотно покращити процес тестування вебдоступності.

Ключові слова: малі мовні моделі, семантичний аналіз, класифікація тексту, вебдоступність, Настанови з доступності вебвмісту, тонке налаштування моделей, обробка природної мови

Introduction

Motivation. Web accessibility is critically important for ensuring equal access to information and services for all users. The Web Content Accessibility Guidelines (WCAG) define relevant standards. One important criterion is WCAG 2.5.3 “Label in Name”, which specifies that the accessible name of an interface component must include text that is visually presented. This allows users with disabilities to rely on visible labels as a means of interaction: individuals using voice control can activate elements by speaking their visible names, and users of text-to-speech technologies gain a better experience due to consistency between seen and heard text [1]. Developers have ethical and legal obligations to comply with accessibility standards. Conducting accessibility testing is fundamental to improving application usability for people with disabilities and generally enhances usability for all users [2].

Automated accessibility testing is recognized as an effective tool for quickly identifying a significant portion of problems. It allows for systematic evaluation of the user interface and code for compliance with numerous rules and recommendations [2]. However, despite its advantages, automated testing is not an exhaustive solution and has significant limitations [3]. In particular, automated tools cannot fully evaluate the context of use, complex interactions, and subjective aspects of user experience, which are critically important for users with different needs [4, 5]. Passing automated tests does not guarantee full application accessibility, and results may contain false positives that require human verification. Thus, a comprehensive approach to ensuring accessibility requires a combination of automated testing with manual testing and involvement of users with disabilities.

The task of verifying WCAG 2.5.3 criterion essentially comes down to fuzzy string comparison, with or without consideration of semantics and structure, depending on the quality of the tool. Existing automated tools may take into account Best Practice recommendations – “The label should begin with visible text”. Classical, deterministic string comparison algorithms can be divided into several categories: fuzzy text comparison (Levenshtein distance, longest common subsequence), phonetic algorithms (Soundex, Metaphone, New York State Identification Intelligence System), and token-based methods (Jaccard coefficient, cosine similarity, BM25). It is important to note that from those listed, only the last category can account for semantic proximity of texts, but its capabilities are limited without using.

Recently, there has been significant interest in artificial intelligence capabilities. Despite impressive results, tools such as ChatGPT have limitations in specific tasks, particularly in accessibility testing, as they use training data that does not always cover the specifics and requirements of modern accessibility standards [6]. In the context of artificial intelligence development, particularly in the field of natural language processing, language models are often classified by their size and computational requirements. Large language models (LLMs), such as GPT-4 or GLaM [7], contain hundreds of billions or even trillions of parameters and require significant computational resources for training and execution. Due to their ability to consider semantic relationships and context, LLMs can understand and evaluate headings and labels with a high degree of accuracy. However, deploying LLMs for real-time accessibility testing faces significant challenges, including high computational requirements, high latency, and potential instability of provider APIs. This creates a need for efficient, reliable, and context-oriented solutions that can operate with limited computational resources.

Small language models (SLMs) are significantly more compact – they typically contain from a few hundred million to a few billion parameters, ensuring their availability and efficient deployment on standard equipment and facilitating integration into everyday tools and development processes [8]. Among SLMs, micro- and nano-language models stand out, with parameter counts typically not exceeding one billion. Despite their smaller size, these models can achieve performance comparable to larger models in specific tasks after appropriate fine-tuning [8, 9]. For example, the Phi-3-mini model with only 3.8 billion parameters demonstrated performance corresponding to models twice its size in various natural language understanding tasks [10].

The aim of the research is to analyze the capabilities of SLMs (up to 1 billion parameters) for automated verification of compliance with WCAG 2.5.3 criterion. We investigate the use of Sentence Transformers (SBERT) models, ready-made classification tasks from the Hugging Face platform, and fine-tuning of a selected SLM. The hypothesis is that properly configured SLMs can provide accurate semantic analysis of relationships between labels and names through 3-class classification (“similar”, “unrelated”, “opposite”), while maintaining the performance characteristics necessary for integration into existing development processes. This could fill the gap between simple string matching and resource-intensive LLM-based solutions.

The main research objectives:

– Develop and adapt a methodology for applying SLMs for automated verification of compliance with WCAG 2.5.3 “Label in Name” criterion in web interfaces.

– Evaluate the effectiveness of SBERT models for semantic analysis of relationships between visible text labels and accessible names in the context of WCAG 2.5.3 criterion.

– Conduct fine-tuning of a selected SLM for 3-class classification of relationships between visible text labels and accessible names (“similar”, “unrelated”, “opposite”).

– Validate the developed solution by evaluating its performance on synthetic datasets, as well as on real data from the Top 500 most visited websites (Top500) and Ukrainian university websites (UaUniv).

Methodology

Dataset Preparation. The research involved both synthetic and real datasets to evaluate and train SLMs. Specifically, synthetic datasets (English – 7,200 samples, Ukrainian – 5,615 samples) were created using leading LLMs (Anthropic Claude, OpenAI ChatGPT, Google Gemini, Grok 3). A diverse range of models was used to increase input data variety and minimize potential biases. To ensure meaningful control over generated samples, a taxonomy of semantic changes was developed beforehand, describing typical text modifications in web content with Accessible Rich Internet Applications (ARIA) attributes. Its application allowed for systematizing change types and ensuring the relevance of synthetic samples.

The taxonomy classifies differences between visible text and its ARIA description, considering both the nature of changes and their potential impact on web resource accessibility and security. Categories include context expansion (“Submit” → “Submit registration form”), action object changes (“Submit payment” → “Submit order”), action type changes (“Save” → “Save and delete”), negation (“Submit” → “Do not submit”), technical modifications (“Submit form” → “SUBMIT FORM”), etc.

In addition to synthetic datasets, the study also utilized real datasets that reflect practical samples of semantic discrepancies in web content:

– Top500 – contains 380 samples of differences between visible text and its representation for assistive technologies. Data was collected from pages of the 500 most popular websites according to the Moz ranking [11], ensuring representation of contemporary publicly accessible internet content.

– UaUniv – includes 319 samples collected from the main pages of official websites of Ukrainian higher education institutions. Data collection was conducted as part of an accessibility study in January 2024 [12], allowing assessment of text information presentation specifics in the educational segment of the Ukrainian internet space.

For training classification models based on SLMs, input data was pre-annotated using the LLM-as-Judge approach [13]. In this approach, large language models serve as expert evaluators, enabling the scaling of the annotation process without requiring extensive human resources. Queries to LLMs consisted of instructions (system prompt) and user data – pairs of visible text and ARIA labels. The model evaluated semantic similarity between these elements on a scale from –1.0 to 1.0, where –1.0 indicates complete opposition or content contradiction, 0 means no connection, and 1.0 represents complete semantic correspondence. Intermediate values reflected partial correspondence, including cases of context expansion, changes in object or action type. To ensure annotation reliability, consistency checks were performed on ratings generated by different LLMs. The numerical ratings obtained from LLMs in the range [–1.0, 1.0] were quantized into three semantic correspondence categories:

1. "Similar" (same/similar) – texts have identical or very close content (LLM ratings within [0.65, 1.0]).

2. "Not related" (not related) – texts have no semantic connection (ratings near zero).

3. "Opposite" (opposite) – texts have opposite or contradictory meanings (ratings within [–1.0, –0.15]).

Results obtained in previous research stages allowed us to create a high-quality annotated dataset that can be used for model training and evaluation. This is an important resource, especially considering the task complexity and the need for high-quality annotations for effective machine learning in the field of semantic text analysis.

As part of the research, more economical models were tested for their suitability for semantic text comparison tasks. The following models were tested: mistral/ministral-8b, qwen/qwen2.5-coder-7b-instruct, meta-llama/llama-3.1-8b-instruct, amazon/nova-micro-v1, liquid/lfm-3b, openai/gpt-4.1-nano, google/gemini-2.5-pro-exp-03-25, liquid/lfm-7b. However, none of these models demonstrated sufficient effectiveness for accurate analysis of semantic relationships between visible text labels and accessible names. This indicates the need to use more powerful models or additional fine-tuning to achieve acceptable results in this domain.

Using SBERT for Semantic Similarity. One promising approach to solving the semantic text comparison problem is using SBERT [14] – a specialized Python module for working with modern vector representation models and rerankers. This framework provides access to creating, using, and training state-of-the-art models for computing text embeddings. Sentence Transformers can be used both for computing vector representations of texts using Sentence Transformer models and for calculating text similarity metrics using Cross-Encoder models. This opens a wide range of applications, including semantic search, determining semantic textual similarity, and paraphrase detection.

Over 10,000 pre-trained Sentence Transformer models are available on the Hugging Face platform for immediate application, including many modern models from the Massive Text Embeddings Benchmark (MTEB) [15] ranking. Additionally, the framework makes it easy to train or fine-tune custom embedding models or rerankers, enabling the creation of specialized models for specific use cases. Particularly promising is the application of this toolkit for Semantic Textual Similarity (STS) tasks. In this approach, vector representations are created for all analyzed texts, after which similarity metrics between them are calculated. Text pairs with the highest similarity score are considered semantically closest.

Testing on a synthetic dataset showed that baseline SBERT models rank text similarity well. However, standard distance metrics (Euclidean, cosine similarity) do not effectively distinguish texts with opposite content, often classifying them in the same category as unrelated texts. This is a significant limitation for our task. Figure 1 shows classification accuracy by category for baseline SBERT models, illustrating this problem.

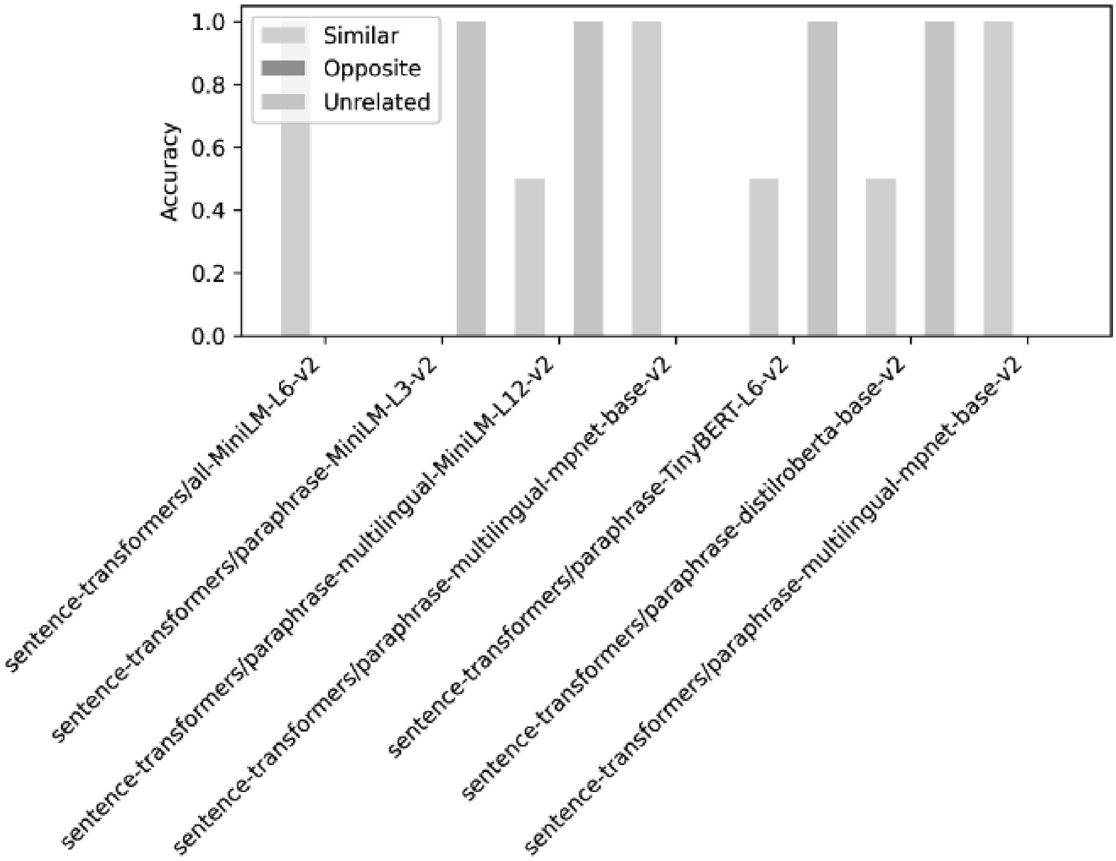


Fig. 1. Per-Category Classification Accuracy for basic SBERT

Hugging Face Tasks and Model Selection for Fine-Tuning. To find a compromise between computational efficiency and result quality, we turned to the Hugging Face ecosystem, which provides ready-made pipelines for natural language processing tasks. The key advantage of this approach is that most pre-trained models are relatively small and can be run locally, significantly reducing their usage cost compared to APIs of large models.

In our research, we considered several types of tasks available in Hugging Face:

– zero-shot-classification – a method that allows classifying texts by categories without seeing examples of these categories during training;

– text-classification – the traditional approach to assigning text to predefined categories;

– fill-mask – a task where the model fills in missing words in text, which can be used to evaluate semantic proximity;

– question-answering – the model answers questions based on context, which can potentially be adapted for text comparison;

– text-generation – creating new text that can be used for paraphrasing and subsequent comparison.

For these tasks, we tested a wide range of models of various architectures and sizes: “google/flan-t5-large”, “google/electra-large-discriminator”, “facebook/bart-large-mnli”, “roberta-large-mnli”, “l-yohai/bigbird-roberta-base-mnli”, “cross-encoder/nli-distilroberta-base”, “distilbert-base-uncased-finetuned-sst-2-english”, “bert-base-uncased”, “albert-base-v2”, “roberta-base”, “distilbert-base-cased-distilled-squad”, “deepset/roberta-base-squad2”, “google/electra-large-discriminator”, “EleutherAI/gpt-neo-125M”, “gpt2”, “distilgpt2” and “facebook/opt-350m”. Testing was conducted taking into account the architectural features of each model and the specifics of the corresponding pipelines. Figure 2 presents a comparison of different models by size (in megabytes) and achieved accuracy on the synthetic dataset.

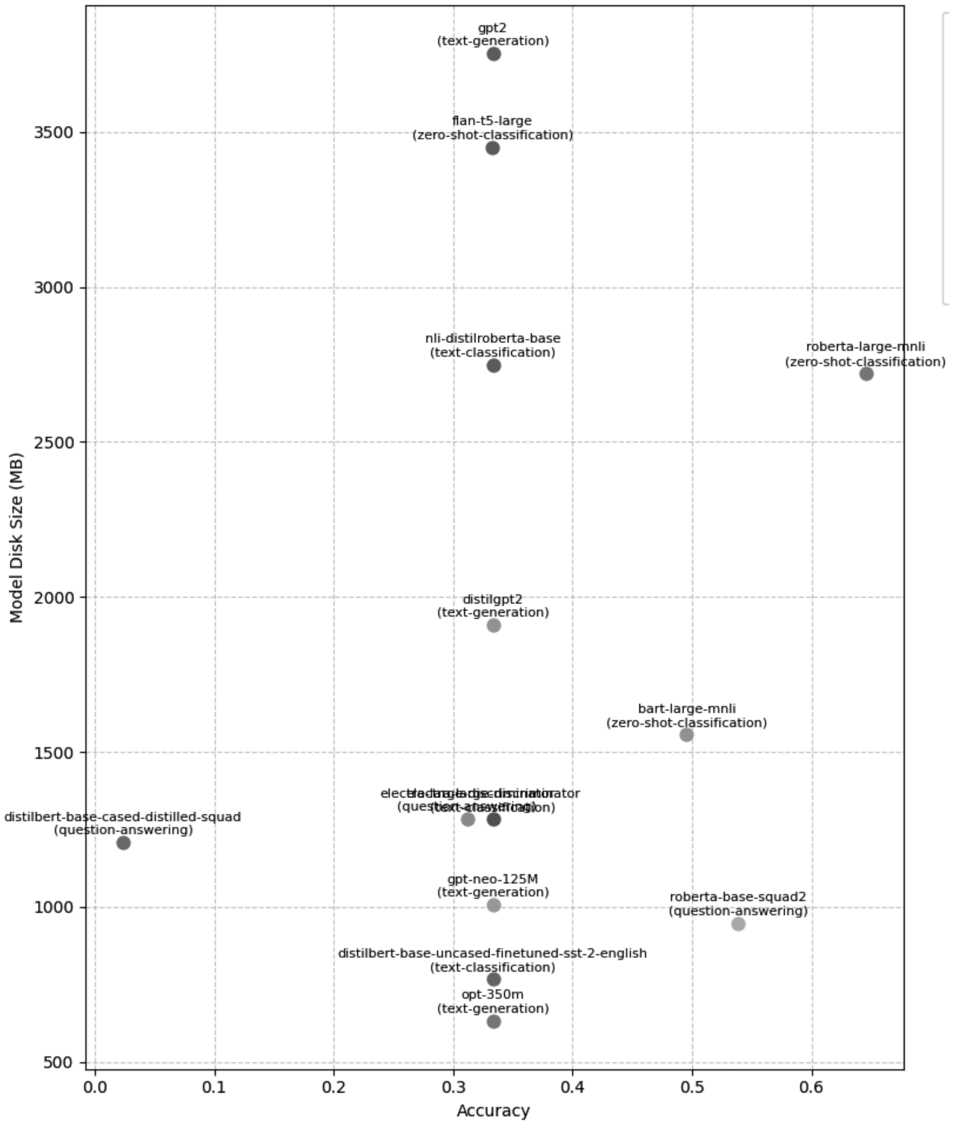


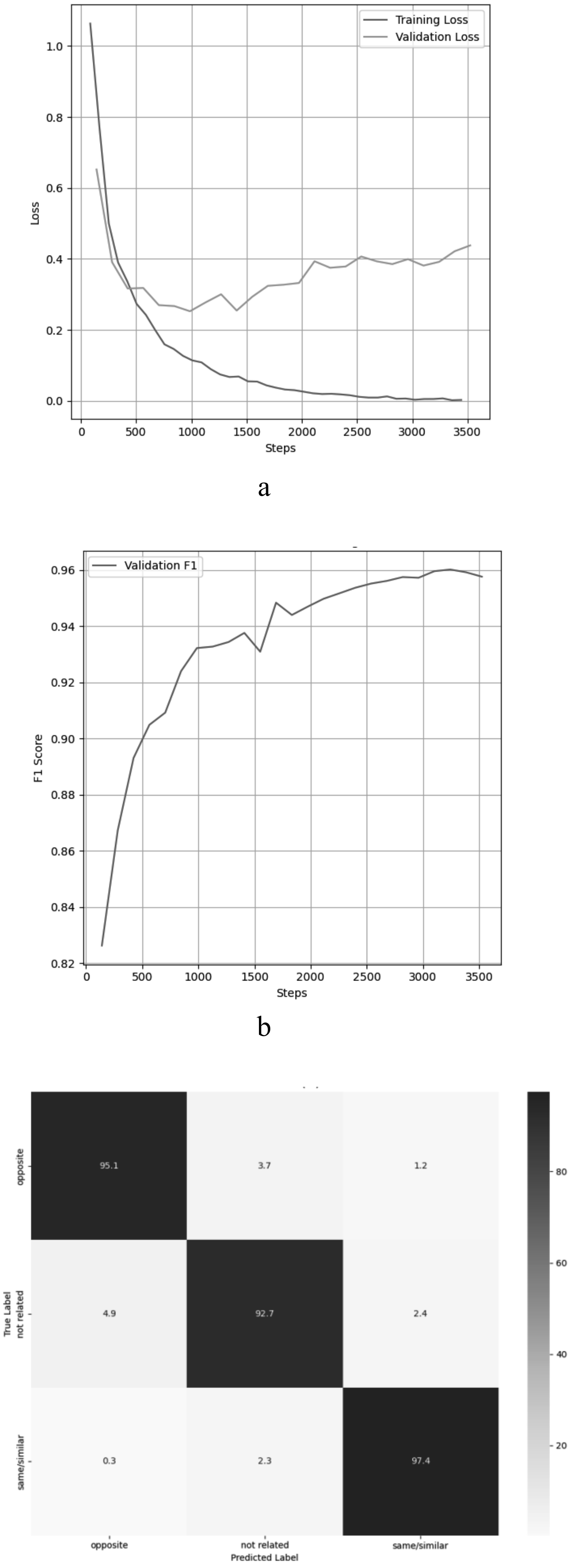
Fig. 2. Model Performance
(Accuracy vs. Size)

Analysis showed that none of the tested combinations of ready-made models and pipelines fully met the task requirements: models either demonstrated insufficient accuracy or were too large for efficient local use. Therefore, a decision was made to fine-tune a model. For this purpose, google/electra-base-discriminator [16] was chosen in combination with the text-classification pipeline as the most promising in terms of balance between size, potential accuracy, and computational requirements.

Fine-tuning of the google/electra-base-discriminator model was conducted on a mixed dataset that included English and Ukrainian synthetic samples. The dataset was augmented by swapping texts in pairs, considering the symmetry of the similarity function. Thus, the training set contained 17,941 samples, and the validation set – 7,689 samples.

Results

As a result of fine-tuning the google/electra-base-discriminator model, we obtained a model with a size of 437 MB. On the validation set (from synthetic data), the model achieved the following metrics: F1-score = 0.96, Recall = 0.96. The results of model fine-tuning are shown in Figure 3.



a

b

c

Fig. 3. Loss (a), F1-Score (b), and Confusion Matrix (c)

The Loss graph demonstrates a stable decrease in both training and validation losses to a level of ≈0.2–0.3 after 2000–2500 steps, indicating effective optimization. The Validation F1-Score increased to 0.96, reflecting high classification accuracy. The confusion matrix confirms accuracy for the classes “opposite” (95.1 %), “not related” (92.7 %), and “same/similar” (97.4 %).

Testing of the fine-tuned model on real data from the Top500 and UaUniv datasets showed acceptable accuracy. The results are presented in Table 1.

Table 1.

Results of testing the fine-tuned model on real data

	Top500	**UaUniv**
Accuracy	0.76	0.72
Precision	0.79	0.74
Recall	0.76	0.72
F1-score	0.77	0.73

Discussion

The article addresses a fundamental contradiction in the comparison of visible text and text for assistive technologies, where attention is often focused solely on formal compliance, such as “The aria-label text begins with the text of the visible label”. This compliance is easily verifiable algorithmically but fails to account for the semantic content of the texts. Basic methods, such as fuzzy string comparison (e.g., Levenshtein distance), are incapable of considering the semantic content of texts. This makes them unsuitable for the task of evaluating semantic similarity, as they may ignore significant differences or incorrectly classify texts as similar when their content differs. As evidence, in the Top500 dataset of 382 samples, 287 (75 %) were identified by LLM as semantically similar but formally violate the WCAG 2.5.3 criterion for algorithmic methods. Similarly, in the UaUniv dataset, 149 out of 320 samples (46 %) were classified by LLM as similar but did not meet the criterion for basic methods. These results demonstrate that basic methods cannot provide adequate semantic analysis, therefore comparison with them was not conducted in this study.

The research results demonstrate that micro- and nano-language models, particularly the fine-tuned google/electra-base-discriminator model, can be effectively used for automated verification of compliance with the WCAG 2.5.3 criterion. The transition from a detailed continuous scale of semantic similarity assessment from –1.0 to 1.0, which was used for data annotation by large language models (where, recall, 1.0 meant complete semantic correspondence, and –1.0 meant opposition), to a more generalized 3-class classification (“similar”, “unrelated”, “opposite”) proved to be a successful approach for training SML. This allowed for high accuracy on synthetic data (F1 = 0.96).

The initial investigation of SBERT models confirmed their ability to rank similarity but also revealed limitations in clearly distinguishing semantically opposite texts using standard distance metrics. This highlighted the need for more specialized approaches, such as fine-tuning for a specific classification task.

The performance of the fine-tuned model on real datasets Top500 (F1 = 0.77) and UaUniv (F1 = 0.73) is somewhat lower than on synthetic data. This is expected, as real data often contains greater diversity and complexity of samples than synthetic data. However, the achieved indicators are still sufficiently high for practical application, especially considering the significantly lower computational resources required for SLM compared to LLM. Analysis of real websites showed that the vast majority of detected errors were related not so much to subtle semantic nuances between visible text and accessible name, but to fundamentally incorrect markup. This often made the use of assistive technologies extremely difficult or even impossible, rather than merely causing confusion due to semantic discrepancies. Among unexpected patterns, it is worth highlighting the contextual sensitivity and certain language independence of SLM, which are positive aspects.

Research Limitations. The analysis of real data was limited to the Top500 and UaUniv datasets, which, although representative, do not cover the entire spectrum of websites. The effectiveness of SLM largely depends on the quality of data for fine-tuning and the fine-tuning process itself.

Despite the achieved results, it is important to remember that automated accessibility testing, even using advanced models similar to those proposed, is not a "silver bullet." It effectively identifies technical compliance with standards but cannot fully replace human verification for evaluating context, complex interactions, and overall user experience [3]. Thus, the presented research contributes to this important field by offering a solution that simplifies accessibility testing and reduces barriers to its implementation, but the best results are achieved when combining automated methods with manual expertise.

Practical Significance. The obtained results confirm that SLMs can serve as a foundation for developing new, more accessible, and efficient tools for automated web accessibility verification. This allows for the integration of semantic analysis into development processes without excessive resource expenditure.

The application of specialized AI tools developed by accessibility experts can significantly improve the testing process and expand its coverage [6]. Thus, the presented study of micro- and nano-language models contributes to this important field by offering a solution that simplifies accessibility testing and reduces barriers to its implementation.

Conclusion

The research demonstrated the effectiveness of using micro- and nano-language models for automating the verification of semantic compliance according to the WCAG 2.5.3 criterion “Label in Name”.

Key findings:

1. SBERT models are useful for obtaining vector representations of texts and initial similarity ranking; however, standard metrics do not reliably distinguish semantically opposite texts.

2. Fine-tuning a relatively small model (google/electra-base-discriminator, 110M parameters) for the task of 3-class classification (“similar”, “unrelated”, “opposite”) allowed for high accuracy (F1 = 0.96) on synthetic data and sufficient accuracy (F1 up to 0.77) on real data (Top500, UaUniv).

3. SLMs are significantly more compact and less resource-intensive compared to LLMs, making them suitable for local deploy-

ment and integration into various development tools.

4. The developed approach offers a practical solution for improving automated accessibility testing, complementing existing tools with semantic analysis capabilities.

Despite the achieved results, it is important to remember the limitations of SLMs and the necessity of human verification in complex cases. Further research may be directed toward expanding training datasets, investigating other SLM architectures and fine-tuning methods, as well as integrating the developed models into comprehensive accessibility testing systems. This will contribute to creating a more accessible web environment for all users.

To ensure the reproducibility of our research, we publish our artifacts on Kaggle [17].

References

1. Web Content Accessibility Guidelines (WCAG) 2.1 [Electronic resource]. 2025. URL: https://www.w3.org/TR/WCAG21/ (accessed: 01.06.2025).
2. Suarez C. Comprehensive Guide to Automated Accessibility Testing [Electronic resource]. 2024. URL: https://kobiton.com/blog/comprehensive-guide-to-automated-accessibility-testing/ (accessed: 01.06.2025).
3. Prasad M. DigitalA11Y. Automated Accessibility Testing Is Not a Sil-ver Bullet [Electronic resource]. 2025. URL: https://www.digitala11y.com/automated-accessibility-testing-is-not-a-silver-bullet/ (accessed: 01.06.2025).
4. Wieland R. Limitations of an Automated-Only Web Accessibility Plan [Electronic resource]. 2024. URL: https://allyant.com/blog/limitations-of-an-automated-only-web-accessibility-plan/ (accessed: 01.06.2025).
5. Intelligence Community Design System. Limitations of automated testing [Electronic resource]. URL: https://design.sis.gov.uk/accessibility/testing/automated-testing-limitation (accessed: 01.06.2025).
6. Barrell N. Enhancing Accessibility with AI and ML [Electronic resource]. 2023. URL: https://www.deque.com/blog/enhancing-accessibility-with-ai-and-ml/ (accessed: 01.06.2025).
7. Du N. et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts // Proc Mach Learn Res. ML Research Press, 2021. Vol. 162. P. 5547–5569.
8. Achary S. The Rise of Small Language Models: A New Era of AI Accessibility and Efficiency [Electronic resource]. 2024. URL: https://medium.com/small-language-models/the-rise-of-small-language-models-a-new-era-of-ai-accessibility-and-efficiency-752322d82656 (accessed: 01.06.2025).
9. Aralimatti R. et al. Fine-Tuning Small Language Models for Domain-Specific AI: An Edge AI Perspective. Preprints, 2025.
10. Abdin M. et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. 2024.
11. Moz. Top 500 Most Popular Websites [Electronic resource]. URL: https://moz.com/top500 (accessed: 01.06.2025).
12. Титов П.О., Шовкопляс О.А., Кузіков Б.О. Аналіз вебдоступності сайтів українських закладів вищої освіти // Системні дослідження та інформаційні технології. 2025. № 2.
13. Gu J. et al. A Survey on LLM-as-a-Judge. 2024. Vol. 1.
14. Reimers N., Gurevych I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation // EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics (ACL), 2020. P. 4512–4525.
15. Muennighoff N. et al. MTEB: Massive Text Embedding Benchmark // EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference. Association for Computational Linguistics (ACL), 2022. P. 2006–2029.
16. Clark K. et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators // 8th International Conference on Learning Representations, ICLR 2020. International Conference on Learning Representations, ICLR, 2020.
17. Tytov P. Semantic Language Models for WCAG [Electronic resource]. URL: https://www.kaggle.com/datasets/tytovpavel/semantic-language-models-for-wcag (accessed: 01.06.2025).

Одержано: 06.06.2025
Внутрішня рецензія отримана: 13.06.2025
Зовнішня рецензія отримана: 16.06.2025

Про авторів:

Кузіков Борис Олегович,
к.т.н., доцент
https://orcid.org/0000-0002-9511-5665
b.kuzikov@cs.sumdu.edu.ua

Шовкопляс Оксана Анатоліївна
к.ф.-м.н., доцент
https://orcid 0000-0002-4596-2524
o.shovkoplyas@mss.sumdu.edu.ua

Титов Павло Олегович
здобувач ступеня доктора філософії
https://orcid.org/0009-0003-6911-5463
stegaspasha@gmail.com

Шовкопляс Сергій Ростиславович
здобувач ступеня доктора філософії
https://orcid.org/0000-0003-1837-0213
s.shovkoplyas@student.sumdu.edu.ua

Місце роботи та навчання авторів:

Сумський державний університет,
кафедра комп’ютерних наук,
40007, м. Суми, вул. Харківська, 116,
тел. +38(0542)687776