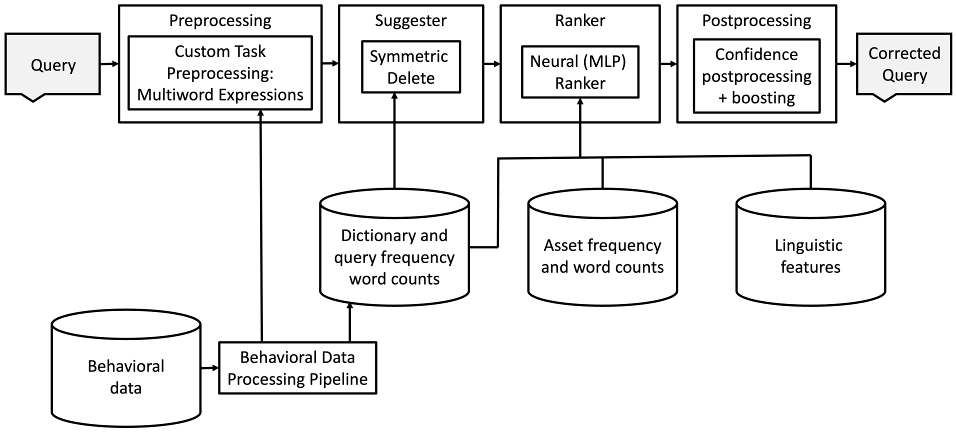

Figure 2: Spellcheck Service Architecture. The MWE module handles task-specific multi-word expressions before the suggester and ranker are called. Behavioral pipelines keep features updated. The postprocessor enables task-specific confidence boosting.

5.1 Adobe Express User Queries

We performed a quantitative analysis on user queries from Adobe Express, a web-based product to create assets from templates. We generated a misspelling dataset from Adobe Express queries by mining queries using the commonly misspelled words from the Wikipedia and Aspell datasets. Additionally we added synthetic perturbations on the mined queries based on common misspellings for each of the 3 languages under consideration (English, French, German). Finally, high frequency spelling errors seen in the application were added via human annotation. There are 6355 queries for English, 1187 for German, and 1128 for French.

This dataset is very different from the dataset that our model was trained on (section 2) but uses dictionaries from the same distribution. This gives us a better representation of real world performance across domains. We tested the performance against NeuSpell (a state-of-the-art neural spelling model) [9] and Aspell (a widely used speller) [1]. As shown in Table 3, our approach outperforms off-the-shelf state-of-the-art approaches in our specific domain, while taking a fraction of the time (under 1 ms on average as opposed to 40+ ms).

Model	Accuracy			Latency
	English	French	German	(ms)
Aspell	51.6%	60.8%	29.7%	40
Neuspell	75.5%	37.5%	36.6%	50
Ours	81.7%	85.0%	84.8%	<1

Table 3: Accuracy and latency of different spell correction models on the Adobe Express query dataset

5.2 Adobe Creative Cloud Home User Queries

We performed a qualitative analysis on user queries from Adobe Creative Cloud Home, one of the main gateways for users to search about Adobe products. We utilized English queries from a single day. The evaluation set comprised 7123 unique queries and their frequencies.

We crowd-sourced and manually checked the correctness of the response from the speller. Results are depicted in Table 4. Nearly 50% of all unique queries entered by users contained a spelling error, highlighting the need for a task-specific speller. Most of the common spelling errors revolved around product names with the words "creative" or "acrobat" being spelled incorrectly in many different ways. For this application, having higher boosting for Adobe product name candidates led to better results due to the nature of the queries, highlighting the need for application-specific contextual signals.

Model	Recall	Precision	Accuracy
Aspell	29.5%	98.9%	45.5%
Neuspell	57.6%	84.2%	75.7%
Ours	96.4%	87.3%	82.2%

Table 4: Accuracy metrics on the Creative Cloud Home dataset. Recall is the rate of incorrect queries that have been properly corrected. Precision is the rate of corrected queries where the correction is correct.

6 CONCLUSIONS AND NEXT STEPS

In this paper we described a novel approach for creating a fast, multilingual spellchecker for user queries. This includes a novel, low latency architecture for deploying and scaling the spellchecker. The resulting speller shows significant improvement over widely available state-of-the-art spellcheckers for short user queries.

Next steps focus on two areas. The first is using the English, French, and German spellers to replace the current production query-time spellers given their success in offline spell correction for autocomplete. The second is extending the speller to ~10 and eventually ~35 languages in order to cover the primary languages used in our search applications. This will allow us to use the same high-quality, custom-tuned, low-latency speller for all query spell correction, both offline for autocomplete suggestions and online for user queries.

ADOBE COMPANY PORTRAIT

Adobe Inc. enables customers to change the world through digital experiences and creativity. The Adobe search and discovery team supports search and recommendations across customer text, image, video, and other document types as well as over Adobe Stock assets and Adobe help and tutorials.

MAIN AUTHOR BIO

Sanat Sharma is a senior machine learning engineer at Adobe Inc. He earned his Master's degree from University of Texas, Austin in 2020, with a focus on NLP. Sanat's work focuses on search improvements and contextual recommendations, and his work has been published at conferences such as CVPR.

REFERENCES

[1] Kevin Atkinson. 2018. GNU Aspell Manual. (2018). http://aspell.net/man-html/index.html Online documentation and code.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching Word Vectors with Subword Information. In *Transactions of the Association for Computational Linguistics*, Vol. 5. MIT Press, Cambridge, MA, 135–146. https://doi.org/10.1162/tacl_a_00051

[3] Anna Bosman, S. Graaff, and Martine Gijsel. 2005. Double Dutch: The Dutch spelling system and learning to spell in Dutch. In *Handbook of Orthography and Literacy*, R. Malatesha Joshi and P. G. Aaron (Eds.). Routledge, 135–150.

[4] Qing Chen, Mu Li, and Ming Zhou. 2007. Improving Query Spelling Correction Using Web Search Results. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, 181–189. https://aclanthology.org/D07-1019

[5] Monojit Choudhury, Markose Thomas, Animesh Mukherjee, Anupam Basu, and Niloy Ganguly. 2007. How Difficult is it to Develop a Perfect Spell-checker? A Cross-Linguistic Analysis through Complex Network Approach. In *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*. Association for Computational Linguistics, Rochester, NY, USA, 81–88. https://aclanthology.org/W07-0212

[6] Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A Large Scale Ranker-Based System for Search Query Spelling Correction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Coling 2010 Organizing Committee, Beijing, China, 358–366. https://aclanthology.org/C10-1041

[7] Wolf Garbe. 2012. 1000x Faster Spelling Correction algorithm. (2012). https://seekstorm.com/blog/1000x-spelling-correction/ SeekStorm blog post.

[8] Zahurul Islam, Md Uddin, and Mumit Khan. 2007. A Light Weight Stemmer for Bengali and its Use in Spelling Checker. In *Proceedings of the 1st International Conference on Digital Communications and Computer Applications (DCCA2007)*. 87–93. Center for Research on Bangla Language Processing, BRAC University.

[9] Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. NeuSpell: A Neural Spelling Correction Toolkit. *CoRR* abs/2010.11085 (2020). arXiv:2010.11085 https://arxiv.org/abs/2010.11085

[10] Alex Kuznetsov and Hector Urdiales. 2021. Spelling Correction with Denoising Transformer. https://doi.org/10.48550/ARXIV.2105.05977

[11] Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10, 8 (1966), 707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[12] Casey Whitelaw, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. ACM, 890–899. https://dl.acm.org/doi/10.5555/1699571.1699629