KNN-ConcritUp! A Language-Agnostic, Interactive, and Interpretable System for Extrapolating Concreteness Norms

Sven Naber[1], Marina Aziz[1], Diego Frassinelli[2], and Sabine Schulte im Walde[1]

[1]Institut für Maschinelle Sprachverarbeitung (IMS), Universität Stuttgart
[2]MaiNLP, Centrum für Informations und Sprachverarbeitung, LMU München
{sven.naber, marina.aziz, schulte}@ims.uni-stuttgart.de
frassinelli@cis.lmu.de

Abstract

We present KNN-ConcritUp, a browser-based demo system that extrapolates language-agnostic concreteness scores (as well as scores for further behavioral variables) from a small-scale rated seed lexicon to a large-scale target vocabulary, using k-nearest-neighbor regression over word embeddings. The contribution is an easy-to-use workflow for configuring runs, selecting hyperparameters, inspecting coverage, exporting artifacts, and auditing predictions through nearest-neighbor evidence. For per-language indexed runs in our repository snapshot (heterogeneous datasets and settings, no controlled multilingual benchmark), held-out Spearman correlations range from $\rho = 0.761$ to $\rho = 0.882$.

1 Introduction

Concreteness describes how strongly a word refers to perceivable entities or experiences (Paivio, 1971; Brysbaert et al., 2014b). For example, nouns such as *apple* and *chair* are typically perceived as referring to more concrete concepts than nouns such as *justice* and *freedom*. This distinction is central in psycholinguistics research: concrete concepts are easily grounded in perceptual features, whereas abstract concepts tend to depend more strongly on linguistic and contextual information (Pecher et al., 2011; Borghi et al., 2017). As a result, concreteness scores are widely used in studies of lexical processing and memory effects, as well as for stimulus selection criteria and as features in cognitive and computational models (Spreen and Schulz, 1966; Paivio et al., 1968; Balota and Neely, 1980; Walker and Hulme, 1999; Brysbaert et al., 2014b).

However, a major limitation of existing concreteness resources is their limited coverage. While large English norms exist, resources for other languages are growing only slowly: available norms for German (Lahl et al., 2009; Kanske and Kotz, 2010), Italian (Montefinese et al., 2014), Chinese (Yao et al., 2017), French (Bonin et al., 2018), and Croatian (Peti-Stantić et al., 2021) contain only between 1,000 and 6,000 words. Larger collections exist for Dutch (Brysbaert et al., 2014a) with 30,000 words and Estonian (Proos and Aigro, 2023) with 36,000, but many languages still lack large pools of reliable ratings. This represents a crucial problem for large-scale studies, where researchers often need reliable ratings for many more words and languages than are currently available.

To fill this gap, we developed a language-agnostic system KNN-ConcritUp for extrapolating concreteness norms with transparent diagnostics, which is also applicable to the norming of further behavioral variables. Semi-automatic procedures to extend human rating collections have been used before (Turney et al., 2011; Köper and Schulte im Walde, 2016; Köper and Schulte im Walde, 2017; Aedmaa et al., 2018; Ljubešić et al., 2018; Rabinovich et al., 2018; Charbonnier and Wartena, 2020; Knupleš et al., 2026), and the use of k-nearest neighbors (k-NN) for concreteness predictions is also not novel in itself: prior work has shown that neighborhood-based extrapolation can successfully generate ratings to unseen words (Bestgen and Vincze, 2012; Mandera et al., 2015; Aziz, 2025). Building on these findings, our contribution is to turn this prediction strategy into a transparent, reproducible, and user-friendly system for large-scale norm extrapolation. Given a seed lexicon and an embedding space for a given language, KNN-ConcritUp assigns concreteness scores to any arbitrary vocabulary specified by the user. Our system also provides an interactive browser interface, file-based configuration for reproducible runs, automated model selection, coverage reporting, and inspectable neighbor evidence. In this way, KNN-ConcritUp supports deterministic, replicable extrapolation and allows users to trace each prediction back to the original neighbors on which it is based.

2 Background

Concreteness norms. For creating concreteness norms, experiment participants assign numerical ratings to words in the abstract–concrete continuum. Early resources measured concreteness, imageability, and related lexical variables for smaller word sets (Spreen and Schulz, 1966; Paivio et al., 1968; Coltheart, 1981; Warriner et al., 2013); later, crowdsourced collections expanded coverage to tens of thousands of lemmas (Brysbaert et al., 2014b; Mohammad, 2018; Lynott et al., 2020). In natural language processing (NLP), concreteness has been used as predictor in metaphor processing (Turney et al., 2011; Tsvetkov et al., 2014; Köper and Schulte im Walde, 2016; Alnafesah et al., 2020; Maudslay et al., 2020; Piccirilli and Schulte im Walde, 2022a,b; Hülsing and Schulte im Walde, 2024; Khaliq et al., 2024; Eichel et al., 2026; Knupleš et al., 2026), document and lexical complexity prediction (Tanaka et al., 2013; North et al., 2023), and embodied agents and robots (Cangelosi and Stramandinoli, 2018; Rasheed et al., 2018; Ichter et al., 2023), among other tasks.

Embedding-based extrapolation. Extrapolation methods rely on the assumption that words that are close in an embedding space are likely to have similar lexical ratings. In this way, an unrated word can be scored using the ratings of its nearest neighbors. This assumption has been used in prior work with similarity indices, anchor-word methods, k-NN regression, support-vector regression, neural models, and multilingual embeddings (Bestgen and Vincze, 2012; Mandera et al., 2015; Köper and Schulte im Walde, 2016; Ljubešić et al., 2018; Plisiecki and Sobieszek, 2023; Aziz, 2025). KNN-ConcritUp uses k-NN because it is computationally efficient and makes predictions interpretable by looking at the selected neighbors.

3 System Overview

As the basis for extrapolating norms from gold-standard ratings, gold files can be uploaded directly to the system. KNN-ConcritUp then runs the prediction pipeline in the background, while user interactions and outputs are handled via a browser interface. The graphical user interface (GUI) is implemented as a Flask application. The model backend uses scikit-learn k-NN regression with cosine distance and brute-force search (Pedregosa et al., 2011).

Overall, the system expects as input: a CSV/TSV gold ratings file, word and score column names, an embedding file or discovered embedding selection, an embedding type, and a target vocabulary with one token per line. The current embedding loaders support fastText `.bin` files and word2vec-style text files, including `.vec` embeddings (Mikolov et al., 2013; Bojanowski et al., 2017). Optional settings, such as token normalization, part-of-speech filtering, random seed, split ratio, k grid, parallel jobs, and output path, are available in an advanced panel. The weighting mode (`uniform` vs. `distance`) is selected automatically during cross-validation. The system writes repository-relative paths into the active configuration. During execution, the interface reports job state, elapsed times and logs.

After each run, the results view provides access to the generated files, with options to search, preview, and download them. The primary output is `vocab_predictions.csv`; it contains target-word predictions, an `is_gold_observed` overlap flag, neighbor words, neighbor gold scores, and cosine distances. The system also writes `summary.json`, `cv_results.csv`, `test_predictions.csv`, and optional out-of-vocabulary (OOV) files depending on the configuration. The summary records the vocabulary and the number of reference words, making clear which reference set was used. Figures 1 and 2 show the editor and result-inspection pages.

4 Prediction Pipeline

Let $G = \{(w_i, y_i)\}_{i=1}^{n}$ be the gold lexicon, where w_i is a rated word with embedding vector $\mathbf{v}_i$, and gold concreteness score y_i. For a target word t with embedding vector $\mathbf{v}_t$, the system retrieves the k nearest rated words under cosine distance:

$$d_{\cos}(t, w_i) = 1 - \cos(\mathbf{v}_t, \mathbf{v}_i) \tag{1}$$

The system then predicts a weighted average of the corresponding k gold scores. For uniform weighting, $\alpha_i = 1$. For distance weighting:

$$\alpha_i = \frac{1}{\max(d_{\cos}(t, w_i), \varepsilon)}, \quad \varepsilon = 10^{-12} \tag{2}$$

and

$$\hat{y}_t = \frac{\sum_{i \in N_k(t)} \alpha_i y_i}{\sum_{i \in N_k(t)} \alpha_i} \tag{3}$$

Note that `neighbor_cosine_distances` are computed as $1 - \cos(\cdot, \cdot)$, so smaller values indicate more similar embedding directions.

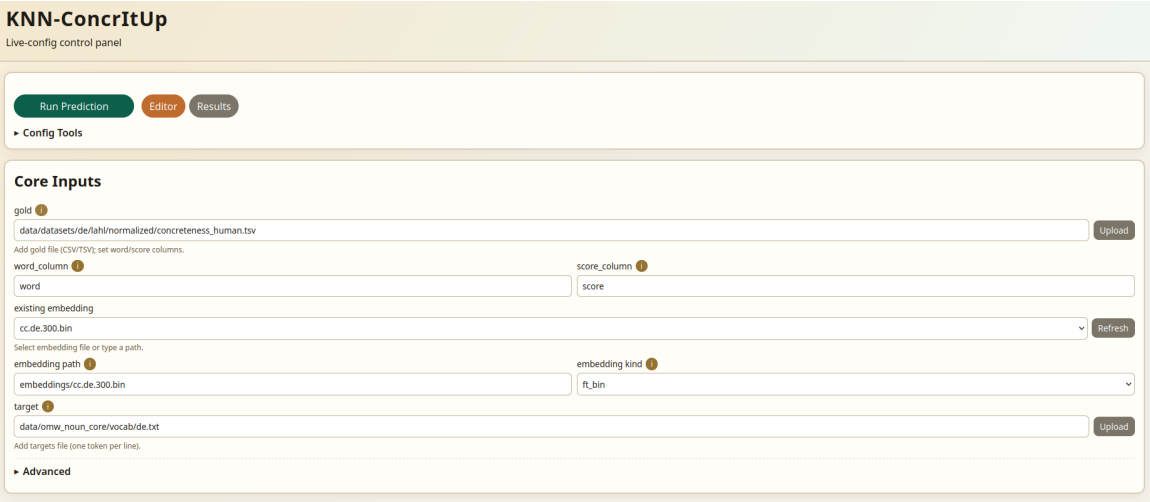

Figure 1: Guided editor view at startup with core inputs.

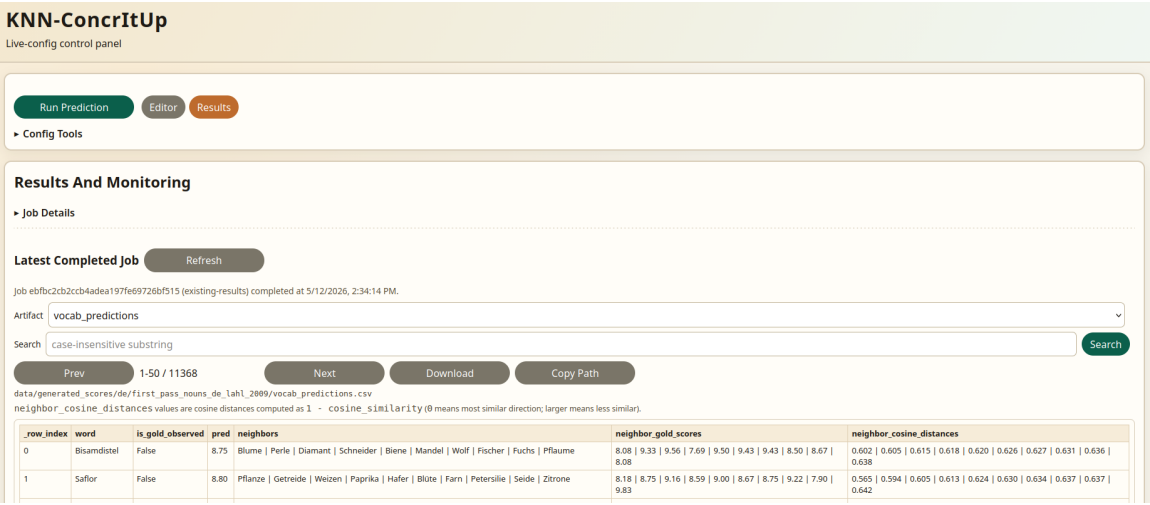

Figure 2: Results view with artifact selection and preview of neighbor diagnostics.

The system first loads the gold table, identifies the columns containing each word and the corresponding score, then it validates the score values, trims the tokens to remove whitespace, and optionally lowercases them. Optional part-of-speech filters are applied before the vector lookup, if the gold data contain suitable tags. Duplicate normalized gold tokens are collapsed deterministically by averaging scores after normalization (by stripping and optional lowercasing them). Then the system looks up all gold and target tokens in the selected embedding space. Gold items without vectors are excluded from the model selection and held-out evaluation but are counted in coverage statistics; target OOV items receive no direct prediction and are written to the OOV artifact.

After this step, the gold items are split into training and held-out test partitions using the configured random seed and split ratio. Cross-validation uses shuffled folds with the same seed. The backend then runs cross-validation over the configured k grid and over `uniform` and `distance` weighting. Candidate k values are capped by fold feasibility, and invalid fold cases are skipped. The selected setting maximizes mean validation Spearman correlation; ties are broken by the lower averaged root mean squared error (RMSE).

For evaluation, the selected regressor is fitted on the training partition and scored on the held-out partition. For vocabulary prediction, the selected configuration is refit on all covered gold items before scoring targets. Target entries that also occur in the gold lexicon are retained in the output but the target is excluded from its nearest-neighbor set during scoring. The summary records this with `vocab_fit_scope=all_covered_gold_post_eval` and `n_vocab_reference_words`.

Language	Gold seed resource	n_{gold}	n_{train}	n_{test}	k^*	ρ_{test}	Scale	$\text{RMSE}_{\text{test}}$
Arabic	Arabic noun concreteness norms (Aziz, 2025)	214	171	43	25	0.795	1–5	0.493
Spanish	Spanish ANEW (Redondo et al., 2007)	612	489	123	65	0.761	1–7	0.724
Croatian	Croatian psycholinguistic norms (Peti Stantić et al., 2018)	1,000	800	200	40	0.790	1–5	0.627
French	French concreteness norms (Bonin et al., 2018)	1,659	1,327	332	20	0.854	1–5	0.533
German	German noun norms (Lahl et al., 2009)	2,654	2,123	531	30	0.863	1–10	0.927
Portuguese	Minho Word Pool (Soares et al., 2017)	3,342	2,673	669	35	0.874	1–7	0.624
English	English concreteness norms (Brysbaert et al., 2014b)	13,132	10,505	2,627	40	**0.882**	1–5	0.476
Dutch	Dutch AoA/concreteness norms (Brysbaert et al., 2014a)	30,009	24,007	6,002	15	0.867	1–5	0.504

Table 1: Validation snapshot from indexed repository outputs. Each run uses one language-specific human-rated seed resource and the corresponding 300-dimensional fastText Common Crawl binary embedding space (Grave et al., 2018). Distance weighting is selected in all currently indexed runs. RMSE is reported on the original rating scale of each resource.

5 Operational Validation

This section provides an overview of the reported metrics for a given configuration. We also compare these metrics across our runs for a variety of languages. Because datasets and vocabularies differ by language, we treat these numbers as operational validation rather than a controlled multilingual benchmark. Each run's summary records the following information:

- **Gold and evaluation size:** `n_gold_words`, `n_test`, and `gold_coverage` report the number of gold items, the held-out test size, and embedding coverage.
- **Selected model configuration:** `best_k` and `best_weights` record the selected number of neighbors and weighting scheme.
- **Held-out performance:** `test_spearman` and `test_rmse` report rank correlation and prediction error on the held-out partition.
- **Vocabulary prediction metadata:** `n_vocab_rows`, `n_vocab_reference_words`, and `vocab_fit_scope` document the size of the predicted vocabulary, and the reference set used for final target scoring.

Shared protocol. All eight runs use the same prediction protocol: `runtime.seed=13`, `prediction.test_size=0.2`, `prediction.cv_folds=5`, `prediction.k_min=5`, `prediction.k_max=100`, `prediction.k_step=5`. The selected configuration is then evaluated on the held-out partition. The reported correlations therefore report on within-resource predictive validity.

n_{train}	k^*	ρ_{test}	$\text{RMSE}_{\text{test}}$
100	25	0.814	0.685
200	20	0.834	0.589
500	45	0.855	0.559
1,000	55	0.867	0.530
2,000	40	0.871	0.506
5,000	50	0.877	0.488
10,000	30	**0.881**	**0.474**

Table 2: English training-data ablation evaluated on the same fixed 2,627-item test set. Smaller train sets are balanced subsets of the next larger set.

Language-specific resources. Each configuration uses one language-specific human-annotated seed lexicon together with a fastText binary embedding space of the same language. Where the spurce resource contains multiple parts-of-speech we restrict gold data to nouns. The seed resources and validation results are listed in Table 1.

In the current snapshot, held-out Spearman ranges from $\rho = 0.761$ to $\rho = 0.882$, with k ranging from 15 to 65. The results show consistently strong correlations across languages, with performance generally higher for larger gold-standard datasets. Because rating scales differ across resources, RMSE values are reported together with the corresponding scale and should not be compared directly across languages.

Training data ablation. To quantify the impact of additional gold labeled data we ran a training size ablation on the English norms (Brysbaert et al., 2014b), cf. Table 2. We selected subsets of gold labeled data by sorting words by gold concreteness and recursively selecting evenly spaced ranks from the next larger subset. Model configuration selection uses again the internal 80/20 split, evaluation is done on the final model fitted on the complete subset. Spearman correlation increases monotonically with gold set size while RMSE decreases.

Word	Nearest rated neighbors	Gold	Pred.
Blume	*Blüte* (8.67), *Tulpe* (9.38), *Pflanze* (8.18), *Knospe* (8.83), *Schmetterling* (8.80)	8.08	8.88
Idee	*Überlegung* (4.00), *Gedanke* (4.11), *Vorstellung* (5.09), *Konzept* (4.42), *Absicht* (4.35)	4.48	4.55
Spirale	*Kugel* (8.76), *Pyramide* (8.47), *Nadel* (9.92), *Kerze* (9.43), *Welle* (7.50)	6.71	8.23

Table 3: German qualitative examples with nearest-neighbor evidence. Numbers in parentheses are human concreteness ratings of the neighboring gold items. For space reasons, only the five nearest neighbors are shown.

We see diminishing returns for larger gold set sizes above 1000 to 2000 annotations.

Qualitative example. Table 3 showcases some predictions and errors. The concrete noun *Blume* 'flower' receives a high concreteness rating based on concrete nearest neighbors such as *Blüte* 'blossom' and *Tulpe* 'tulip'. The abstract noun *Idee* 'idea' is rated more abstract by being close to abstract cognitive nouns like *Überlegung* 'consideration' and *Gedanke* 'thought'. Predictions may also be off, as seen in the case of *Spirale* 'spiral' – whose rating is estimated higher than in the gold data because of highly concrete shape- and object-related neighbors like *Kugel* 'sphere' and *Nadel* 'needle'.

6 Code and Data Availability

Our tool KNN-CONCRITUP is publicly available via https://concr-it-up.ims.uni-stuttgart.de/, the code is available from https://github.com/SNaber/concr-it-up. In addition, Appendix A provides a user walkthrough.

7 Conclusion

KNN-CONCRITUP turns embedding-based concreteness extrapolation into an interactive demo workflow. The current pipeline can produce useful candidate norms across languages and language configurations, while the coverage, held-out metrics, and vocabulary-fit metadata make clear what was validated and what reference set was used for final scoring. The same architecture could also support other scalar lexical norms, such as valence, arousal, imageability, and familiarity, where suitable gold ratings are available.

8 Limitations

Prediction quality depends on the gold data and the embedding space. Small or narrow gold sets produce sparse neighborhoods, and target words missing from the selected embedding cannot be scored directly. Even if a word is covered, the nearest neighbors may reflect frequency effects, domain mismatch, polysemy, or culturally specific usage rather than the intended concreteness signal.

Random held-out splits estimate in-distribution performance, but target vocabularies may come from shifted domains. Users should treat generated scores as model-derived candidate estimates and label them as such.

9 Ethics Statement

The system is intended for lexical resource construction and analysis. Users should ensure that uploaded norm datasets, embeddings, and generated artifacts can be processed and shared under the relevant licenses. Released predictions should include a provenance checklist: gold source, embedding source, embedding version, run date, system version, normalization settings, duplicate-token policy, coverage rates, OOV counts, evaluation protocol, vocabulary fit scope, and whether scores are human-collected or model-derived estimates.

Use of AI Assistants. The authors acknowledge the use of AI assistants solely for correcting grammatical errors, formatting table boundaries, and providing assistance with coding.

Acknowledgments

This research was supported by the DFG Research Grant SCHU 2580/4-1 *MUDCAT – Multimodal Dimensions and Computational Applications of Abstractness*.

References

Eleri Aedmaa, Maximilian Köper, and Sabine Schulte im Walde. 2018. Combining abstractness and language-specific theoretical indicators for detecting non-literal usage of Estonian particle verbs. In *Proceedings of the NAACL 2018 Student Research Workshop*, pages 9–16, New Orleans, LA, USA.

Ghadi Alnafesah, Harish Tayyar Madabushi, and Mark Lee. 2020. Augmenting neural metaphor detection with concreteness. In *Proceedings of the 2nd Workshop on Figurative Language Processing*, pages 204–210, Seattle, Washington (online).

Marina Aziz. 2025. Bridging behavioral gaps: Automatic extrapolation of concreteness norms in Arabic using English-tuned k-NN approaches. Master's thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

David A. Balota and James H. Neely. 1980. Test-expectancy and word-frequency effects in recall and recognition. *Journal of Experimental Psychology: Human Learning and Memory*, 6(5):576–587.

Yves Bestgen and Nadja Vincze. 2012. Checking and bootstrapping lexical norms by means of word similarity indexes. *Behavior Research Methods*, 44(4):998–1006.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Patrick Bonin, Alain Méot, and Aurélia Bugaiska. 2018. Concreteness norms for 1,659 French words: Relationships with other psycholinguistic variables and word recognition times. *Behavior Research Methods*, 50(6):2366–2387.

Anna M. Borghi, Ferdinand Binkofski, Cristiano Castelfranchi, Felice Cimatti, Claudia Scorolli, and Luca Tummolini. 2017. The challenge of abstract concepts. *Psychonomic Bulletin*, 143:263–292.

Marc Brysbaert, Michaël Stevens, Simon De Deyne, Wouter Voorspoels, and Gert Storms. 2014a. Norms of age of acquisition and concreteness for 30,000 Dutch words. *Acta Psychologica*, 150:80–84.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014b. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Angelo Cangelosi and Francesca Stramandinoli. 2018. A review of abstract concept learning in embodied agents and robots. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1752):20170131.

Jean Charbonnier and Christian Wartena. 2020. Predicting the concreteness of German words. In *Proceedings of the joint 5th Swiss Text Analytics Conference and the 16th Conference on Natural Language Processing*, Hanover, Germany.

Max Coltheart. 1981. The MRC Psycholinguistic Database. *Quarterly Journal of Experimental Psychology*, 33A:497–505.

Annerose Eichel, Tonmoy Rakshit, and Sabine Schulte im Walde. 2026. Contextualising (im)plausible events triggers figurative language. In *Proceedings of the Workshop on Learning Non-Literal Expressions with Small Data*, Mallorca, Spain.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan.

Anna Hülsing and Sabine Schulte im Walde. 2024. Cross-lingual metaphor detection for low- to high-resource languages. In *Proceedings of the 4th Workshop on Figurative Language Processing*, pages 22–34, Mexico City, Mexico.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, and 26 others. 2023. Do as I can, not as I say: Grounding language in robotic affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 287–318.

Philipp Kanske and Sonja A. Kotz. 2010. Leipzig affective norms for German: A reliability study. *Behavior Research Methods*, 42(4):987–991.

Mohammed Abdul Khaliq, Diego Frassinelli, and Sabine Schulte im Walde. 2024. Comparison of image generation models for abstract and concrete event descriptions. In *Proceedings of the 4th Workshop on Figurative Language Processing*, pages 15–21, Mexico City, Mexico.

Urban Knupleš, Diego Frassinelli, Alexander Fraser, and Sabine Schulte im Walde. 2026. Literally concrete or figuratively abstract? Multilingual concreteness norms for verb-object expressions. *Transactions of the Association for Computational Linguistics*.

Maximilian Köper and Sabine Schulte im Walde. 2016. Automatically generated affective norms of abstractness, arousal, imageability and valence for 350,000 German lemmas. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 2595–2598.

Maximilian Köper and Sabine Schulte im Walde. 2016. Distinguishing literal and non-literal usage of German particle verbs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 353–362, San Diego, CA, USA.

Maximilian Köper and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30, Valencia, Spain.

Olaf Lahl, Anja S. Göritz, Reinhard Pietrowsky, and Jessica Rosenberg. 2009. Using the World-Wide Web to obtain large-scale word norms: 190,212 ratings on a set of 2,654 German nouns. *Behavior Research Methods*, 41(1):13–19.

Nikola Ljubešić, Darja Fišer, and Anita Peti-Stantić. 2018. Predicting concreteness and imageability of words within and across languages via word embeddings. In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 217–222.

Dermot Lynott, Louise Connell, Marc Brysbaert, James Brand, and James Carney. 2020. The Lancaster sensorimotor norms: Multidimensional measures of perceptual and action strength for 40,000 English words. *Behavior Research Methods*, 52:1–21.

Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2015. How useful are corpus-based methods for extrapolating psycholinguistic variables? *The Quarterly Journal of Experimental Psychology*, 68(8):1623–1642.

Rowan Hall Maudslay, Tiago Pimentel, Ryan Cotterell, and Simone Teufel. 2020. Metaphor detection using context and concreteness. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 221–226.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations Workshop*.

Saif M. Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.

Maria Montefinese, Ettore Ambrosini, Beth Fairfield, and Nicola Mammarella. 2014. The adaptation of the affective norms for English words (ANEW) for Italian. *Behavior Research Methods*, 46(3):887–903.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

Allan Paivio. 1971. *Imagery and Verbal Processes*. Holt, Rinehart and Winston, New York.

Allan Paivio, John C. Yuille, and Stephen A. Madigan. 1968. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of Experimental Psychology*, 76(1):1–25.

Diane Pecher, Inge Boot, and Saskia Van Dantzig. 2011. Abstract concepts. Sensory-motor grounding, metaphors, and beyond. *Psychology of Learning and Motivation – Advances in Research and Theory*, 54:217–248.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Anita Peti-Stantić, Maja Anđel, Vedrana Gnjidić, Gordana Keresteš, Nikola Ljubešić, Irina Masnikosa, Mirjana Tonković, Jelena Tušek, Jana Willer-Gold, and Mateusz-Milan Stanojević. 2021. The Croatian psycholinguistic database: Estimates for 6000 nouns, verbs, adjectives and adverbs. *Behavior Research Methods*, 53(4):1799–1816.

Anita Peti Stantić, Maja Anđel, Gordana Keresteš, Nikola Ljubešić, Mateusz-Milan Stanojević, and Mirjana Tonković. 2018. Psycholinguistic estimates of 3000 words of Croatian: Concreteness and imageability. *Suvremena lingvistika*, 44(85):91–112.

Prisca Piccirilli and Sabine Schulte im Walde. 2022a. Features of perceived metaphoricity on the discourse level: Abstractness and emotionality. In *Proceedings of the 13th International Conference on Language Resources and Evaluation*, Marseille, France.

Prisca Piccirilli and Sabine Schulte im Walde. 2022b. What *Drives* the use of metaphorical language? Negative insights from abstractness, affect, discourse coherence and contextualized word representations. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 299–310, Seattle, Washington, USA. Also non-archival paper at the NAACL 2022 Student Research Workshop.

Hubert Plisiecki and Agnieszka Sobieszek. 2023. Extrapolation of affective norms using transformer-based neural networks and its application to experimental stimuli selection. *Behavior Research Methods*, 56(5):4716–4731.

Mariann Proos and Mari Aigro. 2023. Concreteness ratings for 36,000 Estonian words. *Behavior Research Methods*.

Ella Rabinovich, Benjamin Sznajder, Artem Spector, Ilya Shnayderman, Ranit Aharonov, David Konopnicki, and Noam Slonim. 2018. Learning concept abstractness using weak supervision. arXiv:1809.01285.

Nadia Rasheed, Shamsudin H.M. Amin, Umbrin Sultana, Abdul Rauf Bhatti, and Mamoona N. Asghar. 2018. Extension of grounding mechanism for abstract words: Computational methods insights. *Artificial Intelligence Review*, 50(3):467–494.

Jaime Redondo, Isabel Fraga, Isabel Padrón, and Montserrat Comesaña. 2007. The Spanish adaptation of ANEW (Affective Norms for English Words). *Behavior Research Methods*, 39(3):600–605.

Ana Paula Soares, Ana Santos Costa, João Machado, Montserrat Comesaña, and Helena Mendes Oliveira. 2017. The minho word pool: Norms for imageability, concreteness, and subjective frequency for 3,800 Portuguese words. *Behavior Research Methods*, 49(3):1065–1081.

Otfried Spreen and Richard W. Schulz. 1966. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns. *Journal of Verbal Learning and Verbal Behavior*, 5(5):459–468.

Shinya Tanaka, Adam Jatowt, Makoto P. Kato, and Katsumi Tanaka. 2013. Estimating content concreteness for finding comprehensible documents. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, Rome, Italy.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 248–258, Baltimore, MD, USA.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690.

Ian Walker and Charles Hulme. 1999. Concrete words are easier to recall than abstract words: Evidence for a semantic contribution to short-term serial recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(5):1256–1271.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods*, 45(4):1191–1207.

Zhao Yao, Jia Wu, Yanyan Zhang, and Zhenhong Wang. 2017. Norms of valence, arousal, concreteness, familiarity, imageability, and context availability for 1,100 Chinese words. 49(4):1374–1385.

A User Walkthrough

This appendix illustrates a KNN-CONCRITUP user session.

Step 1: Specify the core inputs. The editor opens in a compact view with the required fields: the gold ratings file, the word and score columns, the embedding source and type, and the target vocabulary. The target vocabulary is expected to contain one token per line. Gold and target files can be uploaded directly through the interface. Embeddings can either be chosen from discovered repository files or entered as a path.

Step 2: Create, load, or save a run configuration. Users can create a new configuration, upload, download or save a JSON configuration for later reruns.

Step 3: Adjust optional advanced settings. Users can enable lowercasing, apply part-of-speech filtering and specify the output directory. Furthermore, they can select a random seed, set the number of parallel jobs, choose the held-out split and cross-validation folds, define the k grid, and select how many nearest neighbors to include in the report outputs.

Step 4: Run the prediction pipeline and inspect the summary. The summary view reports the main metrics, including held-out Spearman correlation, RMSE, selected k, number of gold words, held-out test size, and selected weighting mode.

Step 5: Inspect cross-validation results. The `cv_results.csv` file lists all valid model-selection candidates. For each candidate, the table reports the value of k, the weighting scheme, mean and standard deviation of validation Spearman correlation, and mean and standard deviation of validation RMSE.

Step 6: Inspect vocabulary predictions. The main prediction file contains the target word, the predicted concreteness score, a flag indicating whether the target word was observed in the gold resource, the nearest rated neighbors, their cosine distances and their human gold scores.

Step 7: Search individual predictions. The results view has a case-insensitive substring search over the currently selected output.

Step 8: Inspect held-out errors. The `test_predictions.csv` file reports predictions for the held-out gold items. It contains the gold score, predicted score, signed error, and absolute error.

Step 9: Check out-of-vocabulary outputs. When gold or target items are not covered by the selected embedding space they are written to separate OOV files.

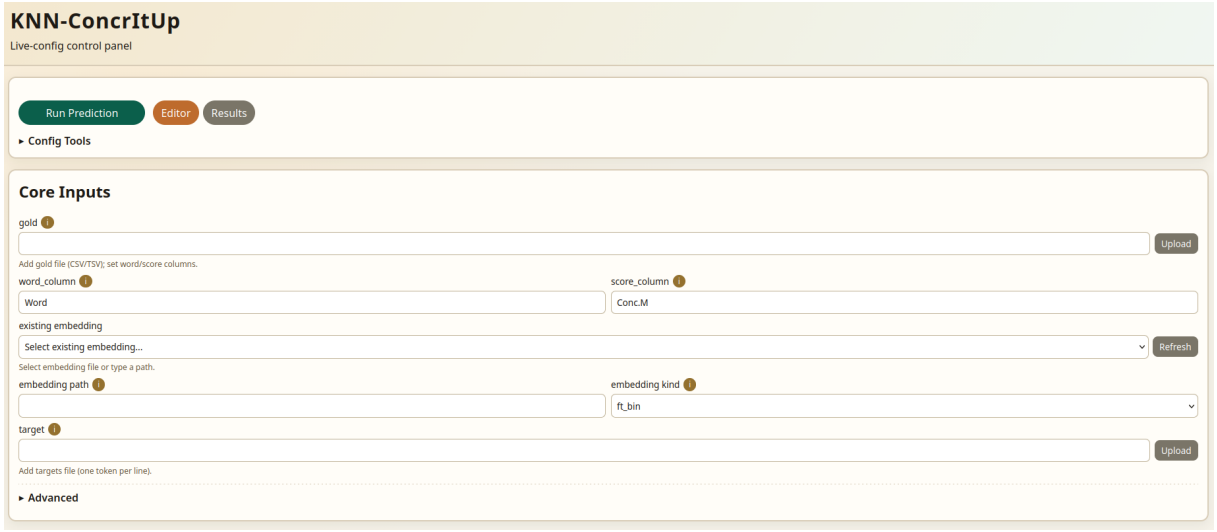

Figure 3: Core input view.

Figure 4: Configuration tools.

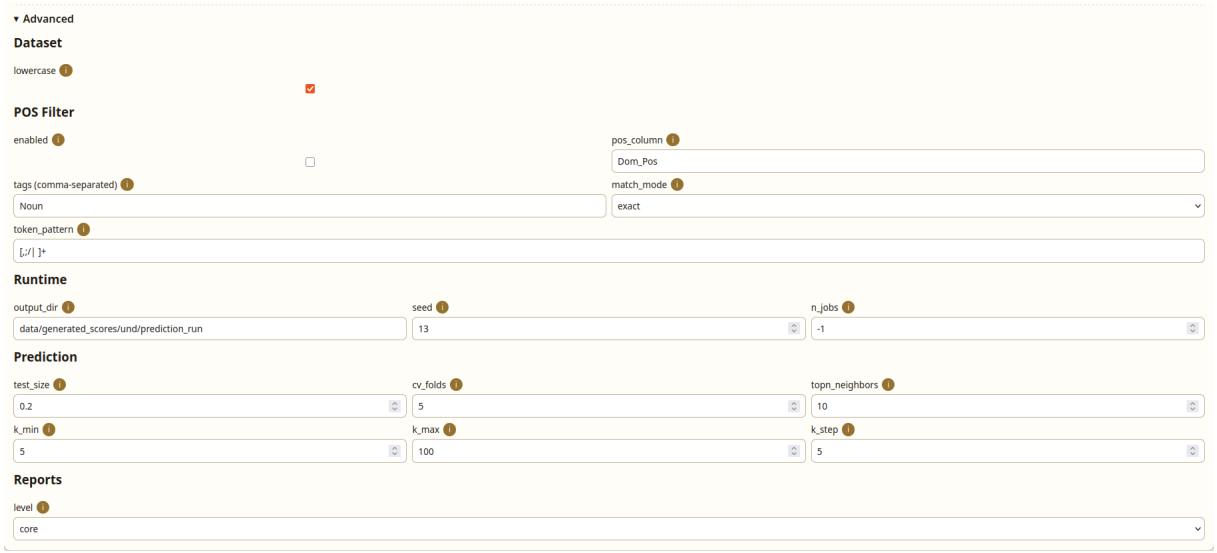

Figure 5: Advanced settings.

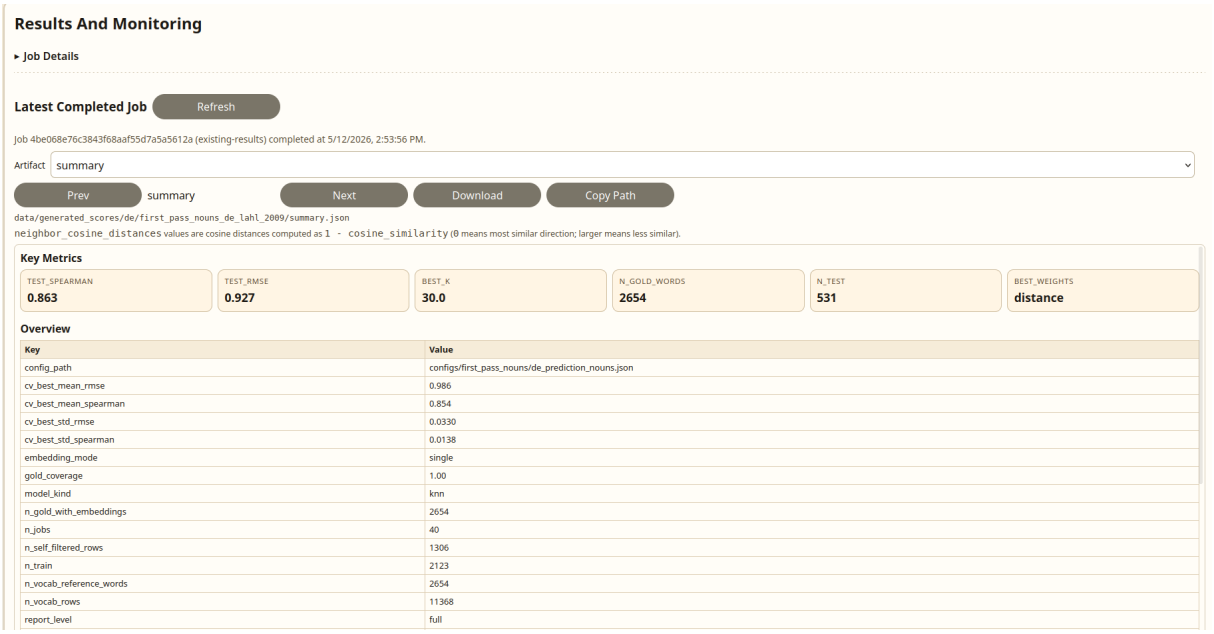

Figure 6: Summary output.

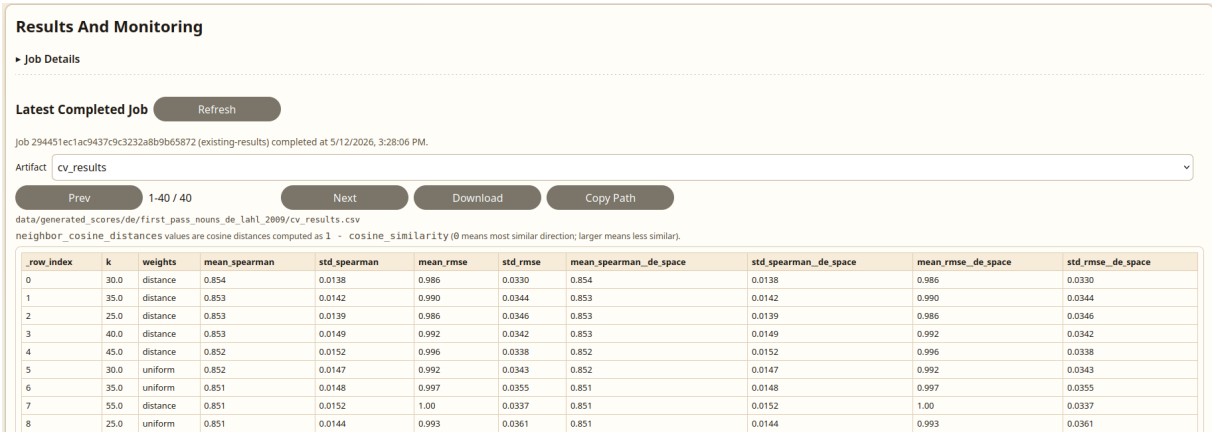

Figure 7: Cross-validation output.

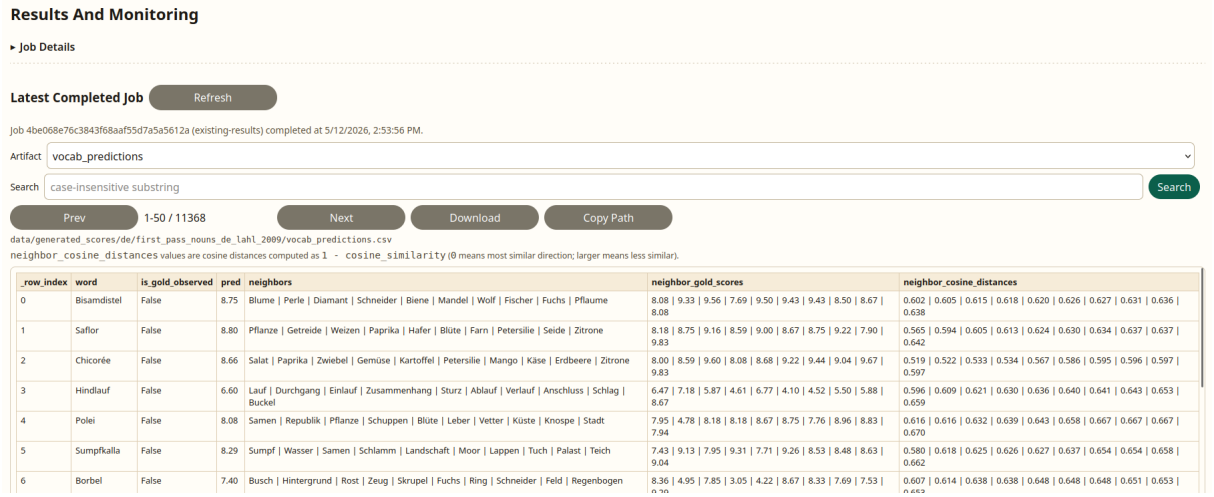

Figure 8: Vocabulary-prediction output.

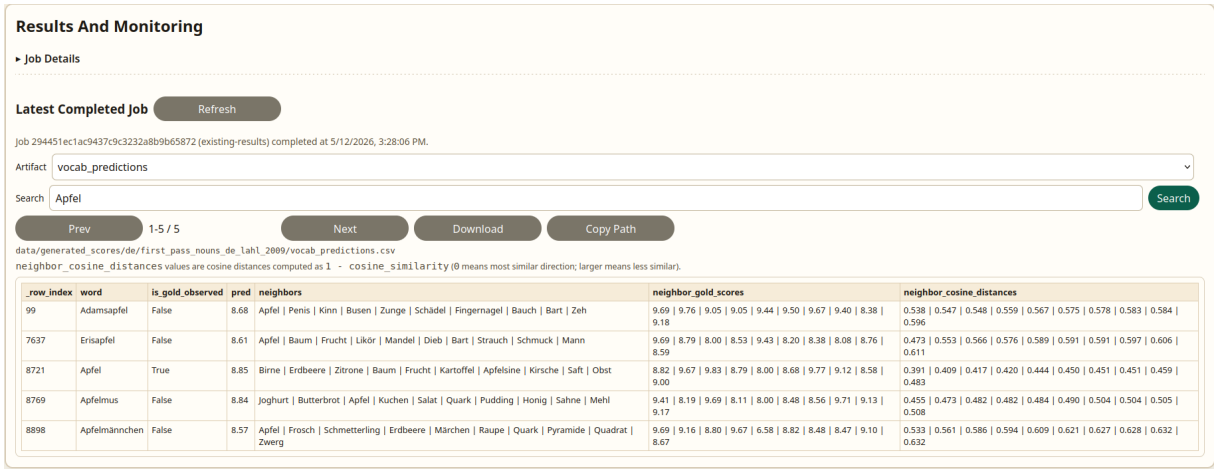

Figure 9: Search within the vocabulary-prediction output.

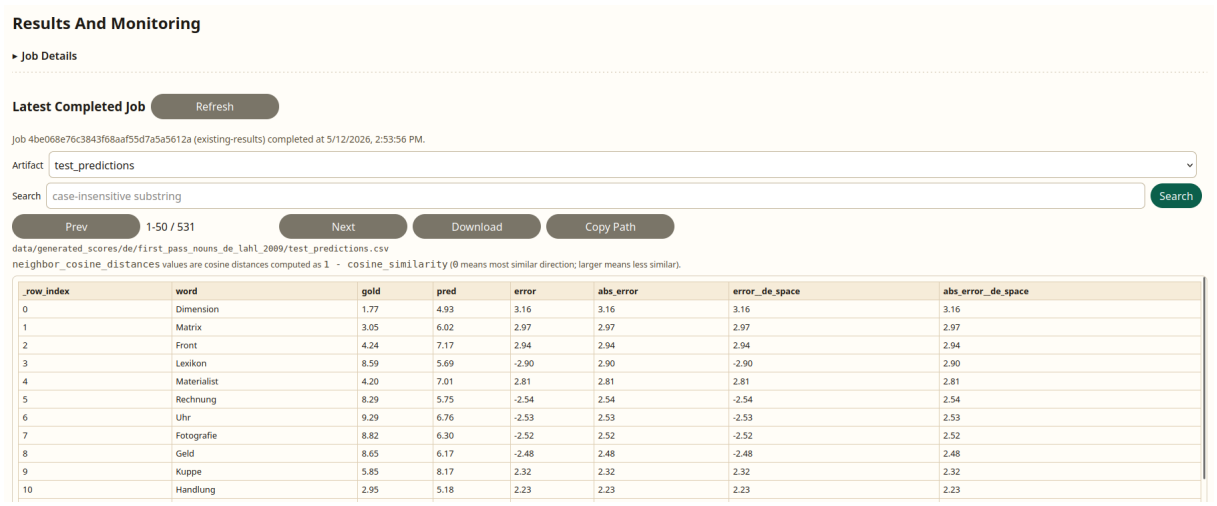

Figure 10: Held-out prediction output.

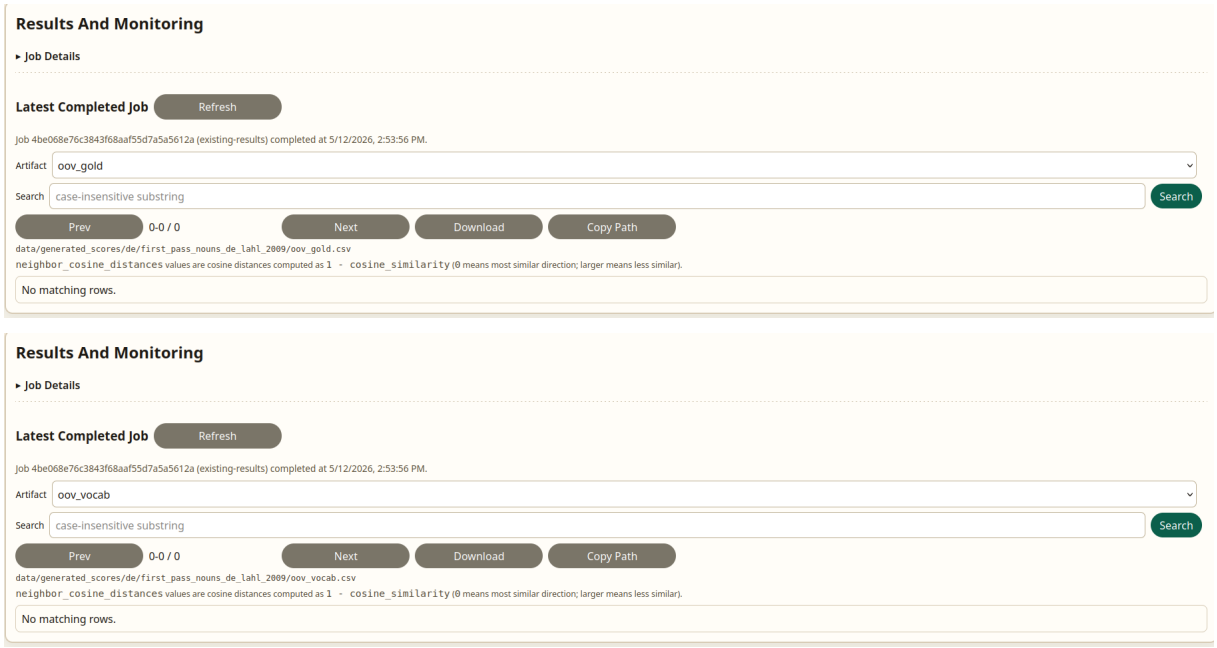

Figure 11: OOV outputs for the gold resource and target vocabulary. Empty tables indicate that the selected embedding space covered all gold and target items in this run.