A RAG Approach for Typological Database Completion

Jonathan Hus, Antonios Anastasopoulos
Department of Computer Science, George Mason University
jhus@gmu.edu, antonis@gmu.edu

Abstract

Linguistic reference material is a trove of information that can be utilized for the analysis of languages. The material, in the form of grammar books and sketches, has been used for machine translation, but it can also be used for language analysis. Retrieval Augmented Generation (RAG) has been demonstrated to improve large language model (LLM) capabilities by incorporating external reference material into the generation process. In this paper, we investigate the use of grammar books and RAG techniques to identify language features. We use Grambank for feature definition and ground truth values, and we evaluate on five typologically diverse low-resource languages. We demonstrate that this approach can effectively make use of reference material.[1]

1 Introduction

Grambank is an online database of language features and contains information on more than 2000 languages (Skirgård et al., 2023a). The database is structured around 195 language features. For each language in the database, the features are coded with a value, covering a wide spectrum of typological information such as subject-verb-object order, definite and indefinite article usage, and morphological markings (Skirgård et al., 2023b). This data is intended to be used for linguistic research; for example, Skirgård et al. (2023b) analyzed the data to make an argument for the significance of genealogical inheritance on language diversity.

The database, similar to others in the past like WALS (Dryer and Haspelmath, 2013), is hand-coded by a team of contributors. Questionnaires are provided to coders that contain the feature name, description, possible values, among other information. The intention is that coders can code the features using grammars or grammar books, even if they are not experts in that language.

While Grambank has feature information on more than 2000 languages, that leaves almost 5000 languages that are missing from the database. Questionnaires for a language are typically provided to a single contributor and completing the remainder of the database will require significant human capital. With the database missing data, researchers will not be able to utilize this resource to answer questions about certain languages or language families, nor will they be able to have the broadest possible coverage of languages when making general analyses (e.g., genealogical inheritance vs geographical diffusion in shaping language diversity).

In order to fill this gap, we propose a Retrieval-Augmented Generation (RAG) approach using a Large Language Model (LLM) that can query linguistic reference material. Since the questionnaire is intended to be completed by non-experts using grammar books and sketches, a sufficiently-capable LLM with access to the correct resources may be able to provide feature codes for languages not present in Grambank.

2 Related Work

RAG has become a popular approach applied to many applications of LLMs (Gao et al., 2024). The technique enables the incorporation of external knowledge, stored in a database, into the LLM input context.

Kornilov and Shavrina (2024) built a benchmark to evaluate RAG approaches, with Grambank and WALS feature prediction as targets. In contrast, our work more closely mimicks the process of Gramank coders' efforts, covering all Grambank features.

Hammarström et al. (2020) used term spotting to extract linguistic information from digitized raw-text descriptions, and Virk et al. (2021) describe

[1]Code and data to reproduce our experiments are provided here: https://github.com/jonathanhus/gram_features.

Proceedings of the 8th Workshop on Research in Computational Linguistic Typology and Multilingual NLP, pages 39–49

an automated deep learning approach to extract linguistic features from textual language descriptions.

The use of full-length grammar books has previously been explored when constructing prompts for LLMs. In Machine Translation from One Book, Tanzer et al. (2023) used a grammar book to perform machine translation between English and Kalamang, and more recently Hus and Anastasopoulos (2024) expanded this translation approach to 15 additional low-resource languages.

3 Methodology

Our RAG architecture consists of two main components: ChromaDB for the vector database and GPT-4o for the LLM.

The vector database is organized into collections, where a collection represents a single language. When loading the vector database, we chose the grammar books that are referenced and used by the Grambank coders when they coded language features. Some languages use a single grammar book while others use multiple. We obtained our grammar books from the DReaM corpus (Virk et al., 2020), which contains digitized versions of thousands of linguistic documents. These grammar books are vectorized and loaded into ChromaDB, with each page stored as a single entry.

Each feature in Grambank has a number of attributes that we use to format our LLM prompts. Each feature is written in the form of a question. Additionally, a summary is provided, which contains amplifying information, and last we provide instructions on how to label the feature. For each feature in Grambank, the attributes are used to create a query, which returns the top N similar entries from the vector database. The Grambank attributes are also used to formulate the prompt for the LLM. The database entries, which ideally contain the relevant grammatical information, are included in the prompt. An example prompt is illustrated in Appendix C. As indicated in the prompt, the LLM outputs the predicted feature code.

The full pipeline, therefore, operates as follows. Feature information is obtained from Grambank files and is used to query for relevant grammar book pages from ChromaDB. These retrieved pages, as well as task information from the Grambank files, are then formatted into a prompt for an LLM.

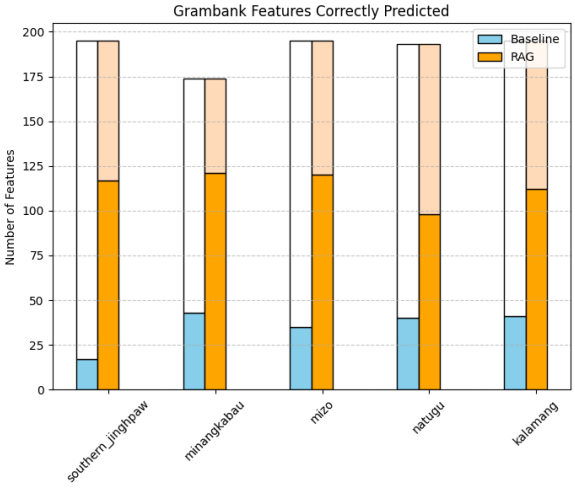

Figure 1: Baseline vs RAG Feature Prediction. RAG approach shows significant improvement over the baseline in identifying grammar features.

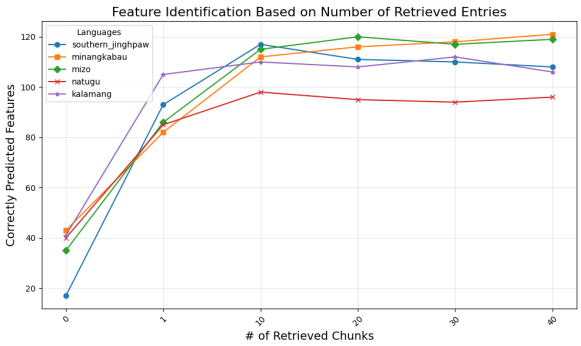

Figure 2: Number of correctly predicted features based on number of retrieved pages that are included in the prompt. Considerable improvements observed when including up to 10 pages in the prompt.

4 Results

4.1 Baseline vs RAG

The first basic question is whether RAG with grammar books improves the performance compared to simply asking the LLM for an answer. We created prompts that both did and did not contain the retrieved document pages from the database. Baseline values (Figure 1) show the performance with no retrieved documents included. RAG values show the number of features correctly predicted when including retrieved pages in the prompt. For all languages, RAG showed a marked improvement over the baseline, with increases of 30% to 52%, indicating the effectiveness of this approach.

4.2 Document Retrieval

Given the performance boost obtained from incorporating RAG, a natural extension would be to include more excerpts from grammar books to see

Language	Sources	Features w/ Pages	≥1 Match	Mean	Max	Skyline	Full Pipeline
Mizo (lus)						120/170	
– without summary	1	170	90	0.125	0.750		
– with summary	1	170	92	0.120	0.667		103/170
Southern Jinghpaw (kac)						112/195	
– without summary	3	195	134	0.143	0.571		
– with summary	3	195	132	0.144	0.500		117/195
Kalamang (kgv)						40/58	
– without summary	1	58	38	0.177	0.500		
– with summary	1	58	39	0.212	0.600		41/58
Minangkabau (min)						91/111	
– without summary	3	111	46	0.078	0.500		
– with summary	3	111	49	0.083	0.400		79/111
Natugu (ntu)						39/70	
– without summary	6	70	25	0.042	0.333		
– with summary	6	70	30	0.57	0.333		53/70

Table 1: Pipeline analysis characterizing the performance of the vector database search, the LLM prediction (skyline), and the full pipeline.

if that further increases performance. For each language, we performed experiments that included an increasing number of pages in the prompt (Figure 2), up to 40. Our results are consistent with other published results (Leng et al., 2024) that show a performance increase as more documents are added. However, more documents give diminishing and eventually negative returns.

4.3 Pipeline Analysis

The system components, namely the database and the LLM, are largely independent. Their performance can be separately assessed to analyze their impact on the system. The experiments and results in this section only look at the features for a given language that have page numbers identified.

Vector Database When coders are specifying feature values for a language in Grambank, they are encouraged to include both the reference book and page number(s) used to determine that value. We can use this information to evaluate the performance of our vector database retrieval. For the features with pages numbers identified, we evaluate the similarity with the pages retrieved via vector search. For each language, Table 1 shows the number of reference sources that the Grambank coders used to determine the feature values in the "Sources" column. Since coders do not always provide page numbers with a feature, we evaluate similarity only on the features that have page numbers. The feature, which is expressed in the form of a question, is always included in the search term. We evaluated both including and omitting the summary in the search, shown in the "with summary" and "without summary" rows, respectively.

For each feature, we retrieve four documents from the database. We calculate the Jaccard similarity between this set of documents and the ground truth provided by Grambank. Within a language, we calculate the mean and maximum of all the similarity scores, shown in the "Mean" and "Max" columns of Table 1, respectively. Not surprisingly, languages with more source documents have lower average similarity scores. While this could be due to a larger search space, it is also possible that the correct answer can be found in a document other than what the coder chose. As a coarser indication of similarity, we evaluate the number of features where at least one of the retrieved pages matches the ground truth, shown in the "≥1 Match" column.

Overall, the RAG approach shows a good deal of agreement between the database search and human coding, with more than 50% of features across all languages having at least one matching page. In general, including the summary in the prompt improves the similarity scores, although the mean similarity score is lower in Mizo when using the summary and the number of features with at least one match is lower in Southern Jinghpaw. Natugu has the worst similarity scores among all the languages, which may be attributed to it also having the most source documents.

Main domain	kac	min	lus	ntu	kgv
Clause	52.0 (26/50)	51.06 (24/47)	62.0 (31/50)	48.0 (24/50)	60.0 (30/50)
Nominal domain	64.79 (46/71)	81.25 (52/64)	61.97 (44/71)	50.0 (35/70)	67.61 (48/71)
Numeral	50.0 (2/4)	75.0 (3/4)	100.0 (4/4)	25.0 (1/4)	25.0 (1/4)
Pronoun	83.33 (10/12)	72.73 (8/11)	33.33 (4/12)	50.0 (6/12)	58.33 (7/12)
Verbal domain	56.9 (33/58)	70.83 (34/48)	63.79 (37/58)	56.14 (32/57)	44.83 (26/58)
Total	60.0 (117/195)	69.54 (121/174)	61.54 (120/195)	50.78 (98/193)	57.44 (112/195)

Table 2: Match accuracy by main domain for each language. Values are formatted as percentage (correct/total).

	LLM output code						
label	**0**	**1**	**2**	**3**	**?**	**IDK**	**Total**
0	312	94	2	0	76	11	495
1	29	144	2	1	24	1	201
2	0	2	9	3	2	1	17
?	84	27	2	0	103	23	239
Total	425	267	15	4	205	36	952

Table 3: Aggregate distribution of code values by category for all languages. We highlight correct predictions.

Full Pipeline vs Skyline To evaluate how well the LLM uses the provided material to determine the feature value, we bypassed the database search and instead directly included the referenced pages in the prompt. As with the vector database characterization, we are only able to utilize this approach for features that have associated reference page numbers in the Grambank dataset. The "Skyline" column in Table 1 shows the number of features correctly predicted. We also evaluate the full pipeline (database and LLM) on the set of features that have reference pages, similar to above. Surprisingly, comparing to the skyline approach, the full pipeline sometimes but not always outperforms skyline. Natugu shows the largest increase from the skyline to the full pipeline. This may be attributed to the increased number of grammar sources, which could increase the amount of highly-relevant excerpts included in the prompt.

4.4 Feature Analysis

Confusion matrices are provided in the appendix for each language, with an aggregate shown in Table 3. When it comes to errors, the LLM is more likely to predict a feature is present when in fact it is absent than to predict a feature is absent when in fact it is present. Such false positives are the most common errors observed in the data. The next highest error categories are a true code "0" (absent feature) with a predicted "?" (unknown) and a true code "?" with a predicted code "0". Both of these situations involve using information from the grammar books to prove that something does not exist. Unless the grammar books explicitly state that some aspect of a grammar is not present in a particular language, the LLMs will struggle to identify excerpts that support this prediction.

The prompt includes an instruction for the LLM to output "IDK", refraining from a prediction if it is unable to determine a code value. In some senses, this is duplicative with the "?" value, but it does provide the LLM with an option instead of a forced guess. That being said, the LLM only returned "IDK" 36 times (3.8%). There were four instances (0.4%) in which the LLM did not conform to the answer options specified in the task instructions and instead chose a "3", which is not a valid choice for any feature.

Grambank categorizes each feature into one of five domains. The domains are broad linguistic topics that are designed to enable further analysis. For the best performing RAG system for each language, the performance for each domain is shown in Table 2. Across all languages, our RAG system performs the best on the "Nominal domain" category, with an accuracy of 65%, while it struggles the most with the "Clause" domain, with an accuracy of 55%. This suggests that reference material about nominal domain features (e.g., case marking, articles, possession) is able to be extracted easier from the grammar books than text about clause domain features (e.g., sentence structure, subordination, coordination).

5 Conclusion

In this paper, we showed the benefit of incorporating grammar books into RAG pipelines. We evaluated the performance on several languages and showed that this approach is capable of answering questions about typological structures of languages. Our work spotlights one application where this approach yields benefits and shows that linguistic reference material is a valuable resource for research on low resource languages.

6 Limitations

Grammar books are vectorized and stored in a database for RAG. In order to take advantage of this approach, we use grammar books in PDF format that contain text. However, high-quality grammar books are difficult to obtain for many languages. The DReaM corpus does an admirable job of curating and digitizing many linguistic references, but not all languages have reference material in the necessary format. Additionally, tables lose information that is conveyed by the location of text relative to other text on the page. The LLMs, therefore, are most likely not taking full advantage of that information.

We used an OpenAI model (gpt-4o-mini). While this model is quite performant, there are some drawbacks. OpenAI models are truly closed models, with only an API available. The architecture, weights, and training scheme are not available to researchers.

Acknowledgements

This work was partially supported by the National Science Foundation under CAREER award 2439202. This work was partially supported by resources provided by the Office of Research Computing at George Mason University (URL: `https://orc.gmu.edu`) and funded in part by grants from the National Science Foundation (Award Number 2018631).

References

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Harald Hammarström, One-Soon Her, and Marc Allassonnière-Tang. 2020. Term spotting: A quick-and-dirty method for extracting typological features of language from grammatical descriptions. In *Selected contributions from the Eighth Swedish Language Technology Conference (SLTC-2020)*, pages 27–34, Göteborg, Sweden.

Jonathan Hus and Antonios Anastasopoulos. 2024. Back to school: Translation using grammar books. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20207–20219, Miami, Florida, USA. Association for Computational Linguistics.

Albert Kornilov and Tatiana Shavrina. 2024. From mteb to mtob: Retrieval-augmented classification for descriptive grammars. *Preprint*, arXiv:2411.15577.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. 2024. Long context rag performance of large language models. *Preprint*, arXiv:2411.03538.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Martine Robbeets, Noor Karolin Abbas, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Robert D. Borges, Swintha Danielsen, Luise Dorenbusch, Ella Dorn, John Elliott, Giada Falcone, Jana Fischer, Yustinus Ghanggo Ate, Hannah Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu Huntington-Rainey, Jessica K. Ivani, Marilen Johns, Erika Just, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Nora L. M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Tânia R. A. Martins, Celia Mata German, Suzanne van der Meer, Jaime Montoya Samamé, Michael Müller, Saliha Muradoglu, Kelsey Neely, Johanna Nickel, Miina Norvik, Cheryl Akinyi Oluoch, Jesse Peacock, India O. C. Pearey, Naomi Peck, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W. P. Schmidt, Amalia Skilton, Wikaliler Daniel Smith, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023a. Grambank reveals the importance of genealogical constraints on linguistic diversity and highlights the impact of language loss. *Science Advances*, 9(16):eadg6175.

Hedvig Skirgård, Hannah J. Haynie, Harald Hammarström, Damián E. Blasi, Jeremy Collins, Jay Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Noor Karolin Abbas, Sunny Ananth, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Anina Bolls, Robert D. Borges, Mitchell Browen, Lennart Chevallier, Swintha Danielsen, Sinoël Dohlen, Luise Dorenbusch, Ella Dorn, Marie Duhamel, Farah El Haj Ali, John Elliott, Giada Falcone, Anna-Maria Fehn, Jana Fischer, Yustinus Ghanggo Ate, Hannah

Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu H. Huntington-Rainey, Guglielmo Inglese, Jessica K. Ivani, Marilen Johns, Erika Just, Ivan Kapitonov, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Kate Lynn Lindsey, Nora L. M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Alexandra Marley, Tânia R. A. Martins, Celia Mata German, Suzanne van der Meer, Jaime Montoya, Michael Müller, Saliha Muradoğlu, HunterGatherer, David Nash, Kelsey Neely, Johanna Nickel, Miina Norvik, Bruno Olsson, Cheryl Akinyi Oluoch, David Osgarby, Jesse Peacock, India O.C. Pearey, Naomi Peck, Jana Peter, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W. P. Schmidt, Dineke Schokkin, Jeff Siegel, Amalia Skilton, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Daniel Wikalier Smith, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023b. Grambank v1.0. Dataset.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2023. A benchmark for learning to translate a new language from one grammar book. In *Arxiv*.

Shafqat Mumtaz Virk, Daniel Foster, Azam Sheikh Muhammad, and Raheela Saleem. 2021. A deep learning system for automatic extraction of typological linguistic information from descriptive grammars. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1480–1489, Held Online. INCOMA Ltd.

Shafqat Mumtaz Virk, Harald Hammarström, Markus Forsberg, and Søren Wichmann. 2020. The DReaM corpus: A multilingual annotated corpus of grammars for the world's languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 878–884, Marseille, France. European Language Resources Association.

A Additional Results

Tabular results for Figure 2 are provided in Table 4.

Confusion matrices for each of the languages are combined into an aggregate table in section 4. The results for each language is provided here.

Results mapped to Grambanks' "finer grouping" categories are provided in Table 10.

B Resources

For our experiments, we gathered grammar books, which were tokenized and stored in the vector database. We used the OpenAI text-embedding-ada-002 model to generate text embeddings from the grammar book text. Grammar books were obtained from DReaM (Virk et al., 2020) and converted to the format required by the code. We used the grammar books that were referenced by Grambank coders for each language. We note the filename that was downloaded from DReaM. There were a handful of instances in which the document was not loaded into the vector database, as is noted in Table 11.

	baseline	rag_1	rag_10	rag_20	rag_30	rag_40	total
southern_jinghpaw	17	93	117	111	110	108	195
minangkabau	43	82	112	116	118	121	174
mizo	35	86	115	120	117	119	195
natugu	40	85	98	95	94	96	193
kalamang	41	105	110	108	112	106	195

Table 4: Number of correctly predicted features based on number of retrieved pages included in the prompt.

	LLM output code						
code	**0**	**1**	**2**	**3**	**?**	**IDK**	**Total**
0	70	25	0	0	19	1	115
1	9	39	0	0	4	1	53
2	0	0	1	2	1	0	4
?	12	2	0	0	7	2	23
Total	91	66	1	2	31	4	195

Table 5: Southern Jinghpaw - Distribution of code values by category.

	LLM output code						
code	**0**	**1**	**2**	**3**	**?**	**IDK**	**Total**
0	85	18	0	0	16	4	123
1	2	30	0	1	4	0	37
2	0	1	2	0	1	0	4
?	4	2	0	0	4	0	10
Total	91	51	2	1	25	4	174

Table 6: Minangkabau - Distribution of code values by category.

	LLM output code					
code	**0**	**1**	**2**	**?**	**IDK**	**Total**
0	78	32	0	16	4	130
1	10	32	2	5	0	49
2	0	0	3	0	0	3
?	1	5	0	7	0	13
Total	89	69	5	28	4	195

Table 7: Mizo - Distribution of code values by category.

	LLM output code					
code	**0**	**1**	**2**	**?**	**IDK**	**Total**
0	57	5	2	10	1	75
1	3	12	0	6	0	21
2	0	1	2	0	0	3
?	39	9	1	41	6	96
Total	99	27	5	57	7	195

Table 8: Kalamang - Distribution of code values by category.

	LLM output code						
code	**0**	**1**	**2**	**3**	**?**	**IDK**	**Total**
0	22	14	0	0	15	1	52
1	5	31	0	0	5	0	41
2	0	0	1	1	0	1	3
?	28	9	1	0	44	15	97
Total	55	54	2	1	64	17	193

Table 9: Natugu - Distribution of code values by category.

Feature Type	**kac**	**min**	**lus**	**ntu**	**kgv**
TAME	42.11 (8/19)	83.33 (10/12)	73.68 (14/19)	36.84 (7/19)	21.05 (4/19)
VP (other)	43.75 (7/16)	78.57 (11/14)	37.5 (6/16)	75.0 (12/16)	31.25 (5/16)
Argument marking (core)	70.0 (14/20)	68.75 (11/16)	65.0 (13/20)	52.63 (10/19)	75.0 (15/20)
Argument marking (non-core)	66.67 (8/12)	41.67 (5/12)	41.67 (5/12)	63.64 (7/11)	33.33 (4/12)
Class	82.35 (14/17)	88.24 (15/17)	70.59 (12/17)	58.82 (10/17)	70.59 (12/17)
Clause order	69.57 (16/23)	57.14 (12/21)	56.52 (13/23)	52.17 (12/23)	69.57 (16/23)
Deixis	66.67 (6/9)	66.67 (6/9)	55.56 (5/9)	11.11 (1/9)	44.44 (4/9)
Non-verbal predication	35.0 (7/20)	70.0 (14/20)	65.0 (13/20)	55.0 (11/20)	45.0 (9/20)
Number	77.27 (17/22)	85.71 (18/21)	68.18 (15/22)	31.82 (7/22)	77.27 (17/22)
Order in NP	50.0 (5/10)	66.67 (4/6)	60.0 (6/10)	60.0 (6/10)	70.0 (7/10)
Quantification	50.0 (4/8)	62.5 (5/8)	87.5 (7/8)	62.5 (5/8)	62.5 (5/8)
Valency	64.29 (9/14)	69.23 (9/13)	57.14 (8/14)	64.29 (9/14)	78.57 (11/14)
Verb complex	40.0 (2/5)	20.0 (1/5)	60.0 (3/5)	20.0 (1/5)	60.0 (3/5)
Total	60.0 (117/195)	69.54 (121/174)	61.54 (120/195)	50.78 (98/193)	57.44 (112/195)

Table 10: Match accuracy by finer grouping for each language. Values are formatted as percentage (correct/total).

Language	Grambank Sources	Chroma (from DReaM)
Mizo	Chhangte, Lalnunthangi. 1986. *A Preliminary Grammar of the Mizo Language*.	chhangte_mizo-grammar1986_o.pdf
Minangkabau	Crouch, Sophie. 2009. *Voice and verb morphology in Minangkabau, a language of West Sumatra, Indonesia*.	crouch_minangkabau2009.pdf
	Reibaud, Rusmidar. 2004. *Parlons Minangkabau. Paris: L'Harmattan.*.	reibaud_minangkabau2004_o.pdf
	Roesli, Oleh. 1967. *Bahasa Minangkabau. Jakarta: Bhratara*.	roesli_minangkabau1967v2_o.pdf
Southern Jinghpaw (Kachin)	Kurabe, Keita. 2012. *Jingpho Dialogue Texts with Grammatical Notes*.	kurabe_jingpho-dialogue2012.pdf
	Kurabe, Keita. 2017. *Jinghpaw*.	kurabe_jinghpaw2017_o.pdf
	Qingxia, Dai and Diehl, Lon. 2003. *Jinghpo*.	Qingxia-diehl_jingpho2003_o.pdf
Kalamang	Visser, Eline. 2016. *A grammar sketch of Kalamang, with a focus on phonetics and phonology*.	visser_kalamang2016_o.pdf
	Galis, Klaas Wilhelm. 1955. *Talen en dialecten van Nederlands Nieuw-Guinea*.	Not loaded. No feature identifies a page number from this source.
Natugu	Næss, Åshild and Boerger, Brenda H. 2008. Reefs-Santa Cruz as Oceanic: Evidence from the Verb Complex. Oceanic Linguistics 47.	naess-boerger_rsc-verb2008.pdf
	Ross, Malcolm D. 2007. Two Kinds of Locative Construction in Oceanic Languages: A Robust Distinction. In Siegel, Jeff and Lynch, John and Eades, Diana (eds.), Language Description, History and Development: Linguistic Indulgence in Memory of Terry Crowley,	Not loaded. Couldn't find free version
	Wurm, Stephen A. 1969. The Linguistics Situation in the Reef and Santa Cruz Islands. In Papers in Linguistics of Melanesia No. 2, 47-105. Canberra: Research School of Pacific and Asian Studies, Australian National University	wurm_reef-santa-cruz1969.pdf
	Wurm, Stephen A. 1971. The Papuan linguistic situation. In Sebeok, Thomas A. (ed.), Linguistics in Oceanea, 541-657. Berlin: Mouton de Gruyter.	sebeok_oceania1971_o.pdf
	Wurm, Stephen A. 1972. Notes on Indication of Possession with Nouns in Reef and Santa Cruz Islands Languages. In Papers in Linguistics of Melanesia 3, 85-113. Canberra: Research School of Pacific and Asian Studies, Australian National University.	wurm_reef-santa-cruz1972v2_o.pdf
	Wurm, Stephen A. 1976. The Reef Islands-Santa Cruz Family. In Wurm, Stephen A. (ed.), New Guinea Area Languages and Language Study Vol 2: Austronesian Languages, 637-674. Canberra: Research School of Pacific and Asian Studies, Australian National University.	wurm_reef-santa-cruz1976_o.pdf
	Wurm, Stephen. (1978) Reef-Santa Cruz: Austronesian, but . . . !. In Stephen Wurm and Lois Carrington (eds.), Proceedings of the 2nd International Conference on Austronesian Linguistics: Fascicle 2 (Pacific Linguistics: Series C 61), 969-1010. Canberra: Research School of Pacific and Asian Studies, Australian National University.	Not loaded. PDF scan had 2 text pages per scanned page.
	van den Berg, Rene and Boerger, Brenda H. 2011. A Proto-Oceanic Passive? Evidence from Bola and Natügu. Oceanic Linguistics 50. 221-246	vandenberg_bola-natugu2011.pdf

Table 11: Grammar books for each language.

C Prompt Format

Each feature to be predicted is formatted into a prompt for GPT-4. In the following sections, we show the format of the prompt by example.

Prompt Template

You are an expert linguist with extensive knowledge about many languages. Answer the following question about the language {language_name}. You are also provided with additional information about the question and you are given a procedure that indicates allowable answers for the question. You MUST provide an answer following the procedure. If you do not know the answer, answer 'IDK'. Output the answer in JSON format with the following key-value pairs: 'code': code, 'comment': other_data. Please format the response as valid JSON that I can parse.

{question}
Here is a summary about the question:
{summary}
{context}
Here is the procedure to follow and the allowable responses:
{procedure}

Example Using GB020-lush1249

Are there definite or specific articles?
Here is a summary about the question:
An article is a marker that accompanies the noun and expresses notions such as (non-)specificity and (in)definiteness. Sometimes these notions of specificity and definiteness are summed up in the term 'identifiability'. The formal expression is irrelevant; articles can be free, bound, or marked by suprasegmental markers such as tone.
Articles are different from demonstratives in that demonstratives occur in a paradigm of markers that have a clear spatial deictic function. As demonstratives can grammaticalize into definite or specific articles, they form a natural continuum, making it hard to define discrete categories, but to qualify as an article a marker should be used in some cases to express definiteness without also expressing a spatial deictic meaning.
To help answer the question, here is relevant data retrieved from a grammar book
GRAMMAR BOOK EXCERPTS RETRIEVED FROM DATABASE
Here is the procedure to follow and the allowable responses:
1. Code 1 if there is a morpheme that can mark definiteness or specificity without also conveying a spatial deictic meaning.
2. Code 0 if the source does not mention a definite article and you cannot find one in examples or texts in an otherwise comprehensive grammar.
3. Code ? if the grammar does not contain enough analysis to determine whether there is a definite article or not.
4. If you have coded 1 for GB020 and 0 for GB021 and GB022, please write a comment explaining the position of the definite or specific article.