OPEN ACCESS
J DATA SCI APPL, VOL. 2, NO. 2, PP.049-059, JULY 2019

E-ISSN 2614-7408
DOI: 10.34818/JDSA.2019.2.23

JOURNAL OF DATA SCIENCE AND ITS APPLICATIONS

Visualizing Language Lexical Similarity Clusters: A Case Study of Indonesian Ethnic Languages

Arbi Haza Nasution [#1], Yohei Murakami [*2]

 # Department of Informatics Engineering, Universitas Islam Riau
Riau, Indonesia

[1] arbi@eng.uir.ac.id

[*] Faculty of Information Science and Engineering, Ritsumeikan University
Kyoto, Japan

[2] yohei@fc.ritsumei.ac.jp

Received on 01-06-2019, revised on 22-07-2019, accepted on 27-08-2019

Abstract

Language similarity clusters are useful for computational linguistic researches that rely on language similarity or cognate recognition. The existing language similarity clustering approach which utilizes hierarchical clustering and k-means clustering has difficulty in creating clusters with a middle range of language similarity. Moreover, it lacks an interactive visualization that user can explore. To address these issues, we formalize a graph-based approach of creating and visualizing language lexical similarity clusters by utilizing ASJP database to generate the language similarity matrix, then formalize the data as an undirected graph. To create the clusters, we apply a connected components algorithm with a threshold of language similarity range. Our interactive online tool allows a user to dynamically create new clusters by changing the threshold of language similarity range and explore the data based on language similarity range and number of speakers. We provide an implementation example of our approach to 119 Indonesian ethnic languages. The experiment result shows that for the case of low system execution burden, the system performance was quite stable. For the case of high system execution burden, despite the fluctuated performance, the response times were still below 25 seconds, which is considered acceptable.

Keywords: graph visualization, lexical similarity, language family, Indonesian ethnic languages

I. INTRODUCTION

Nowadays, machine-readable bilingual dictionaries are being utilized in actual services [1] to support intercultural collaboration [2]–[5], but low-resource languages lack such resources. Indonesia has a population of 221,398,286 and 707 living languages which cover 57.8% of Austronesian Family and 30.7% of languages in Asia [6]. There are 341 Indonesian ethnic languages facing various degree of language endangerment (trouble / dying) where some of the native speaker do not speak Bahasa Indonesia well since they are in remote areas. Unfortunately, there are 13 Indonesian ethnic languages which are already extinct. In order to save low-resource languages like Indonesian ethnic languages from language endangerment, prior works tried to enrich the basic language resource, i.e., bilingual dictionary [7]–[10]. Those previous researchers require language similarity matrix and clusters of the low-resource languages to select the target languages. The existing language similarity clustering approach [11] which utilizes hierarchical clustering and k-means clustering has a difficulty in creating clusters within a middle range of language similarity from n to m where $n > 0\%$ and $m<100\%$. Moreover, the existing approach lack an interactive visualization that user can explore. To address these issues, we formulate a graph-based approach of creating language similarity clusters within any range of language similarity and further enabling visualization and exploration of the languages within or across clusters.

The rest of this paper is organized as follows: In section 2, we will briefly discuss a software where we get our dataset from. We will explain our proposed graph-based clustering approach in section 3. Section 4 details the implementation of the approach, where we developed a language similarity clusters visualization with a dataset of 119 Indonesian ethnic languages as a case study. We evaluate the performance stability of the language similarity clusters visualization in section 5. Section 6 concludes this paper.

II. AUTOMATED SIMILARITY JUDGMENT PROGRAM

Historical linguistics is the scientific study of language change over time in term of sound, analogical, lexical, morphological, syntactic, and semantic information [12]. Comparative linguistics is a branch of historical linguistics that is concerned with language comparison to determine historical relatedness and to construct language families [13]. Many methods, techniques, and procedures have been utilized in investigating the potential distant genetic relationship of languages, including lexical comparison, sound correspondences, grammatical evidence, borrowing, semantic constraints, chance similarities, sound-meaning isomorphism, etc [14]. The genetic relationship of languages is used to classify languages into language families. Closely-related languages are those that came from the same origin or proto-language, and belong to the same language family.

Swadesh List is a classic compilation of basic concepts for the purposes of historical-comparative linguistics. It is used in lexicostatistics (quantitative comparison of lexical cognates) and glottochronology (chronological relationship between languages). There are various version of swadesh list with a number of words equal 225 [15], 215 & 200 [16], and lastly 100 [17]. To find the best size of the list, Swadesh [18] states that “The only solution appears to be a drastic weeding out of the list, in the realization that quality is at least as important as quantity. Even the new list has defects, but they are relatively mild and few in number."

A widely-used notion of string/lexical similarity is the edit distance or also known as Levenshtein distance (LD): the minimum number of insertions, deletions, and substitutions required to transform one string into the other [19]. For example, LD between “kitten" and “sitting" is 3 since there are three transformations needed: kitten → sitten (substitution of “s" for “k"), sitten → sittin (substitution of “i" for “e"), and finally sittin → sitting (insertion of “g" at the end).

There are a lot of previous works using Levenshtein distances such as dialect groupings of Irish Gaelic [20] where they gather the data from questionnaire given to native speakers of Irish Gaelic in 86 sites. They obtain 312 different Gaelic words or phrases. Another work is about dialect pronunciation differences of 360 Dutch dialects [21] which obtain 125 words from Reeks Nederlandse Dialectatlassen. They normalize LD by dividing it by the length of the longer alignment. Tang and Heuven [22] measure linguistic similarity and intelligibility of 15 Chinese dialects and obtain 764 common syllabic units. Petroni and Serva [23] define lexical distance between two words as the LD normalized by the number of characters of the longer of the two. Wichmann et al. [24] extend Petroni definition as LDND and use it in Automated Similarity Judgment Program (ASJP).

The ASJP, an open source software was proposed by Holman et al. [25] with the main goal of developing a database of Swadesh lists [17] for all of the world's languages from which lexical similarity or lexical distance matrix between languages can be obtained by comparing the word lists. The lexical similarity or lexical distance is useful, for instance, for classifying a language group and for inferring its age of divergence. The classification is based on 100-item reference list of Swadesh [17] and further reduced to 40 most stable items [26]. The item stability is a degree to which words for an item are retained over time and not replaced by another lexical item from the language itself or a borrowed element. Words resistant to replacement are more stable. Stable items have a greater tendency to yield cognates (words that have a common etymological origin) within groups of closely related languages.

III. GRAPH-BASED CLUSTERING APPROACH

There are three varieties of language similarity range: a lower range from 0% to m where $m < 100\%$, an upper range from n to 100% where $n > 0\%$, and a middle range from n to m where $n > 0\%$ and $m < 100\%$. The existing language similarity clustering approach [11] utilizes hierarchical clustering to create clusters with an upper range of language similarity. From the generated dendrogram, we manually cut the dendrogram at n which will gives several clusters. To create clusters with a lower range of language similarity, we firstly utilize hierarchical clustering to create clusters with an upper range of language similarity, then labels the generated clusters when applying k-means clustering to obtain clusters with a lower range of language similarity. However, the existing approach has a difficulty in creating clusters with a middle range of language similarity. A better clustering algorithm that can create clusters with any range of language similarity is needed.

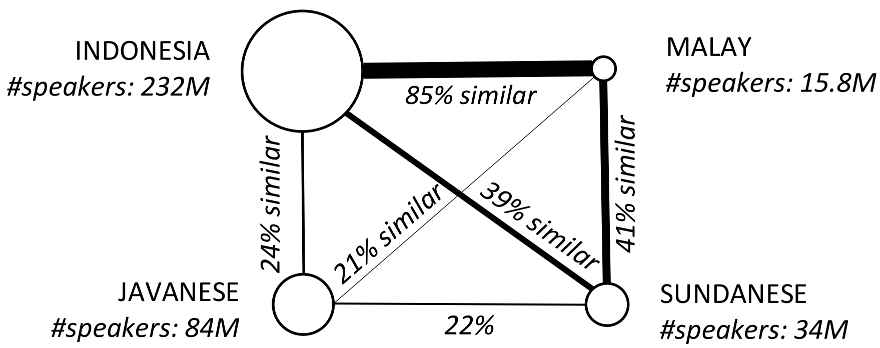

Fig. 1. Example of Language Similarity Graph

We formalize a graph-based approach of creating and visualizing language similarity clusters by utilizing ASJP database[1] to generate the language similarity/distance matrix, then formalize the data as an undirected graph using Neo4j[2]. A node represents a language and an edge represents a language similarity between the two languages. The size or diameter of the node represents the number of speakers the language has. The thickness of an edge represents how similar the two languages are. An example of language similarity graph is presented in Fig. 1.

To create the clusters, we apply a connected components algorithm with a threshold of language similarity range. The algorithm was first described by Galler and Fischer [27] and has been implemented by recent works [28], [29]. The connected components algorithm finds sets of connected nodes in an undirected graph where each node is reachable from any other node in the same set. It is often used early in an analysis to understand a graph's structure. The components in a graph are computed using either the breadth-first search or depth-first search algorithms.

IV. CASE STUDY

In this paper, we provide a dataset of 119 Indonesian ethnic languages as shown in Table I. We obtained 40-item word list of each Indonesian ethnic language with the number of speakers above 100,000 from ASJP database. We further generated the similarity/distance matrix of those languages and formalized it into an

[1] http://asjp.clld.org
[2] https://neo4j.com

undirected graph. Using hierarchical clustering as a baseline method, we created 11 clusters with a language similarity range threshold from 50% to 100% as shown in Fig. 2.

TABLE I
LIST OF 119 INDONESIAN ETHNIC LANGUAGES RANKED BY NUMBER OF SPEAKER

Code	Speaker	#Language	Code	Speaker	#Language
L1	232004800	Indonesian	L61	350000	Sindue Tawaili
L2	84308740	Malang	L62	350000	Tara
L3	84308740	Yogyakarta	L63	340000	Lom
L4	84308740	Old Or Middle Javanese	L64	331700	Salako Badamea
L5	34000000	Sundanese	L65	331000	Tolaki
L6	15848500	Malay	L66	331000	Tolaki Asera
L7	15848500	Palembang Malay	L67	331000	Tolaki Konawe
L8	6770900	Madurese	L68	331000	Tolaki Laiwui
L9	5530000	Minangkabau	L69	331000	Tolaki Mekongga
L10	5000000	Buginese	L70	331000	Tolaki Wiwirano
L11	5000000	Soppeng Buginese	L71	300000	Gayo
L12	5000000	Betawi	L72	300000	Kadatua
L13	3502300	Banjarese Malay	L73	300000	Muna
L14	3500032	Aceh	L74	300000	Sumbawa
L15	3330000	Bali	L75	285000	Kerinci
L16	2130000	Makasar	L76	255000	Sangir
L17	2100000	Sasak	L77	250000	Tae
L18	2000000	Toba Batak	L78	245020	Ambonese Malay
L19	1100000	Batak Mandailing	L79	240000	Kambera
L20	1000000	Gorontalo	L80	240000	Lewa Kambera
L21	1000000	Jambi Malay	L81	240000	Southern Kambera
L22	900000	Manggarai	L82	240000	Umbu Ratu Nggai Kambera
L23	890000	Kapuas Kahayan	L83	230000	Mongondow
L24	890000	Katingan	L84	180000	Abung Sukadana Lampung Nyo
L25	890000	Ngaju Baamang	L85	180000	Lamaholot Ile Mandiri
L26	890000	Ngaju Oloh Mangtangai	L86	180000	Lampung Nyo Abung Kotabumi
L27	890000	Ngaju Pulopetak	L87	180000	Lampung Nyo Melinting
L28	827000	Belalau Lampung Api	L88	180000	Menggala Tulang Bawang Lampung

L29	827000	Daya Lampung Api	L89	175000	Sika
L30	827000	Jabung Lampung Api	L90	150000	Anaiwoi Bajau
L31	827000	Kalianda Lampung Api	L91	150000	Bajoe Bajau
L32	827000	Kota Agung Lampung Api	L92	150000	Boepinang Bajau
L33	827000	Krui Lampung Api	L93	150000	Indonesian Bajau
L34	827000	Lampung	L94	150000	Kaleroang Bajau
L35	827000	Pubian Lampung Api	L95	150000	Kayuadi Bajau
L36	827000	Ranau Lampung Api	L96	150000	Kolo Bawah Bajau
L37	827000	Sukau Lampung Api	L97	150000	Lakaramba Bajau
L38	827000	Sungkai Lampung Api	L98	150000	Lakonea Bajau
L39	827000	Talang Padang Lampung Api	L99	150000	Langara Laut Bajau
L40	827000	Way Kanan Lampung Api	L100	150000	Lapulu Bajau
L41	827000	Way Lima Lampung Api	L101	150000	Lauru Bajau
L42	770000	Nias Northern	L102	150000	Lemo Bajau
L43	750000	Batak Angkola	L103	150000	Luwuk Bajau
L44	750000	Sadan	L104	150000	Moramo Bajau
L45	700000	Uab Meto	L105	150000	Padei Laut Bajau
L46	600000	Karo Batak	L106	150000	Pitulua Bajau
L47	590000	Besemah	L107	150000	Samihim
L48	590000	Ogan	L108	150000	Tontemboan
L49	520000	Delang	L109	137000	Baree
L50	520000	Tamuan	L110	131000	Mambae
L51	500000	Bima	L111	130000	Tukang Besi Southern
L52	475000	Mandar	L112	128000	Selayar
L53	470000	Adumanis Ulu Komering	L113	125000	Banggai
L54	470000	Ilir Komering	L114	125000	Coastal Konjo
L55	470000	Kayu Agung Asli Komering	L115	125000	Konjo
L56	470000	Kayu Agung Pendatang Komering	L116	120000	Tukang Besi Northern
L57	470000	Komering	L117	110000	Ende
L58	470000	Perjaya Ulu Komering	L118	110000	Savu
L59	463500	Tetun	L119	105000	Lio
L60	350000	Rejang			

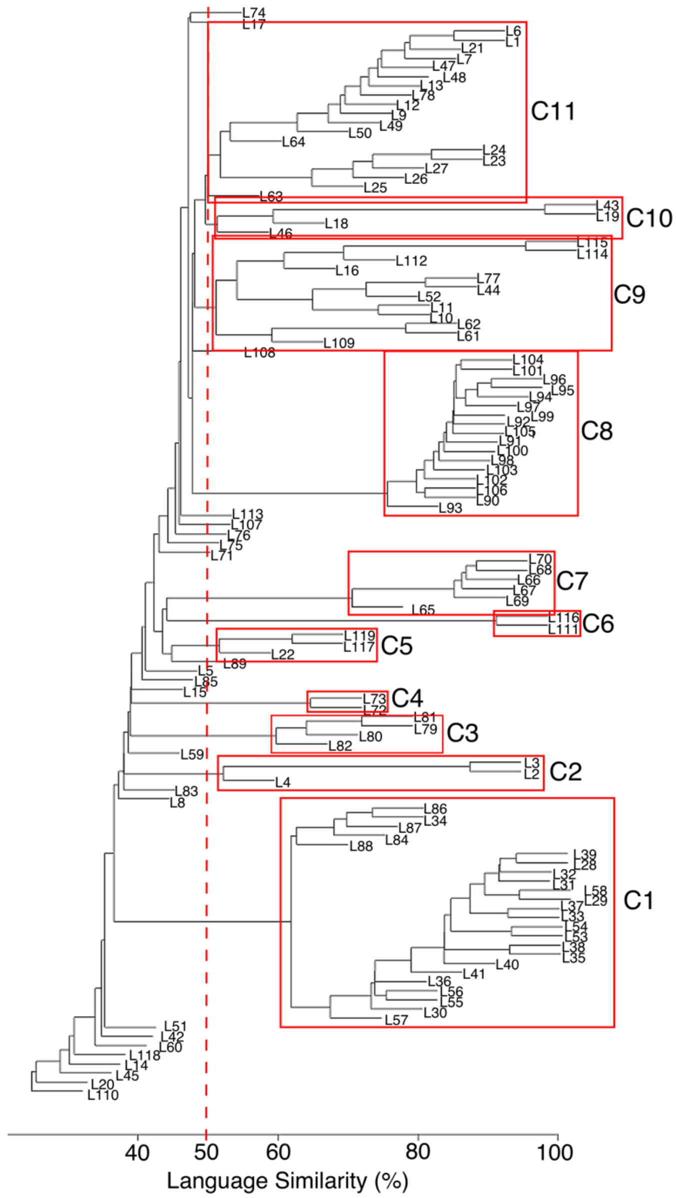

Fig. 2. Eleven Clusters with a Threshold of Language Similarity Range = 50% - 100% Created Using Hierarchical Clustering

We developed a language similarity clusters visualization[3], an online tool to create Indonesian ethnic language similarity clusters given a language similarity range as a threshold and visualize them with a language similarity range and a minimum number of speakers as query. For example, setting language similarity range from 0% to 100% as threshold will create one cluster for all 119 languages as shown in Fig. 3. To re-cluster the languages with a language similarity range threshold from 50% to 100%, we can set the threshold accordingly and check the box “Use the above language similarity range to recreate the clusters" before submitting the query. The generated 11 clusters shown in Fig. 4 are exactly the same as the hierarchical clustering clusters in Fig. 2. This shows that we can replace the hierarchical clustering approach with the graph-based clustering

[3] http://langsphere.org/idcluster

approach. Moreover, we can explore and analyze the generated clusters compared to the static result of the hierarchical clustering approach.

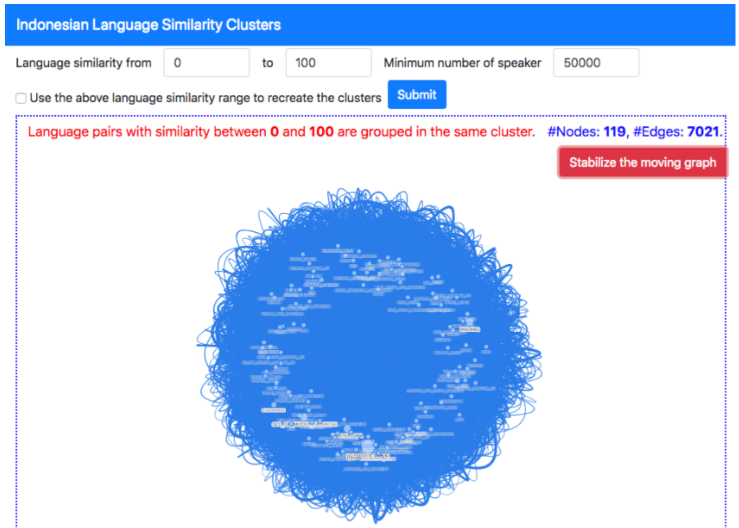

Fig. 3. One Cluster with a Threshold of Language Similarity Range = 0% - 100%

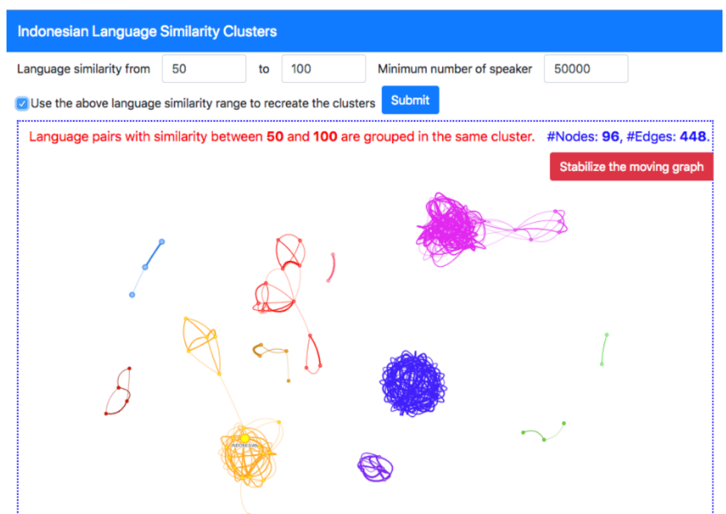

Fig. 4. Eleven Clusters with a Threshold of Language Similarity Range = 50% - 100%

Notice that there are 11 clusters created with 23 less languages (represented as nodes) than the single cluster in Fig. 3. This means that for those 23 languages presented in Table II, each language has no higher similarity than 50% to any other languages. The existing bilingual dictionary induction method [8] works best on closely-related languages. If we only consider the closeness of a language to other languages to select target languages, those 23 languages are not a good candidate. However, in practice the number of speakers also plays an important role in selecting the target languages, so that the generated bilingual dictionaries can be used by many people. Therefore, the proposed system should allow exploration on the graph. In this example, we keep the current 11 clusters and submit two queries by setting the language similarity range and the minimum number of speakers to explore the data as shown in Fig. 5 and Fig. 6. The language similarity between two languages is shown by mouse hovering the edge as shown in Fig. 5. The information about a language (name, language code, cluster number and number of speaker) is also shown by mouse hovering the node as shown in Fig. 6. In Fig. 5 we found that some languages like karo batak, tae, sadan, and tolaki are connectors to other clusters. These languages can be a good pivot when using bilingual dictionary induction method [8]. These connector languages cannot be identified from the hierarchical clustering in Fig. 2.

TABLE II
23 LANGUAGES WITH NO HIGHER SIMILARITY THAN 50% TO ANY OTHER LANGUAGES

Code	Speaker	#Language	Code	Speaker	#Language
L5	34000000	Sundanese	L74	300000	Sumbawa
L8	6770900	Madurese	L75	285000	Kerinci
L14	3500032	Aceh	L76	255000	Sangir
L15	3330000	Bali	L83	230000	Mongondow
L17	2100000	Sasak	L85	180000	Lamaholot Ile Mandiri
L20	1000000	Gorontalo	L89	175000	Sika
L42	770000	Nias Northern	L107	150000	Samihim
L45	700000	Uab Meto	L108	150000	Tontemboan
L51	500000	Bima	L113	125000	Banggai
L59	463500	Tetun	L118	110000	Savu
L71	300000	Gayo			

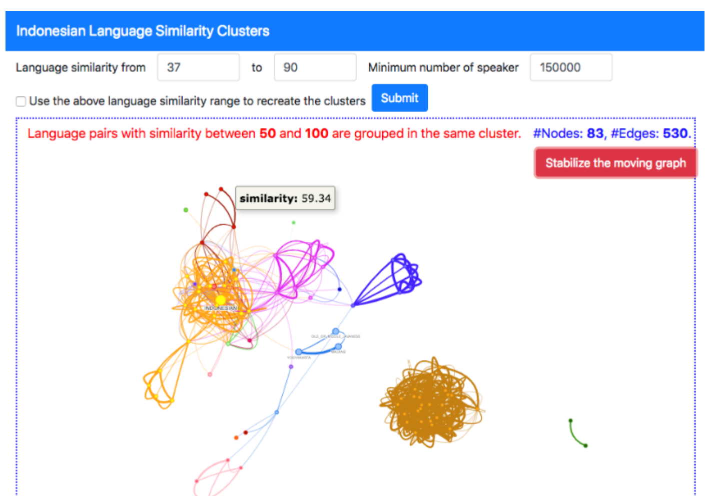

Fig. 5. Query 1: Language Similarity Range = 37% - 90% AND #Speaker >= 150K

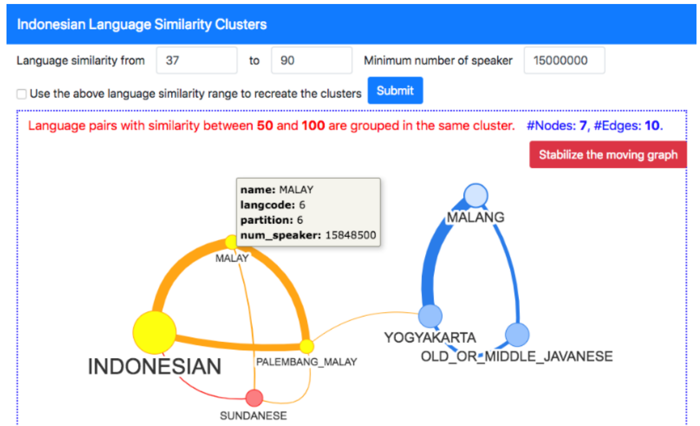

Fig. 6. Query 2: Language Similarity Range = 37% - 90% AND #Speaker >= 15M

V. SYSTEM PERFORMANCE STABILITY EVALUATION

The Indonesian language similarity clusters system run on a virtual private server with the following specification: four cores Intel Xeon E5-2620v3, CentOS 7 (64 Bit), and 12 GB RAM. The system performance stability is evaluated based on the response time to a query. In this experiment, throughout all trials, we maintain the 11 clusters with language lexical similarity range from 50% to 100% as shown in Fig. 4. For the exploration, we use a language similarity range from 0% to 100% in the query so that every node is connected to each other. When we enter the number of speakers >= 105,000 in the query, the system will get the highest execution burden and will output all nodes and edges as shown in Fig. 3. In contrary, when we enter the number of speakers >= 900,000 in the query, the system will get the least execution burden due to small number of languages returned. We divided the execution burden into 5 different queries with 10 trials for each query to measure the performance stability.

TABLE III
RESPONSE TIME OF QUERIES WITH LANGUAGE SIMILARITY RANGE FROM 0% TO 100%

Trial	#Speaker	#Nodes	#Edges	Average Response Time (miliseconds)
1-10	>= 900,000	22	231	2,254.57
11-20	>= 700,000	45	990	4,803.08
21-30	>= 331,000	70	2,415	9,371.67
31-40	>= 175,000	89	3,916	14,546.80
41-50	>= 105,000	119	7,021	20,054.39

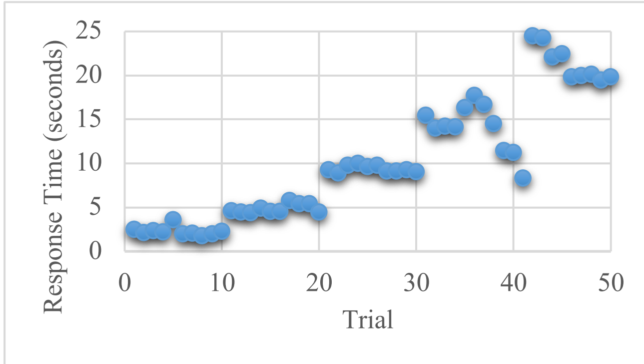

Fig. 7. Response Time of Language Similarity Clusters Visualization

Table III presents the minimum number of speakers, the number of nodes (languages), the number of edges (language lexical similarity), and the average response time for each query. The detailed response time for all 10 trials of each query is presented in Fig. 7. The experiment result shows that for the case of low execution burden, i.e., trial 1-30, the system performance was quite stable. However, for the case of high execution burden, i.e., trial 31-50, the system performance was fluctuated.

VI. Conclusion

We formalize a graph-based approach of creating and visualizing language similarity clusters by utilizing ASJP database to generate the language similarity matrix, then formalize the data as an undirected graph. To create the clusters, we apply a connected components algorithm with a threshold of language similarity range. Our graph-based clustering approach outperformed the existing hierarchical clustering and k-means clustering approach in regard to variety of language similarity range and visualization. Our interactive online tool allows user to dynamically create new clusters with any range of language similarity and explore the data visually based on language similarity range and number of speakers. We provide an implementation example of our approach on 119 Indonesian ethnic languages. The experiment result shows that for the case of low system execution burden, the system performance was quite stable. Even though for the case of high system execution burden, the performance was fluctuated, the response times were still below 25 seconds, which is considerably accepted. Our approach is scalable and can be applied to the other 7,000 languages available in ASJP database.

ACKNOWLEDGMENT

This research was partially supported by a Grant-in-Aid for Young Scientists (A) (17H04706, 2017-2020) from Japan Society for the Promotion of Science (JSPS). The first author was supported by Universitas Islam Riau.

REFERENCES

[1] T. Ishida, Y. Murakami, D. Lin, T. Nakaguchi, and M. Otani, "Language Service Infrastructure on the Web: The Language Grid," *Computer (Long. Beach. Calif).*, vol. 51, no. 6, pp. 72–81, Jun. 2018.

[2] T. Ishida, "Intercultural Collaboration and Support Systems: A Brief History," in *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2016)*, 2016, pp. 3–19.

[3] A. H. Nasution, N. Syafitri, P. R. Setiawan, and D. Suryani, "Pivot-Based Hybrid Machine Translation to Support Multilingual Communication," in *Proceedings - 2017 International Conference on Culture and Computing, Culture and Computing 2017*, 2017, vol. 2017-Decem.

[4] A. H. Nasution, "Pivot-based hybrid machine translation to support multilingual communication for closely related languages," *World Trans. Eng. Technol. Educ.*, vol. 16, no. 2, 2018.

[5] A. H. Nasution, Y. Murakami, and T. Ishida, "Designing a collaborative process to create bilingual dictionaries of Indonesian ethnic languages," in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2018.

[6] G. F. Simons and C. D. F. (eds.), *Ethnologue: Languages of the World*, 18th ed. Dallas, Texas: SIL International, 2015.

[7] A. H. Nasution, Y. Murakami, and T. Ishida, "Constraint-Based Bilingual Lexicon Induction for Closely Related Languages," in *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016, vol. 16, no. 1955.

[8] A. H. Nasution, Y. Murakami, and T. Ishida, "A Generalized Constraint Approach to Bilingual Dictionary Induction for Low-Resource Language Families," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 17, no. 2, pp. 1–29, 2017.

[9] M. Wushouer, D. Lin, T. Ishida, and K. Hirayama, "A Constraint Approach to Pivot-based Bilingual Dictionary Induction," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. V, no. 1, pp. 1–26, Nov. 2015.

[10] A. H. Nasution, Y. Murakami, and T. Ishida, "Plan optimization for creating bilingual dictionaries of low-resource languages," in *Proceedings - 2017 International Conference on Culture and Computing, Culture and Computing 2017*, 2017, vol. 2017-Decem.

[11] A. H. Nasution, Y. Murakami, and T. Ishida, "Generating Similarity Cluster of Indonesian Languages with Semi-Supervised Clustering," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 1, pp. 1–8, 2019.

[12] L. Campbell, "Historical linguistics: The state of the art," in *Linguistics Today – Facing a Greater Challenge*, 2014.

[13] W. P. Lehmann, *Historical linguistics: an introduction*. Routledge, 2013.

[14] L. Campbell and W. J. Poser, "Language classification," *Hist. method. Cambridge*, 2008.

[15] M. Swadesh, "Salish internal relationships," *Int. J. Am. Linguist.*, vol. 16, no. 4, pp. 157–167, 1950.

[16] M. Swadesh, "Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos," *Proc. Am. Philos. Soc.*, vol. 96, no. 4, pp. 452–463, 1952.

[17] M. Swadesh, *The origin and diversification of language*. Routledge, 2017.

[18] M. Swadesh, "Towards greater accuracy in lexicostatistic dating," *Int. J. Am. Linguist.*, vol. 21, no. 2, pp. 121–137, 1955.

[19] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, 1966, vol. 10, no. 8, pp. 707–710.

[20] B. Kessler, "Computational dialectology in irish gaelic," in *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, 1995, pp. 60–66.

[21] W. J. Heeringa, "Measuring dialect pronunciation differences using Levenshtein distance," Citeseer, 2004.

[22] C. Tang and V. J. Van Heuven, "Predicting mutual intelligibility of Chinese dialects from multiple objective linguistic distance measures," *Linguistics*, vol. 53, no. 2, pp. 285–312, 2015.

[23] F. Petroni and M. Serva, "Language distance and tree reconstruction," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 08, p. P08012, 2008.

[24] S. Wichmann, E. W. Holman, D. Bakker, and C. H. Brown, "Evaluating linguistic distance measures," *Phys. A Stat. Mech. its Appl.*, vol. 389, no. 17, pp. 3632–3639, 2010.

[25] E. W. Holman *et al.*, "Automated dating of the world's language families based on lexical similarity," *Curr. Anthropol.*, vol. 52, no. 6, pp. 841–875, 2011.

[26] E. W. Holman, S. Wichmann, C. H. Brown, V. Velupillai, A. Müller, and D. Bakker, "Explorations in automated language classification," *Folia Linguist.*, vol. 42, no. 3–4, pp. 331–354, 2008.

[27] B. A. Galler and M. J. Fisher, "An improved equivalence algorithm," *Commun. ACM*, vol. 7, no. 5, pp. 301–303, 1964.

[28] A. Charguéraud and F. Pottier, "Verifying the correctness and amortized complexity of a union-find implementation in separation logic with time credits," *J. Autom. Reason.*, vol. 62, no. 3, pp. 331–365, 2019.

[29] J. Jaiganesh and M. Burtscher, "A high-performance connected components implementation for GPUs.," in *HPDC*, 2018, pp. 92–104.