Similarity Cluster of Indonesian Ethnic Languages

by Nasution Arbi Haza

Submission date: 30-Oct-2019 04:11PM (UTC+0800)
Submission ID: 1203422387
File name: 4.pdf (1.36M)
Word count: 5189
Character count: 25306

Similarity Cluster of Indonesian Ethnic Languages

Arbi Haza Nasution[1], Yohei Murakami[2], Toru Ishida[3]

[1,3] Department of Social Informatics, Kyoto University, Japan

[2]Unit of Design, Kyoto University, Japan

[1]arbi@ai.soc.i.kyoto-u.ac.jp, [2]yohei@i.kyoto-u.ac.jp, [3]ishida@i.kyoto-u.ac.jp

---

Abstract

Lexicostatistic and language similarity clusters are useful for computational linguistic researches that depends on language similarity or cognate recognition. Nevertheless, there are no published lexicostatistic/language similarity cluster of Indonesian ethnic languages available. We formulate an approach of creating language similarity clusters by utilizing ASJP database to generate the language similarity matrix, then generate the hierarchical clusters with complete linkage and mean linkage clustering, and further extract two stable clusters with high language similarities. We introduced an extended k-means clustering semi-supervised learning to evaluate the stability level of the hierarchical stable clusters being grouped together despite of changing the number of cluster. The higher the number of the trial, the more likely we can distinctly find the two hierarchical stable clusters in the generated k-clusters. However, for all five experiments, the stability level of the two hierarchical stable clusters is the highest on 5 clusters. Therefore, we take the 5 clusters as the best clusters of Indonesian ethnic languages. Finally, we plot the generated 5 clusters to a geographical map.

Keywords*: lexicostatistic, language similarity, hierarchical clustering, k-means clustering*

---

1. INTRODUCTION

Nowadays, machine-readable bilingual dictionaries are being utilized in actual services (Ishida, 2011) to support intercultural collaboration (Ishida, 2016; Nasution et al., 2017c), but low-resource languages lack such sources. 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 (Wushoer et al., 2015; Nasution et al., 2016; Nasution et al., 2017a; Nasution et al., 2017b). Those previous researchers requires lexicostatistic/language similarity clusters of the low-resource languages to select the target languages. However, to the best of our knowledge, there are no published lexicostatistic/language similarity clusters of Indonesian ethnic languages. To fill the void, we address this research goal:

- *Formulating an approach of creating a language similarity cluster*. We first obtain 40-item word lists from the Automated Similarity Judgment Program (ASJP), further generate the language similarity matrix, then generate the hierarchical and k-means clusters, and finally plot the generated clusters to a map.

2. INDONESIAN ENDANGERED LANGUAGES

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 (Lewis et al., 2015). 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 already extinct. Figure 1 shows the level of development or endangerment of Indonesian ethnic languages. (Lewis et al., 2015)

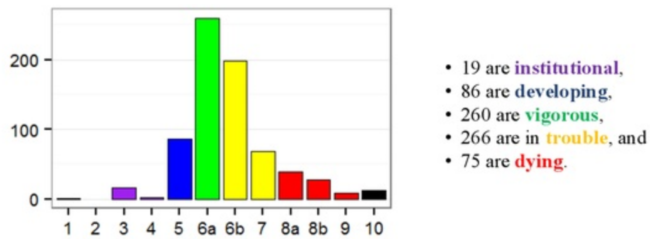

Figure 1. Indonesian Ethnic Languages Level of Development or Endangerment

Here are the definitions of each level of Development or Endangerment:

- *Institutional (EGIDS 0-4)* — The language has been developed to the point that it is used and sustained by institutions beyond the home and community.
  - Buginese (3 (Wider communication), 5,000,000), Javanese (4 (Educational), 84,300,000)
- *Developing (EGIDS 5)* — The language is in vigorous use, with literature in a standardized form being used by some though this is not yet widespread or sustainable.
  - Minangkabau (5 (Developing), 5,530,000), Bali (5 (Developing), 3,330,000)
- *Vigorous (EGIDS 6a)* — The language is unstandardized and in vigorous use among all generations.
  - Iranun (6a (Vigorous), 256,000), Batak Mandailing (6a (Vigorous), 1,100,000)
- *In trouble (EGIDS 6b-7)* — Intergenerational transmission is in the process of being broken, but the child-bearing generation can still use the language so it is possible that revitalization efforts could restore transmission of the language in the home.
  - Temuan (6b (Threatened), 22,700 (2008 JHEOA)), Tambunan Dusun (6b (Threatened), 15,600 (2000))
- *Dying (EGIDS 8a-9)* — The only fluent users (if any) are older than child-bearing age, so it is too late to restore natural intergenerational transmission through the home; a mechanism outside the home would need to be developed.
  - Nusa Laut (9 (Dormant), 2,230 (1989 SIL)), Ura (8b (Nearly extinct),
- *Extinct (EGIDS 10)* — The language has fallen completely out of use and no one retains a sense of ethnic identity associated with the language.
  - Kaniet (10 (Extinct)), Uruava (10 (Extinct))

3. 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 (Campbell, 2013). Comparative linguistics is a branch of historical linguistics that is concerned with language comparison to determine historical relatedness and to construct language families (Lehmann, 2013). 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 (Campbell, L. and Poser, W.J., 2008). 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 as shown in Table 1. To find the best size of the list, Swadesh 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." (Swadesh, 1955)

Table 1. Modification of Swadesh List

Published Year	**Number of Words**
1950	225 (Swadesh, 1950)
1952	215 & 200 (Swadesh, 1952)
1971 & 1972	100 (Swadesh, 1971)

Table 2. Levenshtein Distance Algorithm

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m)
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately **above plus 1: d[i-1, j] + 1** b. The cell immediately to the **left plus 1: d[i, j-1] + 1** c. The cell **diagonally above and to the left plus the cost: d[i-1, j-1] + cost**
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n, m]

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 (Levenshtein, 1966). The Levenshtein Distance algorithm is shown in Table 2. 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). Another example between Indonesian word is LD between “satu" and “baru" is 2 since there are only two transformations needed: satu → batu (substitution of “b" for “s") and then batu → baru (substitution of “r" for “t") as shown in Figure 2.

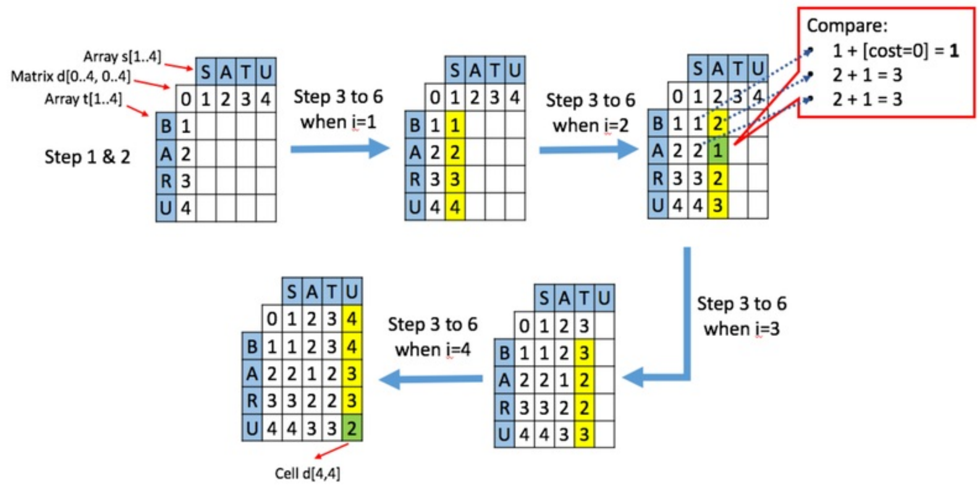

Figure 2. Example of transformations following Levenshtein Distance Algorithm

There are a lot of previous works using Levenshtein Distances such as dialect groupings of Irish Gaelic (Kessler, 1995) 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 (Heeringa, 2004) which obtain 125 words from Reeks Nederlandse Dialectatlassen. They normalize LD by dividing it by the length of the longer alignment. Tang (2015) measure linguistic similarity and intelligibility of 15 Chinese dialects and obtain 764 common syllabic units. Petroni (2008) define lexical distance between two words as the LD normalized by the number of characters of the longer of the two. Wichmann et al. (2010) 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. (2011) with the main goal of developing a database of Swadesh lists (Swadesh, 1955) 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 classification is based on 100-item reference list of Swadesh (Swadesh, 1971) and further reduced to 40 most stable items (Holman et al., 2008). 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.

4. LANGUAGE SIMILARITY CLUSTERING APPROACH

We formalize an approach to create language similarity clusters by utilizing ASJP database to generate the language similarity matrix, then generate the hierarchical clusters, and further extract the stable clusters with high language similarities. The hierarchical stable clusters are evaluated utilizing our extended k-means clustering. Finally, the obtained k-means clusters are plotted to a geographical map. The flowchart of the whole process is shown in Figure 3.

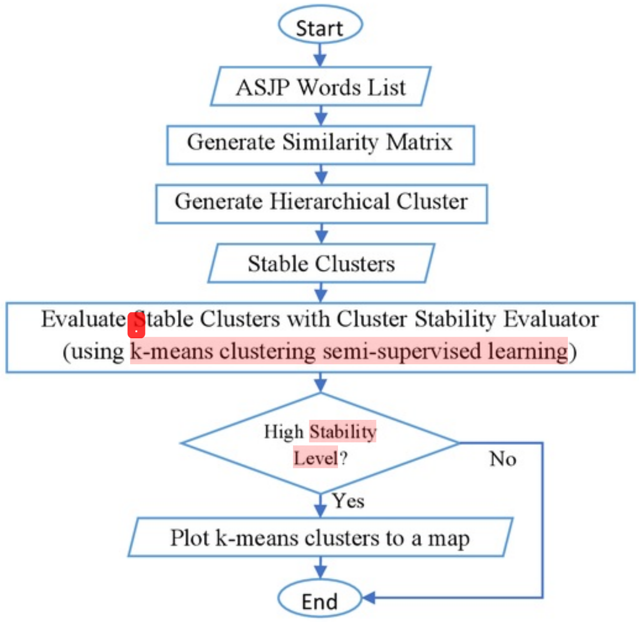

Figure 3. Flowchart of Generating Language Similarity Clusters

In this paper, we focus on Indonesian ethnic languages. We obtain words list of 119 Indonesian ethnic languages with the number of speakers at least 100,000. We further generate the similarity matrix ranked by the number of speakers as shown in Figure 4. We added a white-red color scale where white color means the two languages are totally different (0% similarity) and the reddest color means the two languages are exactly the same (100% similarity).

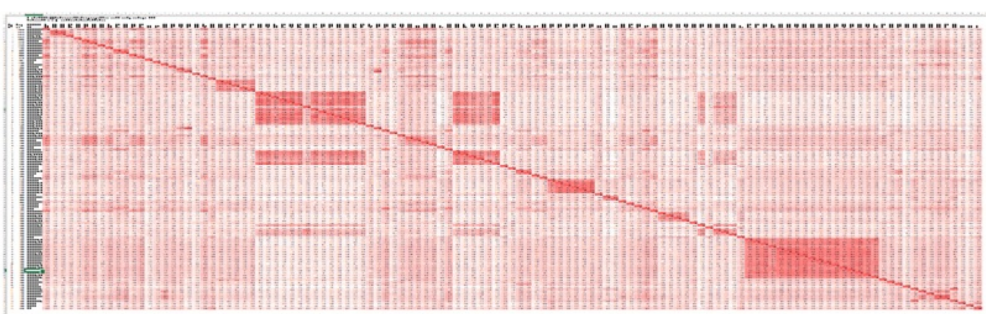

Figure 4. Language Similarity Matrix of 119 Indonesian Ethnic Languages

However, it is difficult to classify 119 languages and obtain a valuable information from the generated clusters, therefore, we further filtered the target languages based on the number of speaker and availability of the language information in Wikipedia. We obtain 32 target languages as shown in Table 3 from the intersection between 46 Indonesian ethnic languages with number of speaker above 300,000 provided by Wikipedia and 119 Indonesian ethnic languages with number of speaker above 100,000 provided by ASJP.

Table 3. List of 32 Indonesian Ethnic Languages Ranked by Population

Code	Ranked by Wikipedia	Ranked by AJSP	Population based on Wikipedia	Population based on AJSP	Language
L 1	1	1	210000000	232004800	INDONESIAN
L 2	3	2	84300000	84300000	OLD_OR_MIDDLE_JAVANESE
L 3	4	3	34000000	34000000	SUNDANESE
L 4	2	4	210000000	15848500	MALAY
L 5	7	5	3900000	15848500	PALEMBANG_MALAY
L 6	5	6	13600000	6770900	MADURESE
L 7	6	7	5500000	5530000	MINANGKABAU
L 8	8	8	3500000	5000000	BUGINESE
L 9	12	9	2700000	5000000	BETAWI
L 10	9	10	3500000	3502300	BANJARESE_MALAY
L 11	10	11	3500000	3500032	ACEH
L 12	11	12	3300000	3330000	BALI
L 13	16	13	1600000	2130000	MAKASAR
L 14	13	14	2700000	2100000	SASAK
L 15	14	15	2000000	2000000	TOBA_BATAK
L 16	17	16	1100000	1100000	BATAK_MANDAILING
L 17	18	17	1000000	1000000	GORONTALO
L 18	19	18	900000	1000000	JAMBI_MALAY
L 19	27	19	500000	900000	MANGGARAI
L 20	21	20	800000	770000	NIAS_NORTHERN
L 21	22	21	700000	750000	BATAK_ANGKOLA
L 22	24	22	600000	700000	UAB_METO
L 23	23	23	600000	600000	KARO_BATAK
L 24	25	24	500000	500000	BIMA
L 25	26	25	500000	470000	KOMERING
L 26	28	26	400000	350000	REJANG
L 27	32	27	300000	331000	TOLAKI
L 28	29	28	300000	300000	GAYO
L 29	30	29	300000	300000	MUNA
L 30	31	30	300000	250000	TAE
L 31	15	31	1900000	245020	AMBONESE_MALAY
L 32	20	32	900000	230000	MONGONDOW

We further generate the similarity matrix of those 32 languages as shown in Table 4. We also added a white-red color scale where white color means the two languages are totally different (0% similarity) and the reddest color means the two languages are exactly the same (100% similarity). For a better clarity and to avoid redundancy, we only show the bottom-left part of the table. The headers follow the language code in Table 3.

Table 4. Lexicostatistic / Similarity Matrix of 32 Indonesian Ethnic Languages by ASJP (%)

	L 1	L 2	L 3	L 4	L 5	L 6	L 7	L 8	L 9	L 10	L 11	L 12	L 13	L 14	L 15	L 16	L 17	L 18	L 19	L 20	L 21	L 22	L 23	L 24	L 25	L 26	L 27	L 28	L 29	L 30	L 31
L 2	24																														
L 3	39	22																													
L 4	85	21	41																												
L 5	68	32	39	73																											
L 6	34	15	20	34	34																										
L 7	62	25	31	62	64	34																									
L 8	31	18	25	32	31	18	32																								
L 9	69	10	25	67	58	23	50	24																							
L 10	72	33	39	71	64	34	60	33	55																						
L 11	27	11	19	27	30	22	25	16	21	25																					
L 12	38	20	29	35	39	23	31	30	24	37	22																				
L 13	33	22	24	30	32	25	33	36	25	33	16	29																			
L 14	44	20	28	42	44	30	44	31	37	47	22	29	35																		
L 15	37	24	23	37	36	21	40	25	35	37	13	21	25	35																	
L 16	25	16	14	27	27	20	27	23	24	25	14	20	18	24	58																
L 17	19	14	16	18	19	9	18	20	14	17	12	12	18	20	17	9															
L 18	79	26	40	78	78	34	69	31	70	73	27	35	38	46	39	21	20														
L 19	30	18	24	30	34	19	32	36	26	32	10	23	29	31	32	21	16	34													
L 20	26	21	17	23	25	13	29	26	24	29	12	16	19	24	29	21	19	24	25												
L 21	24	16	15	26	26	19	26	21	21	24	12	21	18	23	59	98	9	20	19	20											
L 22	13	10	9	11	14	12	18	19	10	19	10	12	21	18	15	9	14	15	22	16	9										
L 23	47	22	28	48	50	23	40	30	40	44	21	32	27	35	51	40	17	47	28	33	40	12									
L 24	18	10	16	17	18	12	18	21	18	19	6	14	21	25	22	14	8	17	30	19	14	18	19								
L 25	33	19	25	33	33	18	25	23	29	36	14	23	22	22	24	24	16	30	26	29	25	20	36	14							
L 26	28	20	16	27	32	18	30	17	21	29	15	17	17	30	25	20	11	32	18	15	19	12	29	4	19						
L 27	30	14	18	28	27	17	26	32	23	33	11	21	27	21	26	14	11	28	36	25	14	19	28	26	20	13					
L 28	37	27	28	36	37	20	37	26	28	38	18	25	23	35	28	18	17	40	26	23	17	20	41	18	37	29	28				
L 29	14	12	12	14	13	13	11	21	18	12	8	16	24	14	14	9	11	13	15	15	10	11	14	21	14	4	29	11			
L 30	42	29	31	41	39	27	42	60	30	47	20	28	42	40	34	27	23	44	38	35	26	29	38	30	29	21	38	38	25		
L 31	72	23	35	70	58	37	59	36	62	60	23	34	36	43	33	28	19	69	33	29	26	17	36	19	29	24	29	31	16	42	
L 32	30	18	24	32	31	13	26	26	27	34	11	21	25	24	24	17	26	32	23	24	17	12	28	14	24	20	20	27	15	38	24

Hierarchical clustering is an approach which builds a hierarchy from the bottom-up, and does not require us to specify the number of clusters beforehand. The algorithm works as follows:

- Put each data point in its own cluster
- Identify the closest two clusters and combine them into one cluster
- Repeat the above step until all the data points are in a single cluster

Once this is done, it is usually represented by a dendrogram like structure. There are a few ways to determine how close two clusters are:

- ✓ Complete linkage clustering: Find the maximum possible distance between points belonging to two different clusters.
- ✓ Single linkage clustering: Find the minimum possible distance between points belonging to two different clusters.
- ✓ Mean/Average linkage clustering: Find all possible pairwise distances for points belonging to two different clusters and then calculate the average.
- ✓ Centroid linkage clustering: Find the centroid of each cluster and calculate the distance between centroids of two clusters.

Complete linkage and mean (average) linkage clustering are the ones used most often. We generate the distance matrix from the similarity matrix shown in Table 4 and further generate the hierarchical clusters with hclust function with a complete linkage clustering method as shown in Figure 5 and a mean linkage clustering method as shown in Figure 6 using R[1], a free software environment for statistical computing and graphics.

From those two hierarchical clusters in Figure 5 and Figure 6, we select two stable clusters that always grouped together despite of changing the linkage clustering method. The first cluster consists of TOBA_BATAK, BATAK_MANDAILING, and BATAK_ANGKOLA, while the second cluster consists of MINANGKABAU, BETAWI, AMBONESE_MALAY, BANJARESE_MALAY, PALEMBANG_MALAY, JAMBI_MALAY, MALAY, and Indonesia. Since the two stable custers have language similarities above 50% between the languages, they are good clusters to be referred when selecting target languages for computational linguistic researches that depends on language similarity or cognate recognition for inducing bilingual lexicons from the target languages (Mann, G.S., and Yarowsky, D., 2001; Wushouer et al., 2015; Nasution et al., 2016; Nasution et al., 2017a). The two clusters are actually enough for selecting the target languages for those researches. However, we still need to evaluate the stability of those clusters and we also need to identify the low language similarities clusters in order to graps the whole picture of Indonesian ethnic languages. Thus, we utilize the alternative clustering approach which is a k-means clustering.

[1] https://www.r-project.org/

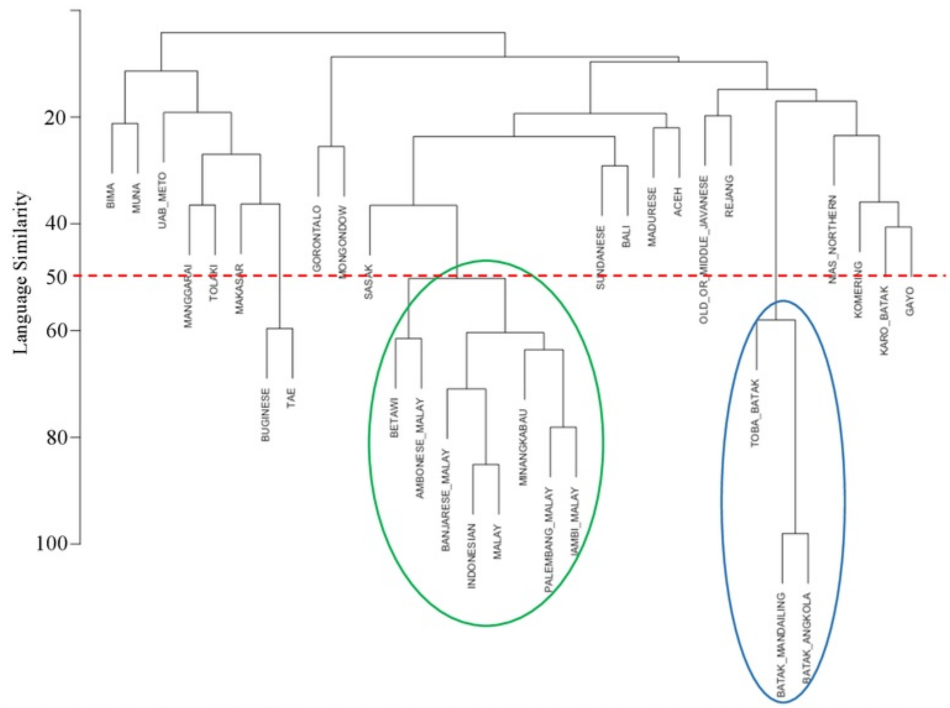

Figure 5. Hierarchical Clusters Dendogram of 32 Indonesian Ethnic Languages – method: complete

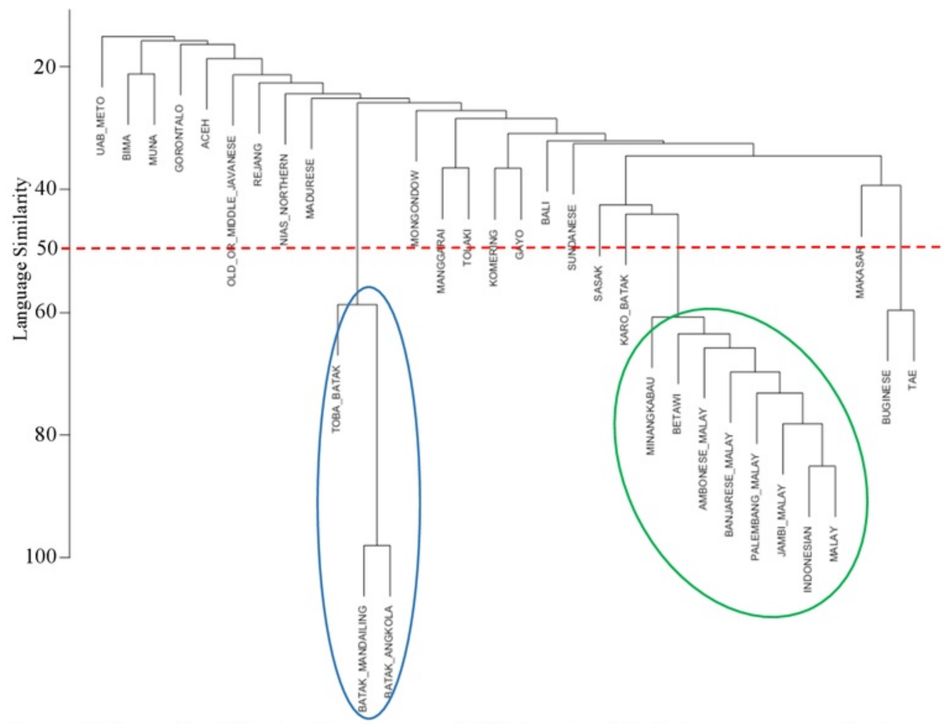

Figure 6. Hierarchical Clusters Dendogram of 32 Indonesian Ethnic Languages – method: average

K-means clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. In k-means clustering, we have to specify the number of clusters we want the data to be grouped into. The algorithm works as follows:

- The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster.
- Then, the algorithm iterates through two steps:
  - Reassign data points to the cluster whose centroid is closest.
  - Calculate new centroid of each cluster.

These two steps are repeated until the within cluster variation cannot be reduced any further. The within cluster variation is calculated as the sum of the euclidean distance between the data points and their respective cluster centroids.

It is well known that standard agglomerative hierarchical clustering techniques are not tolerant to noise (Nagy, 1968; Narasimhan et al., 2006). There are many previous works on finding clusters which robust to noise (Guha et al., 1999; Langfelder, P., & Horvath, S., 2012; Balcan et al., 2014). However, to evaluate the stability of the hierarchical stable clusters, we introduced a simple approach of calculating their stability level of being grouped together despite of changing the number of k-means clusters. We extend the k-means clustering unsupervised learning to a k-means clustering semi-supervised learning by labeling the two hierarchical stable clusters beforehand.

ALGORITHM 1: Cluster Stability Evaluator

Input: *similarity_matrix, stable_clusters, minimum_k, maximum_trial*
Output: *stability_level*
trial ← *1*
current_k ← *minimum_k*
maximum_k ← *length(similarity_matrix)*
scale2D ← *cmdscale(similarity_matrix) //multidimensional to 2D scaling*
while *current_k <= maximum_k,* **do**
    successful_trial ← *0*   *// initialized for each current_k*
    while *trial <= maximum_trial*, **do**
        k-clusters ← *kmeans(scale2D, current_k)*
        if *stable_clusters* distinctly found in *k-clusters*, **then**
            successful_trial++
            trial++   *// try again with the same number of cluster (current_k)*
    end
    stability_level[current_k] = successful_trial / maximum_trial
    current_k++   *// increase the number of clusters*
    trial = 1   *// reset the number of trial*
end
return *stability_level*

Initially, we manually conduct several trials to estimate the minimum and maximum number of k-means cluster to obtain clusters which consist of the stable clusters distinctly. Based on the initial trials, we estimate the *minimum_k* = 4 and *maximum_k* = 21. Then, we calculate the stability level of the two hierarchical stable clusters where the number of clusters ranging from *minimum_k* = 4 to *maximum_k* = 21 following Algorithm 1. We have five sets of experiments with the *maximum_trial* equals 50, 500, 5,000, 50,000, and 500,000. In each experiment, a stability level of the two hierarchical stable clusters is measured for each number of k-means clusters by calculating the success rate of obtaining the two hierarchical stable clusters in the generated *k-clusters* as shown in Figure 7 to 11.

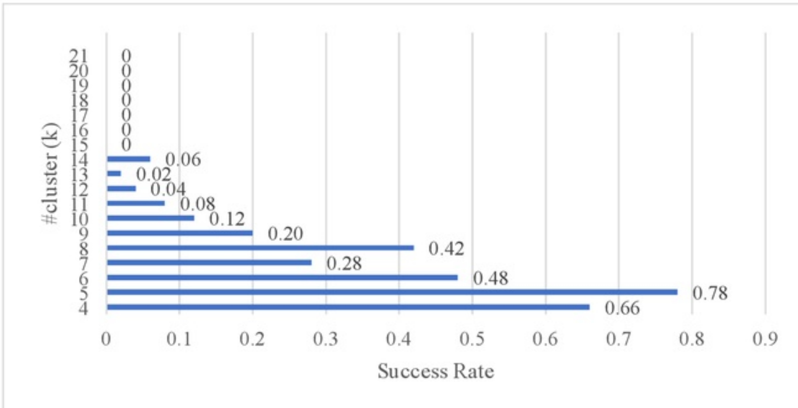

Figure 7. Obtaining Stable Clusters in 50 Trials

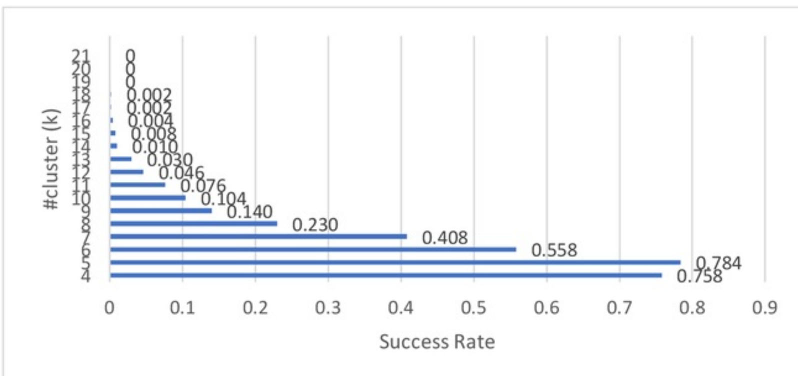

Figure 8. Obtaining Stable Clusters in 500 Trials

The higher the number of the trial, the more likely we can distinctly find the two hierarchical stable clusters in the generated *k-clusters* with a big number of clusters. For example, within 50 trials, we can not find the two hierarchical stable clusters distinctly in the generated *k-clusters* for big number of clusters (k>14). However, within 50,000 and 500,000 trials, we can find the two hierarchical stable clusters distinctly in the generated *k-clusters* for all number of clusters between the *minimum_k* = 4 and the *maximum_k* = 21, even though the success rate is getting lower as the number of clusters increases. For all five experiments, the stability level of the two hierarchical stable clusters is the highest (0.78) on 5 clusters.

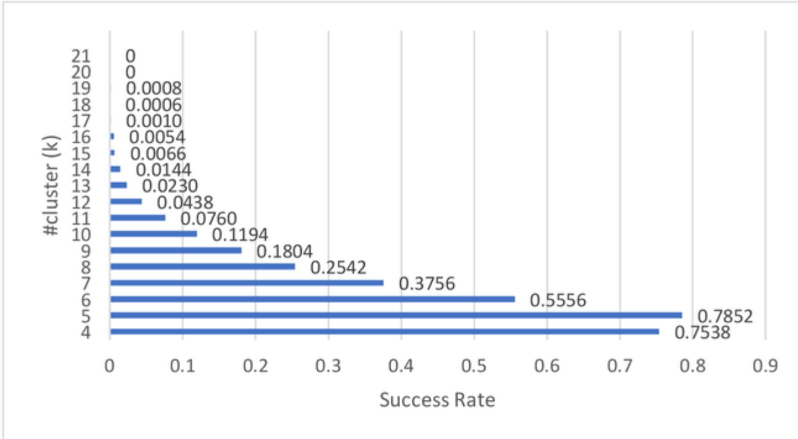

Figure 9. Obtaining Stable Clusters in 5,000 Trials

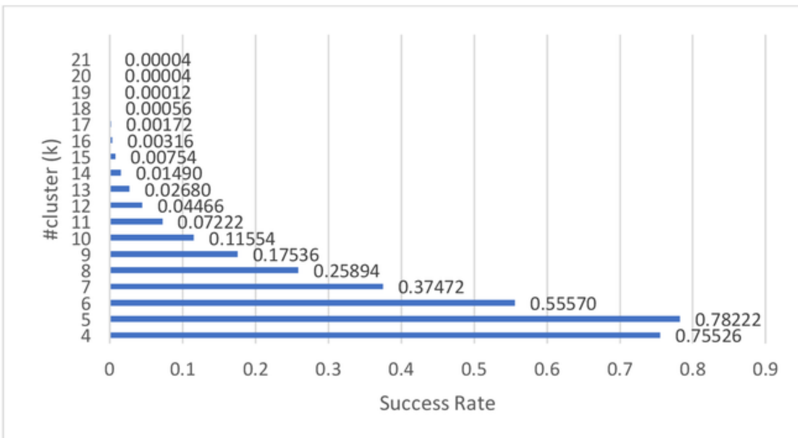

Figure 10. Obtaining Stable Clusters in 50,000 Trials

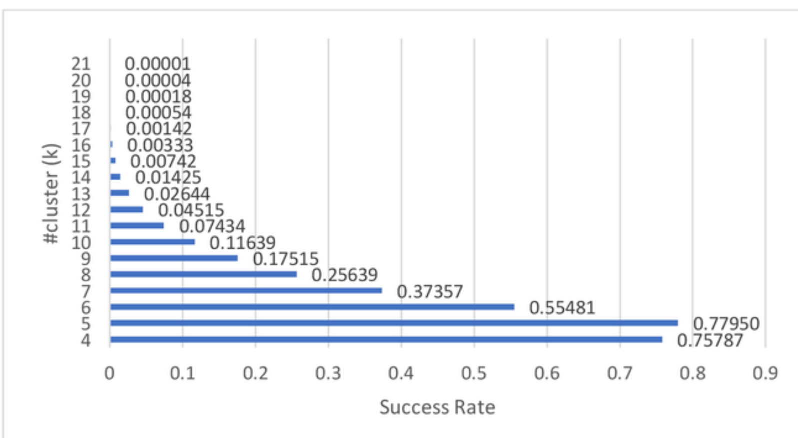

Figure 11. Obtaining Stable Clusters in 500,000 Trials

Therefore, we take the 5 clusters as shown in Figure 12 as the best clusters of Indonesian ethnic languages to be referred when selecting target languages for computational linguistic researches that depends on language similarity or cognate recognition. We further plot the 5 clusters to a geographical map as shown in Figure 13.

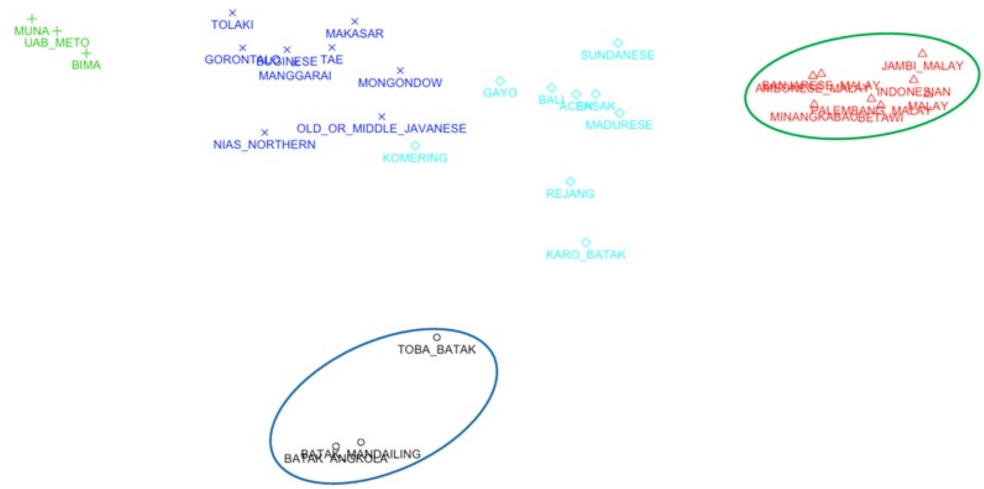

Figure 12. K-means Clusters of 32 Indonesian Ethnic Languages – 5 Clusters

Figure 13. Similarity Clusters Map of 32 Indonesian Ethnic Languages – 5 Clusters

4. CONCLUSION

We utilized ASJP database to generate the language similarity matrix, then generate the hierarchical clusters with complete linkage and mean linkage clustering, and further extract two stable clusters with the highest language similarities. We apply our extended k-means clustering semi-supervised learning to evaluate the stability level of the hierarchical stable clusters being grouped together despite of changing the number of clusters. The higher the number of the trial, the more likely we can distinctly find the two hierarchical stable clusters in the generated *k-clusters*. However, for all five experiments, the stability level of the two hierarchical stable clusters is the highest (0.78) on 5 clusters. Therefore, we take the 5 clusters as the best clusters of Indonesian ethnic languages to be referred to select target languages for computational linguistic researches that depends on language similarity or cognate recognition. Finally, we plot the generated 5 clusters to a geographical map. Our algorithm can be used to find and evaluate other stable clusters of Indonesian ethnic languages or other language sets.

ACKNOWLEDGMENT

This research was partially supported by a Grant-in-Aid for Scientific Research (A) (17H00759, 2017-2020) and 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 Indonesia Endownment Fund for Education (LPDP).

BIBLIOGRAPHY

Balcan, M. F., Liang, Y., & Gupta, P. (2014). Robust hierarchical clustering. The Journal of Machine Learning Research, 15(1), 3831-3871.

Campbell, L. (2013). *Historical Linguistics*. Edinburgh University Press.

Campbell, L. and Poser, W.J. (2008). *Language classification. History and method.* Cambridge.

Guha, S., Rastogi, R., & Shim, K. (1999, March). ROCK: A robust clustering algorithm for categorical attributes. In Data Engineering, 1999. Proceedings., 15th International Conference on (pp. 512-521). IEEE.

Holman, E.W., Brown, C.H., Wichmann, S., Mu ̈ller, A., Velupillai, V., Hammarstro ̈m, H., Sauppe, S., Jung, H., Bakker, D., Brown, P., and others. (2011). Automated dating of the world's language families based on lexical similarity. Current Anthropology 52, 6 (2011), 841–875.

Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Müller, A., & Bakker, D. (2008). Explorations in automated language classification. *Folia Linguistica*, *42*(3-4), 331-354.

Ishida, T. editor. (2011). *The Language Grid: Service- Oriented Collective Intelligence for Language Resource Interoperability*. Springer Publishing Company, Incorporated.

Ishida, T. (2016). Intercultural collaboration and support systems: A brief history. In International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2016), pages 3–19. Springer.

Langfelder, P., & Horvath, S. (2012). Fast R functions for robust correlations and hierarchical clustering. Journal of statistical software, 46(11).

Lehmann, W.P. (2013). *Historical linguistics: an introduction*. Routledge.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).

Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig (eds.) (2015). *Ethnologue: Languages of the World, Eighteenth edition*. Dallas, Texas: SIL International. Online version: http://www.ethnologue.com

Mann, G.S., and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. Association for Computational Linguistics, 1–8.

Nagy, G. (1968). State of the art in pattern recognition. Proceedings of the IEEE.

Narasimhan, M., Jojic N., and Bilmes, J. (2006). Q-clustering. Advances in Neural Information Processing Systems.

Nasution, A. H., Murakami, Y., and Ishida, T. (2016). Constraint-based bilingual lexicon induction for closely related languages. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pages 3291–3298, Paris, France, May.

Nasution, A. H., Murakami, Y., and Ishida, T. (2017a). A generalized constraint approach to bilingual dictionary induction for low-resource language families. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 17(1):1– 29, September.

Nasution, A. H., Murakami, Y., and Ishida, T. (2017b). Plan optimization for creating bilingual dictionaries of low-resource languages. In 2017 International Conference on Culture and Computing (Culture Computing), Sep.

Nasution, A. H., Syafitri, N., Setiawan, P. R., and Suryani, D. (2017c). Pivot-based hybrid machine translation to support multilingual communication. In 2017 International Conference on Culture and Computing (Culture Computing), Sep.

Swadesh, Morris. (1950). Salish Internal Relationships. *International Journal of American Linguistics*, Vol. 16, 157–167

Swadesh, Morris. (1952). Lexicostatistic Dating of Prehistoric Ethnic Contacts. Proceedings of the American Philosophical Society, Vol. 96, 452–463.

Swadesh, Morris. (1955). Towards Greater Accuracy in Lexicostatistic Dating. International Journal of American Linguistics, Vol. 21, 121–137.

Swadesh, Morris. (1971). The Origin and Diversification of Language. Ed. post mortem by Joel Sherzer. Chicago: Aldine, p. 283.

Tang, C., & van Heuven, V. J. (2015). Predicting mutual intelligibility of Chinese dialects from multiple objective linguistic distance measures.*Linguistics*, *53*(2), 285-312.

Wichmann, S., Holman, E. W., Bakker, D., & Brown, C. H. (2010). Evaluating linguistic distance measures. *Physica A: Statistical Mechanics and its Applications*, *389*(17), 3632-3639.

Wushouer, M., Lin, D., Ishida, T., and Hirayama, K. (2015). A constraint approach to pivot-based bilingual dictionary induction. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 15(1):4:1–4:26, November.

Similarity Cluster of Indonesian Ethnic Languages

ORIGINALITY REPORT

34%	33%	14%	17%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

12%
★ www.iaescore.com
Internet Source

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		