edumatic

by Wirarama Wedashwara

Submission date: 07-Mar-2022 02:13PM (UTC+0700)
Submission ID: 1778356759
File name: NRF_Edumatic_Wirarama_genetic_mapreduce.docx (532.68K)
Word count: 3514
Character count: 21094

e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Vol. 5 No. 2, Desember, 2021, Hal. x-xx
DOI: 10.29408/edumatic.v5i2.XXX

Klasifikasi Teks menggunakan Genetic Programming dengan Implementasi Web Scraping dan Map Reduce

Wirarama Wedashwara[*1], Budi Irmawati[2], Andy Hidayat Jatmika[3], Ariyan Zubaidi[4]

[1,2,3,4]Program Studi Teknik Informatika, Universitas Mataram
email: wirarama@unram.ac.id*[1], budi-i@unram.ac.id[2], andy@unram.ac.id[3] dan zubaidi13@unram.ac.id[4]

(Received: xx xxx xxxx/ Accepted: xx xxx xxxx / Published Online: xx xxx xxxx)

Abstrak

Klasifikasi dokumen text pada media online menjadi permasalahan data besar dan memerlukan otomatisasi. Akurasi klasifikasi teks dapat menurun jika terdapat banyak term yang ambigu antar class. Hadoop Map Reduce merupakan framework pemrosesan parallel untuk data besar yang sudah banyak digunakan sebagai platform OLAP (Online Analytic Processing). Penelitian telah mengembangkan sistem klasifikasi teks dengan pre-processing menggunakan map-reduce dan pengumpulan data menggunakan web scraping. Penelitian bertujuan untuk melakukan evaluasi kinerja klasifikasi teks dengan menggabungkan algoritma genetik programming, map reduce dan web scraping untuk pemrosesan data besar berbentuk teks. Melalui web scraping telah dikumpulkan data dengan mengurangi duplikat sebanyak 17718. Map-reduce telah melakukan tokenisasi dan stop-word removal dengan total 36639 term dengan 5189 term unik dan 31450 term umum. Evaluasi ARM dengan jumlah data yang berbeda multi tree bisa menghasilkan rule lebih banyak dan panjang serta menghasilkan support yang lebih baik. Multi tree juga menghasilkan specific rule lebih banyak dan menghasilkan kinerja ARM lebih baik dari single tree. Evaluasi klasifikasi teks menunjukkan single tree menghasilkan akurasi lebih baik(0.7042) dari decision tree(0.6892) dan terendah adalah multi tree(0.6754).

Kata kunci: Genetic Programming, Klasifikasi teks, Map Reduce

Abstract

Classification of text documents on online media is a big data problem and requires automation. Klasifikasi teks accuracy can decrease if there are many ambiguous terms between classes. Hadoop Map Reduce is a parallel processing framework for big data that has been widely used as an OLAP (Online Analytic Processing) platform. Hadoop Map Reduce has also been widely used for text processing on big data. Research has developed a klasifikasi teks system with pre-processing using map-reduce and web scraping data collection. Through web scraping, data has been collected by reducing duplicates as much as 17718. Map-reduce has tokenized and stopped-word removal with 36639 terms with 5189 unique terms and 31450 common terms. Evaluation of ARM with different amounts of multi-tree data can produce more and longer rules and deliver better support. The multi-tree also has more specific regulations and better ARM performance than a single tree. Klasifikasi teks evaluation shows that a single tree produces better accuracy(0.7042) than a decision tree(0.6892), and the lowest is a multi-tree(0.6754).

Keywords: *Genetic Programming, Klasifikasi teks, Map Reduce*

PENDAHULUAN

Klasifikasi dokumen text pada media online menjadi permasalahan data besar dan memerlukan otomatisasi(Pintye et al., 2021). Akurasi klasifikasi teks dapat menurun jika terdapat banyak term yang ambigu antar class(Alt\inel & Ganiz, 2018). Melakukan pengelompokkan term untuk data yang besar memerlukan pemrosesan paralel(Du & Li, 2019). Hadoop Map Reduce merupakan framework pemrosesan parallel untuk data besar yang sudah banyak digunakan sebagai platform OLAP (Online Analytic Processing)(Jeong & Cha, 2019). Hadoop Map Reduce juga sudah banyak digunakan untuk pemrosesan text pada data besar(Ranjitha et al., 2020).

Penelitian mempresentasikan klasifikasi teks menggunakan genetic programming dengan melakukan pre-processing text menggunakan hadoop map reduce dan pengumpulan data menggunakan web scraping(Tahmassebi & Gandomi, 2018). Genetic programming digunakan karena melakukan association rule mining (ARM) sebelum klasifikasi teks sehingga mendapatkan analisis tambahan untuk pola data besar(Thomas & Mathur, 2019). Data yang digunakan artikel dari science-direct dengan kata kunci Internet of Things, Big Data dan Machine Learning(Telikani et al., 2020).

Penelitian bertujuan untuk melakukan klasifikasi teks dengan analisis pola data berbasis ARM. Penelitian juga bertujuan untuk melakukan evaluasi kinerja klasifikasi teks dengan menggabungkan algoritma genetik programming, map reduce dan web scraping untuk pemrosesan data besar berbentuk teks. Melalui ARM diharapkan dapat diketahui pola data antar label yang dapat berpengaruh pada perolehan akurasi. Penelitian juga bertujuan membentuk sistem dari pengumpulan data melalui web scraping, pre-processing menggunakan hadoop map-reduce dan klasifikasi teks menggunakan genetic programming.

Evaluasi diawali dengan pembahasan data yang telah dikumpulkan menggunakan web scraping dan map reduce hingga penjabaran tokenisasi kata(Ramsingh & Bhuvaneswari, 2018). Selanjutnya dilakukan perbandingan antara model single tree dan multi tree pada genetic programming. Terakhir dilakukan perbandingan hasil akurasi dengan algoritma decision tree yang dianggap memiliki kesamaan sifat.

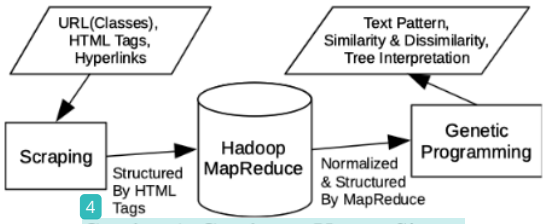
Penelitian terkait klasifikasi teks menggunakan algoritma berbasis tree telah dilakukan menggunakan decision tree sebagai feature selection(Deng et al., 2019), term weighting scheme untuk short-klasifikasi teks(Alsmadi & Hoon, 2019) dan klasifikasi teks dan clustering dari Twitter data untuk business analytics(Halibas et al., 2018). Ketiga penelitian tersebut menggunakan decision tree yaitu algoritma yang akan dibandingkan dengan genetic programming. Selain itu belum ada penelitian yang menggabungkan dengan pre-proccessing text menggunakan map-reduce.

Penelitian terkait pemanfaatan map reduce untuk pre-processing sudah dilakukan meninjau aspek algoritmic dari pemrosesan parallelnya(Koutris et al., 2018), Scalable Distributed Data Processing(Anjum, 2018), hingga Effective processing untuk unstructured data menggunakan python(Kousalya & Parvez, 2018). Penelitian yang diusulkan menggunakan bahasa pemrograman python dan parallel processing. Tetapi menggunakan jenis pre-processing dan algoritma yang berbeda. Penelitian melakukan klasifikasi teks secara umum dan tidak melakukan sentimen analisis seperti penelitian yang sudah ada(Sihombing et al., 2021).

Pemanfaatan genetic programming untuk keperluan pemrosesan text sudah pernah dilakukan Automated selection dan configuration dari multi-label grammar based(de Sá et al., 2018) dan feature selection pada highly dimensional skewed data(Viegas et al., 2018). Kedua penelitian

tidak melibatkan web scraping dan map reduce seperti pada penelitian ini. Penelitian ini juga menggunakan perbandingan single tree dan multi tree model dalam melakukan rule extraction.

METODE

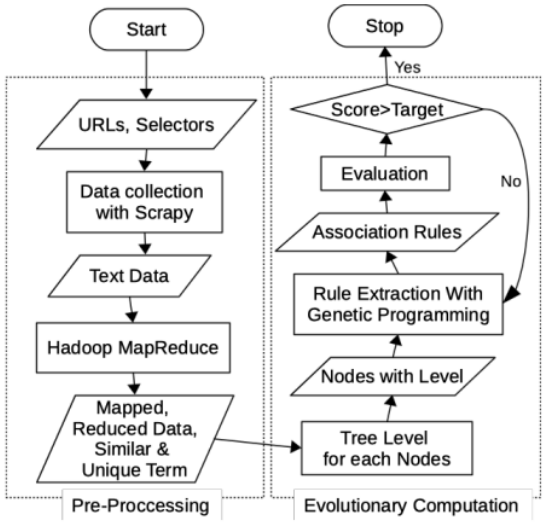


Gambar 1. Gambaran Umum Sistem

Gambar 1 menunjukkan gambaran umum sistem. Data dikumpulkan melalui proses web scraping menggunakan library scrapy pada bahasa pemrograman python. Proses web scraping dilakukan melalui mengekstrak teks dari tag html yang spesifik dari halaman HTML sumber yaitu science-direct. Label class dipisahkan berdasarkan kata kunci pencarian pada form pencarian science direct yaitu kata kunci Internet of Things, Big Data dan Machine Learning.

Penyimpanan dilakukan pada hadoop untuk dilakukan proses map reduce. Proses map reduce memungkinkan pemrosesan secara paralel sehingga cocok untuk pemrosesan data yang banyak. Proses map dilakukan untuk memisahkan kata pada artikel yang di kumpulkan. Proses stopword removal juga dilakukan pada proses map. Pada proses reduce dilakukan proses tokenization yaitu menghitung jumlah kemunculan kata. Perhitungan tokenization pada reduce dilakukan dengan memisah kemunculan kata yang hanya terjadi di dalam satu label maupun muncul secara umum.

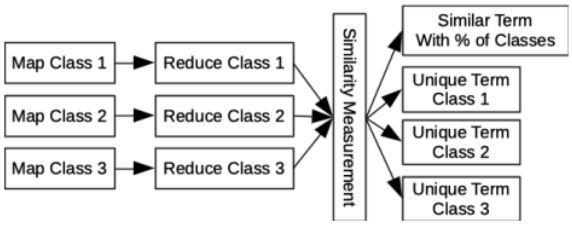
Genetic programming digunakan untuk melakukan ektraksi text pattern, perhitungan similarity dan dissimilarity, tree interpretation hingga tujuan utama penelitian yaitu klasifikasi teks. Data yang diproses oleh genetic programming adalah data yang sudah dalam bentuk objek yang dibuat melalui proses map reduce pada hadoop.



Gambar 2. Diagram Alir Sistem

Gambar 2 menunjukkan flowchart dari sistem. Proses terbagi menjadi pre-proccessing dan evolutionary computation. Proses pre-proccessing terdiri dari input URL (science direct) dan tag yang akan di download oleh scrapy. Text data diproses oleh map reduce hingga menjadi objek yang siap diproses oleh genetic programming.

Genetic programming melakukan tingkatan kata berdasarkan frekuensi kemunculannya. Genetic programming akan melakukan ekstraksi rule dengan memprioritaskan kata dengan frekuensi kemunculan tinggi ke rendah. Rule yang diekstraksi akan digunakan untuk melakukan klasifikasi hingga mencapai target akurasi yang diharapkan.



Gambar 3. Struktur Map Reduce

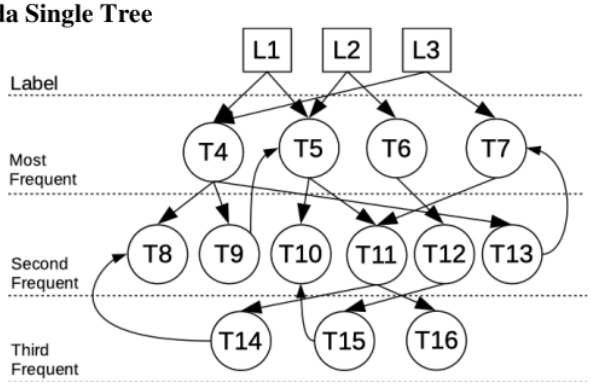
Gambar 3 menunjukkan proses map reduce pada sistem. Proses map dilakukan berdasarkan data per class dengan memisahkan term (kata) dari artikel yang di download. Proses reduce juga dilakukan berdasarkan masing-masing class dengan melakukan tokenization untuk term yang sudah di map sebelumnya. Proses akhir dilakukan dengan memisahkan kata-kata yang hanya muncul di class masing-masing sebagai Unik term dan similar term. Unik term akan digunakan untuk membentuk specific rules, sedangkan similar term digunakan untuk membentuk common rule pada genetic programming.

HASIL DAN PEMBAHASAN

Hasil

Pada sub-bab ini dijelaskan struktur gen rule extractor dari genetic programming model single tree dan multi tree yang digunakan sebagai rule based classifier pada penelitian ini. Pembahasan mencakup struktur gen dalam tampilan grafik tree, struktur objek dalam pemrograman dan contoh rules yang dihasilkan oleh masing-masing rule extractor. Rule yang dihasilkan oleh rule extractor hanya ditampilkan sebagian karena keterbatasan halaman.

Struktur Gen pada Single Tree



Gambar 4. Struktur Gen pada Single Tree

Gambar struktur gen genetic programming dengan model single tree ditunjukkan oleh gambar 4. Pada struktur single tree seluruh node tergabung dalam satu pohon dengan root lebih dari satu yang berisikan label atau kata kunci dalam pencarian. Node dengan bentuk persegi berisi label. Node dengan bentuk lingkaran berisi term yaitu kata dan frekuensi kemunculannya. Tingkatan pohon terbagi menjadi 4 bagian yaitu label dan 3 tingkatan frekuensi kemunculan kata yang direpresentasikan oleh masing-masing node.

Tabel 1. Struktur Gen pada Single Tree

i	NT_i	Lv_i	C_i	T_i
1	L	0	4,5	IoT
2	L	0	5,6	BD
3	L	0	4,7	ML
4	T	1	8,9,13	Internet
5	T	1	10,11	Algorithm
6	T	1	12	Data
7	T	1	11	Iteration
8	T	2	-	Nodes
9	T	2	5	Sensor
10	T	2	-	Storage
11	T	2	14,16	Cloud
12	T	2	15	Network
13	T	2	7	Training
14	T	3	8	Information
15	T	3	10	Classification
16	T	3	-	Clustering

Struktur tree pada single tree merupakan directed graph yang memungkinkan multiple direction dari satu node ke node lainnya. Direction juga tidak selalu dari atas kebawah tetapi juga memungkinkan naik ke node yang ada pada tingkatan diatasnya. Sehingga walaupun single tree memungkinkan ekstaksi rule yang sangat bervariatif.

Tabel 1 menunjukkan struktur gen dari genetic programming dengan rule extractor single-tree pada gambar 4. Kolom pertama i menunjukkan indeks dari node 1 hingga 16. Kolom NT_i menunjukkan node type dari masing-masing node. Tipe L menunjukkan label yaitu Internet of Things (IoT), Big Data (BD) dan Machine Learning (ML). Sedangkan tipe T menunjukkan term yaitu kata yang muncul dalam artikel yang sudah dikumpulkan melalui proses web scraping.

Kolom Lv_i menunjukkan tingkatan pada struktur pohon yaitu 0 untuk root, 1 untuk most frequent, 2 untuk second dan 3 untuk third most frequent. Kolom C_i menunjukkan koneksi antar node. Untuk multiple connection direpresentasikan dengan array pada pemrograman. T_i menunjukkan *term* yaitu kata yang direpresentasikan oleh masing-masing node.

Tabel 2 Rule yang diektraksi oleh Single Tree

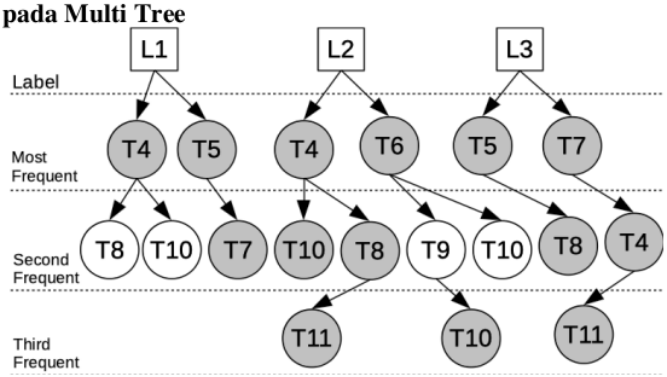
Rules	Length	Conf	Support	Score
$L1 \rightarrow T4 \wedge T13$	2	0.456	0.398	2.626
$L1 \rightarrow T4 \wedge T13 \wedge T7$	3	0.348	0.512	3.686
$L1 \rightarrow T4 \wedge T13 \wedge T7 \wedge T11$	4	0.321	0.234	4.394
$L1 \rightarrow T4 \wedge T13 \wedge T7 \wedge T11 \wedge T14$	5	0.234	0.102	5.219
$L1 \rightarrow T4 \wedge T13 \wedge T7 \wedge T11 \wedge T14 \wedge T8$	6	0.012	0.002	6.008

Rata-rata	4.386

Tabel 2 menunjukkan rule yang diekstraksi oleh single tree. Tanda panah pada struktur rules pada kolom pertama menunjukkan pemisah antara precedent dan dependent. Precedent ditempatkan oleh label dan dependent menunjukkan frequent item set. Length menunjukkan jumlah node pada dependent. Confidence dan support menunjukkan hasil evaluasi association rule mining yang dapat dilihat pada penelitian sebelumnya. Score berisi gabungan length, confidence dan support yang ditunjukkan pada rumus 1.

Hanya rule oleh L1 saja yang ditunjukkan oleh tabel 2 karena keterbatasan halaman. Rule yang diekstraksi bisa lebih banyak yang berasal dari L2 dan L3. Rule yang di ekstraksi berupa incremental dan tidak selalu harus berakhir hingga paling bawah. Semakin pendek rule bisa menghasilkan support yang lebih tinggi karena syarat yang lebih sedikit. Tetapi manjadi kurang kuat untuk digunakan pada pemrosesan klasifikasi atau regresi. Sehingga dipriotitaskan pada panjang rule untuk menghasilkan syarat yang lebih spesifik untuk menentukan label.

Struktur Gen pada Multi Tree



Gambar 5 Struktur Gen Multiple Tree

Struktur multi tree pada genetic programming ditunjukkan oleh gambar 5. Berbeda dengan single tree antar label tidak ada node yang terhubung. Sehingga terdapat term yang duplikat antar tree seperti T4 yang terdapat pada setiap tree. Berbeda dengan struktur single tree yang memungkinkan balik ke node diatasnya, dalam satu tree juga terdapat term yang sama seperti T10 yang terdapat 3 duplikat pada tree L2. Pada representasi grafik multi tree tampak lebih sederhana tetapi lebih rumit pada struktur objek yang akan dibawas selanjutnya.

Tabel 3 Struktur Gen Multi Tree

i	NT_i	L_i	Lv_i	C_i	V_i
1	L		0	[4,5]	IoT
2	L		0	[4,6]	BD
3	L		0	[5,7]	ML
4	T	1,2,3	1,1,2	[8,10],[10,8],[11]	Internet
5	T	1,3	1	[7],[7,8]	Algorithm
6	T	2	1	[9,10]	Data
7	T	1,3	1,2	[],[14]	Iteration
8	T	1,2,3	2,2,2	[],[11],[]	Nodes
9	T	2	2	[10]	Network
10	T	1,2,2,2	2,2,2,3	[],[],[],[]	Storage
11	T	2,3	3,3	[],[]	Cloud

Struktur objek pada multi tree ditunjukkan pada tabel 3. Kolom i, NT_i, Lv_i, C_i dan V_i memiliki fungsi yang sama dengan single tree. Kolom L_i memiliki fungsi untuk menunjukkan kepemilikan oleh label. Jumlah L_i dan Lv_i selalu sama untuk menunjukkan tingkatan node dalam masing-masing tree. Karena tidak adanya direksi untuk balik ke node diatasnya sehingga Lv_i juga bisa di duplikasi pada satu tree yang sama seperti T10.

Connection Ci ditunjukkan dengan array tambahan. Jumlah array selalu sama dengan jumlah Li dan Lvi. Jika tidak ada koneksi selanjutnya akan berisi array kosong. Struktur objek ini memungkinkan representasi node tanpa melakukan duplikasi pada anggota array.

Tabel 4 Rule yang diekstraksi Multi Tree

Rules	Length	Conf	Support	Score
$L1 \rightarrow T4 \wedge T8$	2	0.357	0.349	2.5275
$L1 \rightarrow T4 \wedge T10$	2	0.291	0.548	2.6935
$L1 \rightarrow T5 \wedge T7$	2	0.348	0.346	2.52
$L2 \rightarrow T4 \wedge T10$	2	0.267	0.647	2.7805
$L2 \rightarrow T4 \wedge T10$	2	0.479	0.178	2.4175
$L2 \rightarrow T4 \wedge T8$	2	0.678	0.789	3.128
$L2 \rightarrow T4 \wedge T8 \wedge T10$	3	0.567	0.658	3.9415
Rata-rata				2.858

Tabel 4 menunjukkan contoh rule yang diekstrak oleh struktur multi tree. Contoh menunjukkan sebagian rule yang diekstraksi oleh L1 dan L2. L1 menunjukkan maksimal dua kombinasi dependent karena hanya terdapat dua tingkat. Sedangkan L2 bisa mencapai tiga kombinasi karena memiliki tiga tingkatan. Perbedaan hasil dengan single tree secara spesifik akan dibahas pada sub bab selanjutnya.

Pembahasan

Hasil evaluasi diawali dengan pembahasan data yang telah dikumpulkan menggunakan web scraping dan map reduce hingga penjabaran tokenisasi kata. Selanjutnya dilakukan perbandingan antara model single tree dan multi tree pada genetic programming. Terakhir dilakukan perbandingan hasil akurasi dengan algoritma decision tree yang dianggap memiliki kesamaan sifat.

Data hasil web scraping

Tabel 4 Data yang telah dikumpulkan melalui web scraping

	IoT	BD	ML	Total
Collected	6000	6000	6000	18000
Used	5923	5873	5922	17718
Duplikasi	57	127	78	262
IoT		56	35	91
BD	17		43	60
ML	40	71		111

Tabel 4 menunjukkan data yang sudah dikumpulkan melalui proses web scraping. Data dikumpulkan dari artikel terbaru per 1 desember 2021 hingga batas 6000 judul untuk masing-masing kata kunci IoT, Big Data (BD) dan Machine Learning (ML). Header horizontal menunjukkan masing-masing kata kunci yang scraping dan totalnya. Masing-masing kata kunci dikoleksi sebanyak 6000 judul dan abstrak dan total berjumlah 18000.

Header vertikal menunjukkan keterangan jumlah yang dikumpulkan, dipakai, total duplikat yang sama pada setiap kata kunci dan relasi yang menunjukkan dengan kata kunci apa terjadi

duplikat. Sebagai contoh label IoT memiliki total 57 duplikat yaitu 17 dengan BD dan 40 dengan ML. Duplikat terbanyak dimiliki oleh BD sebanyak 127 yaitu 56 dengan IoT dan 71 dengan ML. Dengan mengurangi total 262 artikel maka total data yang dipakai adalah 17718. Melalui pengumpulan duplikat ini bisa diperkirakan false positive yang akan muncul antar label karena persamaannya pada hasil pencarian.

Hasil Map Reduce

Tabel 5 Hasil pengumpulan kata menggunakan mapreduce

	IoT	BD	ML	Total
Map	66838	70327	72803	209968
Reduce	11867	12716	12056	36639
Unik	2189	1736	1264	5189
Duplikasi	9678	10980	10792	31450
IoT		7897	5283	13180
BD	6689		5509	12198
ML	2989	3083		6072

Tabel 5 menunjukkan kata yang diekstraksi dari artikel yang telah dikumpulkan sebelumnya melalui proses web scraping. Pada proses map dilakukan pemisahan setiap kata pada artikel. Jumlah kata yang ditunjukkan oleh tabel adalah kata-kata spesifik yang telah melalui proses stopword removal sebelumnya. Pada proses map diekstrak 209968 kata dari seluruh kata kunci.

Pada proses reduce dilakukan proses tokenizer dengan menghitung kata yang sama untuk mendapatkan term frequency nya. Selanjutnya dihitung kata yang hanya muncul pada masing-masing label (Unik) dan juga terdapat di pada kata kunci lain (Duplikasi) atau yang juga disebut inverse document frequency. Tiga baris bawah menunjukkan kesamaan kata kata kunci dengan yang lainnya. Misalnya kata kunci IoT memiliki total 9678 kesamaan kata yaitu 6689 dengan BD dan 2989 dengan ML.

Total kata yang sama berjumlah 31450 dan hanya 5189 kata unik yang akan digunakan yang akan digunakan untuk membuat pohon rule extractor pada genetic programming. Kata unik terbanyak dimiliki oleh IoT yaitu 2189 diikuti BD yaitu 1736 dan ML yaitu 1264. Melalui tabel ini bisa dianalisis kerumitan pohon dari masing-masing kata kunci.

Perbandingan akurasi text classiofication dengan decision tree

Tabel 6 Perbandingan akurasi klasifikasi teks dengan decision tree

	GP Single Tree		GP Multi Tree		Decision Tree	
Data	**support**	**Akurasi**	**support**	**Akurasi**	**support**	**Akurasi**
3544	0.189	0.598	0.278	0.546	0.234	0.679
7088	0.276	0.678	0.289	0.637	0.323	0.658
10631	0.307	0.708	0.349	0.649	0.324	0.618
14175	0.439	0.739	0.512	0.759	0.445	0.779
17718	0.569	0.798	0.524	0.786	0.468	0.712
Rata-rata	0.356	0.7042	0.3904	0.6754	0.3588	0.6892

Tabel 9 menunjukkan perbandingan akurasi klasifikasi teks antara genetic programming dengan decision tree. Evaluasi mencakup support dari rules yang dihasilkan dan akurasi hasil klasifikasi teks. Evaluasi dilakukan dengan data mulai dari 3544 hingga seluruh data yaitu 17716. Data uji dilakukan menggunakan 3000 data yang di scraping secara terpisah dengan masing-masing 1000 data per kata kunci.

Untuk data yang paling sedikit yaitu 3544, decision tree memiliki hasil akurasi yang paling tinggi yaitu 0.679. Hasil ini memnunjukkan decision tree memiliki akurasi yang lebih baik dengan data training yang lebih sedikit. Single tree memiliki hasil akurasi yang lebih tinggi dibanding decision tree sejak jumlah data 7088. Multi tree hanya menghasilkan akurasi lebih baik dari decision tree pada jumlah data 10631 dan 17718 saja. Secara rata-rata genetic programming dengan single tree menghasilkan akurasi tertinggi yaitu 0.7042 diikuti decision tree dengan 0.6892 dan paling kecil oleh multi tree dengan 0.6754.

Untuk perolehan nilai support genetic programming dengan single tree menghasilkan rata-rata support tertinggi yaitu 0.3904 diikuti decision tree dengan 0.3588 dan paling kecil single tree dengan 0.356. Multi tree memiliki hasil support paling tinggi di semua jumlah data training. Sedangkan single tree memiliki support terendah untuk jumlah data 3544 hingga 14175. Secara umum jumlah support tidak sejalan dengan nilai akurasi yang dicapai.

SIMPULAN

Penelitian telah mengembangkan sistem klasifikasi teks dengan pre-processing menggunakan map-reduce dan pengumpulan data menggunakan web scraping. Melalui web scraping telah dikumpulkan data dengan mengurangi duplikat sebanyak 17718. Map-reduce telah melakukan tokenisasi dan stop-word removal dengan total 36639 term dengan 5189 term unik dan 31450 term umum. Evaluasi ARM dengan jumlah data yang berbeda multi tree bisa menghasilkan rule lebih banyak dan panjang serta menghasilkan support yang lebih baik. Multi tree juga menghasilkan specific rule lebih banyak dan menghasilkan kinerja ARM lebih baik dari single tree. Evaluasi klasifikasi teks menunjukkan single tree menghasilkan akurasi lebih baik(0.7042) dari decision tree(0.6892) dan terendah adalah multi tree(0.6754).

REFERENSI

Alsmadi, I., & Hoon, G. K. (2019). Term weighting scheme for short-text classification: Twitter corpuses. *Neural Computing and Applications*, *31*(8), 3819–3831.

Alt\inel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing \& Management*, *54*(6), 1129–1153.

Anjum, B. (2018). MapReduce--The Scalable Distributed Data Processing Solution. In *Topics in Parallel and Distributed Computing* (pp. 173–190). Springer.

de Sá, A. G. C., Freitas, A. A., & Pappa, G. L. (2018). Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. *International Conference on Parallel Problem Solving from Nature*, 308–320.

Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. *Multimedia Tools \& Applications*, *78*(3).

Du, S., & Li, J. (2019). Parallel processing of improved KNN text classification algorithm based on Hadoop. *2019 7th International Conference on Information, Communication and Networks (ICICN)*, 167–170.

Halibas, A. S., Shaffi, A. S., & Mohamed, M. A. K. V. (2018). Application of text classification and clustering of Twitter data for business analytics. *2018 Majan International Conference (MIC)*, 1–7.

Jeong, H., & Cha, K. J. (2019). An efficient mapreduce-based parallel processing framework for user-based collaborative filtering. *Symmetry*, *11*(6), 748.

Kousalya, K., & Parvez, S. J. (2018). Effective processing of unstructured data using python in Hadoop map reduce. *International Journal of Engineering \& Technology*, *7*(2.21),

417–419.

Koutris, P., Salihoglu, S., Suciu, D., & others. (2018). Algorithmic aspects of parallel data processing. *Foundations and Trends® in Databases*, *8*(4), 239–370.

Pintye, I., Kail, E., Kacsuk, P., & Lovas, R. (2021). Big data and machine learning framework for clouds and its usage for text classification. *Concurrency and Computation: Practice and Experience*, *33*(19), e6164.

Ramsingh, J., & Bhuvaneswari, V. (2018). An efficient Map Reduce-Based Hybrid NBC-TFIDF algorithm to mine the public sentiment on diabetes mellitus--A big data approach. *Journal of King Saud University-Computer and Information Sciences*.

Ranjitha, K. V, Prasad, B. S. V., & others. (2020). Optimization Scheme for Text Classification Using Machine Learning Na{\"\i}ve Bayes Classifier. In *ICDSMLA 2019* (pp. 576–586). Springer.

Sihombing, L. O., Hannie, H., & Dermawan, B. A. (2021). Sentimen Analisis Customer Review Produk Shopee Indonesia Menggunakan Algortima Na{\"\i}ve Bayes Classifier. *Edumatic: Jurnal Pendidikan Informatika*, *5*(2), 233–242.

Tahmassebi, A., & Gandomi, A. H. (2018). Genetic programming based on error decomposition: A big data approach. In *Genetic programming theory and practice XV* (pp. 135–147). Springer.

Telikani, A., Gandomi, A. H., & Shahbahrami, A. (2020). A survey of evolutionary computation for association rule mining. *Information Sciences*, *524*, 318–352.

Thomas, D. M., & Mathur, S. (2019). Data analysis by web scraping using python. *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 450–454.

Viegas, F., Rocha, L., Gonçalves, M., Mourão, F., Sá, G., Salles, T., Andrade, G., & Sandin, I. (2018). A genetic programming approach for feature selection in highly dimensional skewed data. *Neurocomputing*, *273*, 554–569.

edumatic

ORIGINALITY REPORT

5%	5%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Jenderal Soedirman Student Paper	2%
2	e-journal.hamzanwadi.ac.id Internet Source	2%
3	www.researchgate.net Internet Source	<1%
4	jtrm.polman-bandung.ac.id Internet Source	<1%
5	www.mdpi.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		