Table of Contents

Sr. No	Topic
1.	Scope of the Journal
2.	The Model
3.	The Advisory and Editorial Board
4.	Papers

First Published in the United States of America.

Originated and printed by Foundation of Computer Science Press, New York, USA

Scope of the Journal

International Journal of Computer Applications (IJCA) creates a place for publication of papers which covers the frontier issues in Computer Science and Engineering and their applications which will define new wave of breakthroughs. The journal is an initiative to identify the efforts of the scientific community worldwide towards inventing new-age technologies. Our mission, as part of the research community is to bring the highest quality research to the widest possible audience. International Journal of Computer Applications is a global effort to consolidate dispersed knowledge and aggregate them in a search-able and index-able form.

The perspectives presented in the journal range from big picture analysis which address global and universal concerns, to detailed case studies which speak of localized applications of the principles and practices of computational algorithms. The journal is relevant for academics in computer science, applied sciences, the professions and education, research students, public administrators in local and state government, representatives of the private sector, trainers and industry consultants.

Indexing

International Journal of Computer Applications (IJCA) maintains high quality indexing services such as Google Scholar, CiteSeer, UlrichsWeb, DOAJ (Directory of Open Access Journals) and Scientific Commons Index, University of St. Gallen, Switzerland. The articles are also indexed with SAO/NASA ADS Physics Abstract Service supported by Harvard University and NASA, Informatics and ProQuest CSA Technology Research Database. IJCA is constantly in progress towards expanding its contents worldwide for the betterment of the scientific, research and academic communities.

Topics

International Journal of Computer Applications (IJCA) supports a wide range of topics dealing in computer science applications as: Embedded Systems, Pattern Recognition, Signal Processing, Robotics and Micro-Robotics, Theoretical Informatics, Quantum Computing, Software Testing, Computer Vision, Digital Systems, Pervasive Computing etc.

The Model

Open Review

International Journal of Computer Applications approach to peer review is open and inclusive, at the same time as it is based on the most rigorous and merit-based 'blind' peer-review processes. Our referee processes are criterion-referenced and referees selected on the basis of subject matter and disciplinary expertise. Ranking is based on clearly articulated criteria. The result is a refereeing process that is scrupulously fair in its assessments at the same time as offering a carefully structured and constructive contribution to the shape of the published paper.

Intellectual Excellence

The result is a publishing process which is without prejudice to institutional affiliation, stage in career, national origins or disciplinary perspective. If the paper is excellent, and has been systematically and independently assessed as such, it will be published. This is why International Journal of Computer Applications has so much exciting new material, much of it originating from well known research institutions but also a considerable amount of brilliantly insightful and innovative material from academics in lesser known institutions in the developing world, emerging researchers, people working in hard-to-classify interdisciplinary spaces and researchers in liberal arts colleges and teaching universities.

The Advisory and Editorial Board

The current editorial and Advisory committee of the International Journal of Computer Applications (IJCA) includes members of research center heads, faculty deans, department heads, professors, research scientists, experienced software development directors and engineers.

PAPERS

Naming Conflict Resolution in Multidatabase Systems using Bridge Table *Authors : Samia M. Masaad, Rania A. Tabeidi, Omer I. Eldai*	1-6
Novel Sentiment Analysis using Twitter *Authors : Athare Sharayu Shivaji, Rathod Vijay Uttam*	7-9
Enhancement Effects of the STATCOM on the Distance Relay Protection *Authors : Ahmed Nasser Alsammak, Sarmad Arkan Janderma*	10-14
Multiprocessing Stemming: A Case Study of Indonesian Stemming *Authors : Novi Yusliani, Rifkie Primartha, Mastura Diana Marieska*	15-19
Real-Time DC Servomotor Identification and Control of Mechanical Braking System for Vehicle to Vehicle Communication *Authors : Mohammad A. Obeidat*	20-30
A Study on Cloud Computing Quality of Service for Big Data: Contribution to Computation *Authors : Gibet Tani Hicham, El Amrani Chaker*	31-38
A New Robust Methodology of Image Compression based on CDF Wavelets *Authors : Pradeep Kumar Pal, Kapil Dev Sharma*	39-44
Fabrication and Study of Memory Cell Switching Properties based on Ag2S Compound *Authors : Azam Adnan Mohammed Al-Kubaa, Khalid Khaleel Mohammad*	45-48

Multiprocessing Stemming: A Case Study of Indonesian Stemming

Novi Yusliani
Artificial Intelligence Laboratory
Comp. Science Department
Universitas Sriwijaya

Rifkie Primartha
Artificial Intelligence Laboratory
Comp. Science Department
Universitas Sriwijaya

Mastura Diana Marieska
Artificial Intelligence Laboratory
Comp. Science Department
Universitas Sriwijaya

ABSTRACT

Research in the field of Natural Language Processing (NLP) is currently increasing especially with the arrival of a new term that is “big data”. The needs of the programming library that ready-touse becomes very important to speed up the phases of research. Some libraries that have already been mature is available but generally for English language and its dependently. So, it can’t be used for other languages. Stemming is one of the basic processes that exist in NLP. Indonesian stemming algorithm that often used is ECS (Enhanced Confix-Stripping). One of the libraries that already implemented the algorithm is Sastrawi[1]. Results from the experiment show that the time of stemming processing by Sastrawi is still slow. Therefore, this research will optimize the speed of stemming processing using multiprocessing (MP). The data test are used in this research has manually taken from Wikipedia[2]. The experiment results show that the MP technique can decrease the average time of stemming processing about 98.45%.

General Terms

Natural Language Processing, Enhanced Confix-Stripping

Keywords

Multiprocessing, Stemming, ECS, Sastrawi

1. INTRODUCTION

Natural Language Processing (NLP) is a sub-field of Artificial Intelligence (AI) that learn how to develop a computer program in order to understand the human language . Many applications are NLP-based, such as Question Answering System (QAS), a system that able to answer the questions automatically. Text Summarization (TS), a system for summarizing a lengthy text. Machine Translator (MT), the system for translating languages. Today, research in this topics area are still done with a variety of additional new techniques to improve performance system [1][2][3]. One process that is often used at the stage of pre-processing in NLP is stemming that aims to find the root of a word with eliminating all forms of affixes. For example, the word “running” will be “run” after being processed by stemming. One of the stemming benefits is able to reduce the sentence [4] so that it will increase the speed of computing than without stemming.

Stemming usually dedicated to specific language. This is due to the grammatical differences of each language. For example, stemming algorithms in English language, however would be not maximum or even not suitable if it used in another languages. Research in Indonesia stemming has already explored by some scientist. Those of them do a pretty popular are Adriani [7], Arifin [8], dan Tahitoe [9]. This research group have linkages because the post research were improved the prior research. The implementation of Indonesian stemming algorithms are also easy to be obtained, the popular one is from Sastrawi.

Stemming algorithm from Sastrawi has provided sufficient good result in the side of effectiveness. But the problems arises in the efficiency of processing time. Therefore, we proposed multiprocessing (MP) technique to solve the efficiency problem. MP is a technique in computer programming which aims to make processes run in parallel or simultaneously. This technique is allowed to create computer programs doing works just in time so it can accelerate the computing. Furthermore, this article is organized into several sections as follows, Section 2 describe the research associated with Indonesian stemming. Section 3 describe the methodology is used in this study. Section 4 contains discussion and experimental results. The last Section are conclusions and future work.

2. RELATED WORK

Stemming is the process to looking for root of a word. Stemming is often found on the NLP-based systems because of its ability to reduce sentences. For example, there are three words as inputs: “doing”, “does”, and “did”. Then stemming will seek root of a word from the third words and give the result word “do”. Generally the techniques used in stemming is to remove all forms of affix, prefix or suffix. Based on track record of research in Indonesian stemming, the stemming research began back in 2005 were pioneered by the Asian [6]. Previously, stemming is only became part of the complementary fields of study and used for a specific task, such as performed by Vega in [7] that uses stemming for Information Retrieval (IR). Furthermore, Arifin and Setiono in [7] uses stemming for document classification, and [5] that examines the effect of stemming against IR.

Work from [6] began to raise up more research about stemming. They modified Nazief and Adriani algorithms, one of standard stemming algorithm in Indonesia. They can improve stemming performance from 93.0% to 95.0%. Further research [7] began to change the name of the algorithm they develop that was originally named is Nazief and Adriani algorithm became Confix-Stripping (CS) algorithm. The name is choosen based on their understanding from the stemming rules that apply in the Indonesia language. CS algorithm provides 34 rules and it gain accuracy of 97.0% with a dataset contains 3,986 not unique words. Also, research from [6] make improvements CS algorithm and it is called Enhanced Confix-Stripping (ECS). They do modification against some rules from the previous CS. ECS approach could fix some mistakes done by CS and give file size reduction of 30.95% to 32.66%. Lastly, research conducted by [9] try to fix some of mistakes stemming conducted by ECS. Especially for

[1] https://github.com/sastrawi
[2] https://www.wikipedia.org

the two main issues, namely: 1) Overstemming that is the process of stemming that too many cutting of word so that it lead to wrong result, for example the word "penyidik" could be "sidi"; and 2) Understemming that is the process of stemming that allows to produce some words with equal meaning, for example, the word "mengalami" could be "alami" and "alam". They used Corpus-based Stemming (CBS) method to fix the errors. The basic idea of CBS is similar to the text representation using n-gram, precisely 2-gram because it uses two pairs of words. The meaning of a word can be influenced by the nearest word, hence this technique can choose the stemming results that match based on a corpus.

The results from previous studies have shown that Indonesian stemming algorithms is pretty good, above 95%. So that in the side of effectiveness, the algorithm is worthy to used. But as far as we know, there is no research that discuss in the side of efficiency, such as stemming processing time. Generally the researchers focusing on the modification and improvement of stemming quality. In fact, the response time is very important to the systems that require fast output.

Multiprocessing (MP) is technique in computer programming aims to make the process run in parallel way or simultaneously. The technique allows computer programs doing a lot of works in one time so that it can be accelerate the computing process. Research in this area generally focus on process efficiency to get fast response time so that the system is ready to be used in real problems. Researches related to MP are still active until now include by [10] compares the performance of MP with single process (SP) technique for scheduling algorithm. The result is technique of MP gave more good responds than SP. Next [11] uses MP to reduce time on simulation interaction application in the field of cell biology. They did an experiment MP with three levels, namely: multiple-threads, parallelization (processor core-level), and multiple-computer architecture. The results of the experiments showed that parallelization techniques provides quite nice result. The last study done by [12] is comparative studies for application-based Information Computer Technology (ICT) that uses MP technique. Generally, the procurement of ICT devices will accompanied by high hardware specs to smooth activities without regard to the influence of the software used. Whereas, if the selection of the software considered then it is possible to reduce the cost of procurement. Therefore, the role of software use the technique of MP is expected to add to the performance of ICT devices

Based on reviews some research that related to the MP technique, it can be inferred that this technique can improve system performance. Especially for systems that requires best response time. So that, this technique is also very possible to applied in stemming to solve the slow processing time.

3. RESEARCH METHODOLOGY

3.1 System Architecture

Figure 1 shown multiprocessing stemming architecture that proposed in this research. We can see that the file.txt (yellow color) will be divided into few text-blocks (green color). Each text-blocks will be processed by different processor (pink color). After the process is completed, the results will be merged and saved back into the file.text.

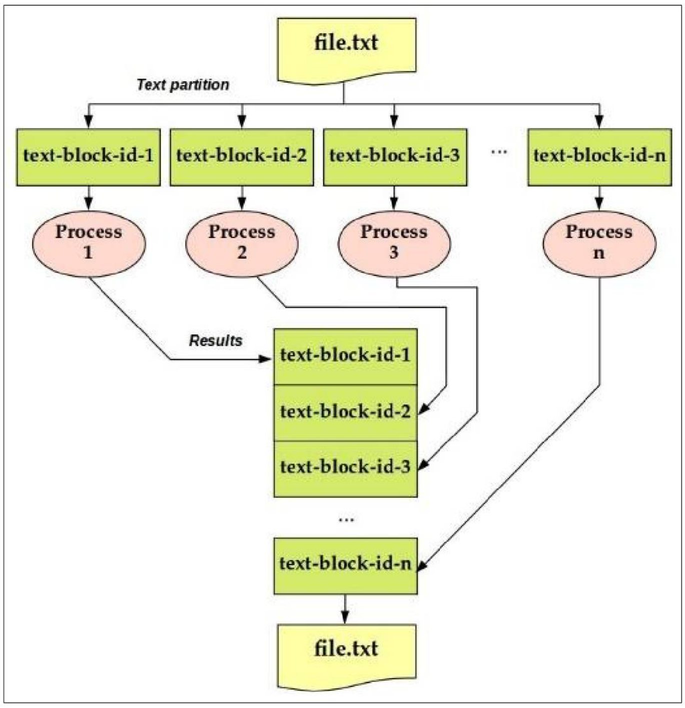

Fig 1: Multiprocessing Stemming Architecture

3.2 Data Test

Data test type used in this research is plain text. The data manually retrieved from Wikipedia site. It is divided into seven part as follows: wiki-1.txt file (1.9 MB), wiki-2. txt (2.5 MB), wiki-3 .txt file (3.5 MB), wiki-4. txt (4.4 MB), wiki-5. txt (4.8 MB), wiki-6. txt (5.3 MB), and wiki-7. txt (5.7 MB). The size of data test which used in this study appartently smaller (under 10 MB). But, the processing time is influenced not only by the size of data set but also by text processing. Therefore, the hardware and software specification will affect the processing time.

3.3 Tools Specification

The specifications of hardware and software used in this experiments as follows:

- Intel i5 3320 (4 cores), RAM 8 GB, 250 SSD.
- Linux (Ubuntu 16.04.5 LTS) Operating System, Python Programming Language.
- Sastrawi library which contains implemented ECS (Enhanced Confix Stripping) algorithm, and for naming convention we called it **ECS-Sastrawi**.
- Self-implemented ECS algorithm based on [6], we called it **ECS-Dev**.

3.4 Measurment Method

This study aims to measure stemming processing time using **Single Process** (SP) and **Multiprocess** (MP) techniques in similar data test and tools specification. There are three comparisons for each ECS-Sastrawi and ECS-Dev, namely: (1) ECS-Sastrawi comparison using SP and MP techniques; (2) ECS-Dev comparison using SP and MP techniques; and (3) The comparison results of ECS-Sastrawi and ECS-Dev. The measurement results used time/second. The smaller processing time, the better performance.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Sastrawi's Stemming

Table 1 contains the results of stemming experiments using SP and MP techniques with ECS-Sastrawi. The table consists of five columns, namely: test data name, file size, time of processing stemming using SP, time of processing stemming using MP, and the percentage of time reduction from SP to MP that calculated using Eq. (1).

Table 1. The experimental results of ECS-Sastrawi using SP and MP techniques.

Data test	Size (MB)	SP Time (Sec.)	MP Time (Sec.)	Reduction Time (%)
Wiki-1	1.9	4998	2972	40.54
Wiki-2	2.5	6664	4002	39.95
Wiki-3	3.5	9163	5612	38.75
Wiki-4	4.4	11662	7195	38.30
Wiki-5	4.8	127772	7939	37.86
Wiki-6	5.3	14161	8901	37.14
Wiki-7	5.7	14994	9485	36.74

The experimental results show that the time of processing stemming by ECS-Sastrawi is still taking long time. For example, 1.9MB text file (wiki-1.txt) takes amount of 4.998 seconds (1 hour, 23 minutes, 18 seconds) running on SP. After running on MP, it occurs a significant reduction time to 2.972 seconds (49 minutes, 32 seconds) or 40.54%. Nevertheless, the percentage of reduction time is still below average processing time. This is shown by the graph in Figure 2. It can be seen that the blue line (MP) is above the black line (mean). It is means that the MP processing time is still longer than the average processing time. In addition, even though the percentage of ECS-Sastrawi using MP is reducing significant time but the results is not acceptable because 1.9 MB text file in 49 minutes is still long in real environment. Especially, in big data where their smallest size data generally in hundreds of Mega Bytes (MB).

$$reduction\ time = 100 - \left(\frac{MP\ Time * 100}{SP\ Time}\right) \quad (1)$$

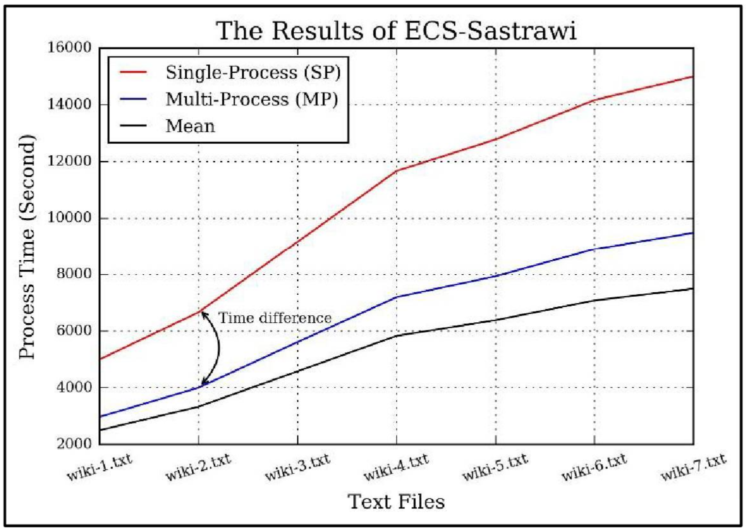

Fig 2: ECS-Sastrawi with time difference and mean between SP and MP techniques.

4.2 ECS's Steaming

Table 2 contains the results of stemming experiments using SP and MP techniques with ECS-Dev. Experiment results indicates that the difference in processing time between the ECS-SP Sastrawi and ECS-Dev quite significantly from 4.998 seconds (1 hour, 23 minutes, 18 seconds) to 111 seconds (1 minute 51 seconds). If it made into percentage then ECS-Dev using SP was able to reduce the time of 97.78%. Such a big difference time. It happens because code complexity. At first, Sastrawi is designed specifically for PHP programming. So there is no optimization for another programming languages. Therefore, the goals of this research is to prove the need optimization stemming process for Sastrawi.

Table 2. The experimental results of ECS-Dev using SP and MP techniques.

Data test	Size (MB)	SP Time (Sec.)	MP Time (Sec.)	Reduction Time (%)
Wiki-1	1.9	111	45	59.46
Wiki-2	2.5	147	59	59.86
Wiki-3	3.5	214	87	59.35
Wiki-4	4.4	267	114	57.30
Wiki-5	4.8	288	123	57.29
Wiki-6	5.3	321	132	58.88
Wiki-7	5.7	348	153	56.03

Furthermore, the time of stemming processing using ECS-Dev using MP generate a fairly good time reduction from 111 seconds to 45 seconds with percentage of time reduction of 59.46%. The results of the average time reduction from ECS-Dev is also quite high that is above 50%. The graph in Figure 3 shows that processing time stemming ECS-Dev (blue) are below the average processing time (black color). This proves that the technique of MP very efficient to reduce the time of stemming processing.

4.3 Results Comparison

This section discusses the comparison results from previous experiments. In table 3 can be seen that the average processing time of ECS-Sastrawi using SP technique is 1.0631 seconds (2 hours, 57 minutes, 11 seconds) or almost 3 hours. After using MP technique, the processing time decreases to 6.586 seconds (1 hour, 49 minutes, 46 seconds). So, the average of reduction time for ECS-Stemming is 38.47%. While on ECS-Dev, the average processing time using SP is 242 seconds (4 minutes, 2 seconds). After using MP, the processing time decreases to 102 seconds (1 minute, 42 seconds) and the average of reduction time is 58.31%. In addition, the different percentage of average processing time between ECS-Sastrawi and ECS-Dev fairly big enough which is 97.72% for SP and 98.45% SP for MP.

Table 3. Average time comparison between ECS-Sastrawi and ECS-Dev

Stemmer	SP Time (Sec.)	MP Time (Sec.)	Reduction Time (%)
ECS-Sastrawi	111	45	59.46
ECS-Dev	147	59	59.86

Figure 4 shows the comparison of average processing time from the results in table 3. The best result obtained by ECS-Dev using MP technique viz 0.6%. Conversely, the worst result obtained by ECSSastrawi using SP technique viz 60.5%. Based on these results it can be concluded that ECS-Dev outperforms than ECS-Sastrawi using both SP and MP techniques.

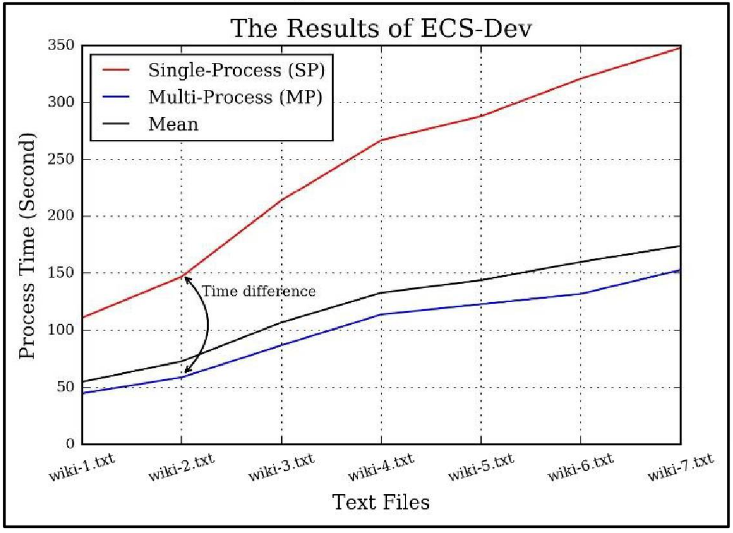

Fig 3: ECS-Dev with time difference and mean between SP and MP techniques.

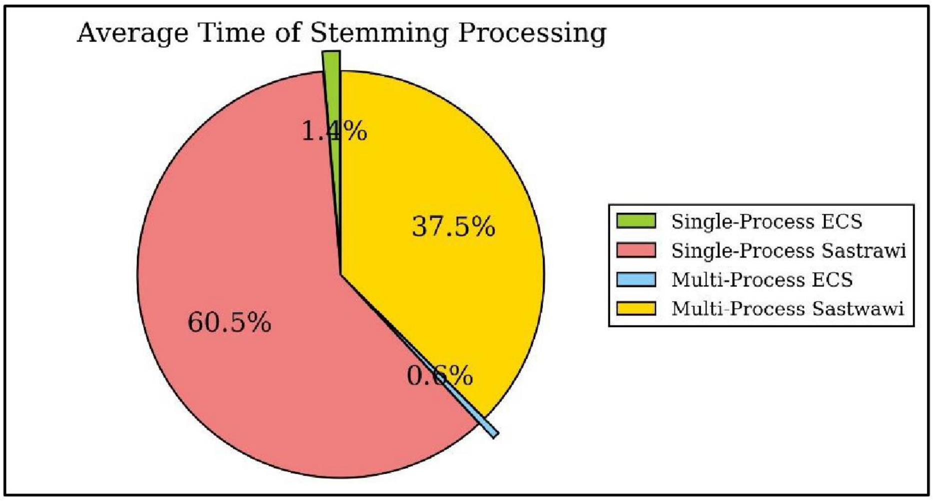

Fig 4: Results comparison for average time of stemming processing

5. CONCLUSION AND FUTURE WORK

Research trend in the field of Natural Language Processing (NLP) is currently increasing. Moreover with the emergence of big data that gives a very large amount of text. So that the need ready-touse programming library for NLP is very urgently. Currently, it is quite widely available. Unfortunately, the library is generally for English and dependently. Therefore, developed specific library for particular language is needed. One technique that is often used in NLP is stemming. ECS algorithm [6] is most widely used for Indonesian stemming. The algorithm has been implemented by Sastrawi library.

Based on some experiments, the time of stemming processing by Sastrawi is still slow, so the speed optimization is needed. This study tries to do the optimization using multiprocess techniques (MP). Beside use the ECS algorithm that already in Sastrawi (ECS-Sastrawi), we also implement the algorithm itself (ECS-Dev). Test results are surprising, the difference average time between ECS-Sastrawi and ECS-Dev is far enough, that is 97.72%. Though the experiment is still using single process (SP) technique. While it using MP technique, the difference is even greater, namely 98.45%. The final results of the experiment can be concluded that the performance of ECS-Dev has outperformed that ECS-Sastrawi using both SP and MP technique.

This research is our steps stone to optimize all natural language processing libraries that specific for Indonesian language. Next study we will do optimization other techniques, such as POS-Tagging, NER and other techniques related to natural language processing.

6. REFERENCES

[1] Sugiyama H., Meguro T., and Higashinaka R., 2017, Evaluation of Question-answering System about Conversational Agents Personality, Dialogues with Social Robots, Springer, Singapore, 183

[2] Yousefi-Azar M. and Hamey L., 2017, Text Summarization Using Unsupervised Deep Learning, Expert Systems with Applications, 93

[3] Negri M., Ataman D., Sabet M. J., Turchi M., and Federico M., 2017, Automatic Translation Memory Cleaning, Machine Translation, 1

[4] Abdiansah A. and Winarko E., 2015, Question Classification Menggunakan Support Vector Machines dan Stemming, Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 34

[5] Tala F. Z., 2003, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia, thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, The Netherlands

[6] Asian J., Williams H. E., and Tahaghoghi S. M., 2005, Stemming Indonesian, the Twenty-eighth Australasian conference on Computer Science, 307

[7] Adriani M., Asian J., Nazief B., Tahaghoghi S. M., and Williams H. E., 2007, Stemming Indonesian: A Confi xStripping Approach, ACM Transactions on Asian Language Information Processing (TALIP), 1

[8] Arifin A. Z., Mahendra I. P., and Ciptaningtyas H. T., 2009, Enhanced Confix Stripping Stemmer and Ants Algorithm for Classifying News Document in Indonesian Language, International Conference on Information and Communication Technology and Systems (ICTS), 60

[9] Tahitoe A. D. and Puriwatasari D., 2010, Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming, thesis, Fakultas Teknologi Informasi, Institut Teknologi Surabaya, Surabaya

[10] Haris M., Maqsood N., Haq U., Zaman T., and Zubair M., 2016, UNI Processor and Multi-processor Performance Comparison, International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), 72

[11] Kertsz G., Kiss D., Lovrics A., Sznsi S., and Vmossy Z., 2016, Multiprocessing of an Individual-Cell Based Model for Parameter Testing, Applied Computational Intelligence and Informatics (SACI), IEEE, 491

[12] Siddiqui I. F., Abbas A., Ariffin A. R., and Lee S. U., 2016, A Comparative Study of Multithreading APIs for Software of ICT Equipment, Indian Journal of Science and Technology, 9