

IMPLEMENTASI METODE RECURRENT NEURAL NETWORK PADA TEXT SUMMARIZATION DENGAN TEKNIK ABSTRAKTIF

Kasyfi Ivanedra¹, Metty Mustikasari²

^{1,2}Teknik Informatika, Universitas Gunadarma
Email: ¹kasyfiivan@gmail.com, ²metty@staff.gunadarma.ac.id

(Naskah masuk: 18 September 2018, diterima untuk diterbitkan: 18 Januari 2019)

Abstrak

Text Summarization atau peringkas text merupakan salah satu penerapan Artificial Intelligence (AI) dimana komputer dapat meringkas text pada suatu kalimat atau artikel menjadi lebih sederhana dengan tujuan untuk mempermudah manusia dalam mengambil kesimpulan dari artikel yang panjang tanpa harus membaca secara keseluruhan. Peringkasan teks secara otomatis dengan menggunakan teknik Abstraktif memiliki kemampuan meringkas teks lebih natural sebagaimana manusia meringkas dibandingkan dengan teknik ekstraktif yang hanya menyusun kalimat berdasarkan frekuensi kemunculan kata. Untuk dapat menghasilkan sistem peringkas teks dengan metode abstraktif, membutuhkan metode Recurrent Neural Network (RNN) yang memiliki sistematis perhitungan bobot secara berulang. RNN merupakan bagian dari Deep Learning dimana nilai akurasi yang dihasilkan dapat lebih baik dibandingkan dengan jaringan saraf tiruan sederhana karena bobot yang dihitung akan lebih akurat mendekati persamaan setiap kata. Jenis RNN yang digunakan adalah LSTM (Long Short Term Memory) untuk menutupi kekurangan pada RNN yang tidak dapat menyimpan memori untuk dipilah dan menambahkan mekanisme Attention agar setiap kata dapat lebih fokus pada konteks. Penelitian ini menguji performa sistem menggunakan Precision, Recall, dan F-Measure dengan membandingkan hasil ringkasan yang dihasilkan oleh sistem dan ringkasan yang dibuat oleh manusia. Dataset yang digunakan adalah data artikel berita dengan jumlah total artikel sebanyak 4515 buah artikel. Pengujian dibagi berdasarkan data dengan menggunakan Stemming dan dengan teknik Non-stemming. Nilai rata-rata recall artikel berita non-stemming adalah sebesar 41%, precision sebesar 81%, dan F-measure sebesar 54,27%. Sedangkan nilai rata-rata recall artikel berita dengan teknik stemming sebesar 44%, precision sebesar 88%, dan F-measure sebesar 58,20 %.

Kata kunci: Text Summarization, Recurrent Neural Network, Abstraktif, Deep Learning, LSTM.

THE IMPLEMENTATION OF TEXT SUMMARIZATION WITH ABSTRACTIVE TECHNIQUES USING RECURRENT NEURAL NETWORK METHOD

Abstract

Text Summarization is the application of Artificial Intelligence (AI) where the computer can summarize text of article to make it easier for humans to draw conclusions from long articles without having to read entirely. Abstractive techniques has ability to summarize the text more naturally as humans summarize. The summary results from abstractive techniques are more in context when compared to extractive techniques which only arrange sentences based on the frequency of occurrence of the word. To be able to produce a text summarization system with an abstractive techniques, it is required Deep Learning by using the Recurrent Neural Network (RNN) rather than simple Artificial Neural Network (ANN) method which has a systematic calculation of weight repeatedly in order to improve accuracy. The type of RNN used is LSTM (Long Short Term Memory) to cover the shortcomings of the RNN which cannot store memory to be sorted and add an Attention mechanism so that each word can focus more on the context. This study examines the performance of Precision, Recall, and F-Measure from the comparison of the summary results produced by the system and summaries made by humans. The dataset used is news article data with 4515 articles. Testing was divided based on data using Stemming and Non-stemming techniques. The average recall value of non-stemming news articles is 41%, precision is 81%, and F-measure is 54.27%. While the average value of recall of news articles with stemming technique is 44%, precision is 88%, and F-measure is 58.20%.

Keywords: Text Summarization, Recurrent Neural Network, Abstractive, Deep Learning, LSTM.

1. PENDAHULUAN

Seiring dengan perkembangan data pada internet yang tumbuh dengan pesat, informasi yang tersedia pada dunia internet sangatlah berlimpah dengan jumlah penulisan dan teks yang begitu besar dan panjang. Untuk mempermudah manusia, maka dibutuhkan suatu alat yang dinamakan peringkas teks otomatis atau *Text Summarization*. Peringkas teks otomatis bertujuan untuk menghasilkan suatu ringkasan dari teks panjang yang diberikan. Dengan menggunakan peringkas teks otomatis, manusia dapat menerima informasi dari teks yang begitu panjang hanya dalam waktu singkat sehingga dapat mempersingkat waktu yang dibutuhkan dibandingkan dengan membaca keseluruhan informasi yang diinginkan (Khan & Naomie, 2014).

Untuk mewujudkan peringkas teks otomatis, dibutuhkan teknik yang terbagi menjadi dua, yaitu peringkas dengan teknik Ekstraktif dan teknik Abstraktif. Teknik ekstraktif mengambil setiap kata pada teks asal lalu dikelompokkan dan disusun tanpa mengubah kata asal. Biasanya kalimat yang dihasilkan tersusun sama dengan kalimat pada dokumen asal (Prabowo, 2016). Teknik abstraktif memiliki tujuan untuk menghasilkan kalimat ringkasan dengan konsep sebagaimana manusia mengambil intisari dari sebuah dokumen yang dibaca (Khan & Naomie, 2014). Dengan begitu, kalimat yang dihasilkan akan terkesan lebih simpel, alami, dan tidak kaku.

Untuk dapat menerapkan peringkas teks otomatis secara abstraktif dibutuhkan perhitungan yang lebih dalam mencari persamaan setiap kata sehingga menghasilkan ringkasan. Untuk itu dibutuhkan pendekatan *Deep Learning*. Dengan *deep learning*, input akan dihitung melalui layer yang lebih banyak dibandingkan JST sederhana (Goodfellow, 2016). RNN merupakan penerapan *deep learning* yang digunakan.

Penggunaan metode RNN untuk peringkas teks otomatis diawali dari keberhasilan pemecahan masalah untuk mesin translasi kata (Nallapati et al, 2016). RNN diharapkan dapat menghasilkan model yang dapat menghasilkan teks ringkasan secara abstrak. Selain menggunakan RNN, model membutuhkan arsitektur tambahan yaitu *attention* (Bahdanau, 2014). Arsitektur ini berfungsi untuk memfokuskan setiap kata yang masuk ke sel RNN pada target output sehingga hasil dapat sesuai dengan konteks dokumen asal.

Penelitian yang dilakukan oleh Nedunchelian Ramanujam dan Manivannan Kaliappan menggunakan *Naive Bayesian Classifier (NBC)* sebagai metode utama dalam menghasilkan model peringkas teks otomatis dalam bentuk ekstraktif. Metode NBC menggunakan teknik perhitungan frekuensi kata. Kata dengan frekuensi terbanyak akan dipilih untuk dijadikan kalimat baru berupa ringkasan. Metode NBC dapat ditingkatkan dengan teknik abstraktif untuk menghasilkan ringkasan yang

lebih pendek. Kelemahan metode ekstraktif ini adalah pada hasil ringkasan yang cenderung lebih kaku dan tidak alami. Hasil ringkasan hanya berupa perpaduan kalimat-kalimat terpilih dengan frekuensi kemunculan terbanyak sehingga terkadang tidak sesuai dengan konteks kalimat awal (Ramanujam, 2016). Sedangkan penelitian oleh C. Sunitha menjelaskan perbedaan antara metode ekstraktif dan abstraktif. Metode abstraktif dapat menghasilkan ringkasan yang dinilai lebih tepat, komplit, dan lebih bermakna dibandingkan dengan metode ekstraktif. Metode abstraktif juga dapat mengurangi redundansi pada konsep dan relasi antar kata sehingga ringkasan dapat lebih tepat sesuai konteks (Sunitha, 2016).

Pada paper *SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents*, metode yang digunakan adalah RNN dengan teknik ekstraktif. RNN digunakan untuk mencari hubungan antara kata dari banyaknya frekuensi kata yang muncul. Sedangkan dengan LSTM, memiliki mekanisme yang lebih mengedepankan memori dari langkah pelatihan sebelumnya untuk mendapatkan hubungan antar kata sehingga ideal untuk memproses *text speech, translation, dan summarization* (Nallapati, dkk, 2016).

Paper ini terdiri dari 5 bab, dimulai dari pendahuluan menjelaskan secara global penggunaan metode LSTM. Dasar teori pada bab 2 menjelaskan teori-teori yang digunakan pada penulisan. Bab 3 berisikan metode penelitian yang digunakan. Hasil pengujian dan evaluasi dijelaskan pada bab 4 dan bab 5 berisikan kesimpulan serta saran.

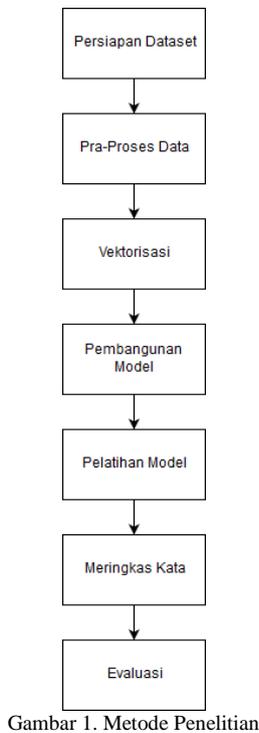
2. METODE PENELITIAN

Metode penelitian yang dilakukan ditampilkan pada Gambar 2.

Metode dimulai dari mempersiapkan dataset yang digunakan sebagai data latih. Lalu data tersebut diolah agar menjadi data yang memiliki makna pada tahap pra-proses data. Pra-proses data memiliki tahapan mulai dari tokenisasi, mengubah kata menjadi *lower case, stopword removal, dan stemming* (Johnston, 2013). Kata diubah menjadi vektor dalam proses vektorisasi. Model dibangun dengan menggunakan metode yang menjadi variansi dari RNN yaitu LSTM (*Long Short Term Memory*). Model tersebut digunakan untuk melatih data yang sudah dalam bentuk vektor. Setelah pelatihan selesai, dilakukan uji coba peringkas teks. Tahap terakhir adalah evaluasi terhadap keakuratan model yang sudah dibuat.

Metode dimulai dari mempersiapkan dataset yang digunakan sebagai data latih. Lalu data tersebut diolah agar menjadi data yang memiliki makna pada tahap pra-proses data. Pra-proses data memiliki tahapan mulai dari tokenisasi, mengubah kata menjadi *lower case, stopword removal, dan stemming* (Johnston, 2013). Kata diubah menjadi vektor dalam proses vektorisasi. Model dibangun

dengan menggunakan metode yang menjadi variasi dari RNN yaitu LSTM (*Long Short Term Memory*). Model tersebut digunakan untuk melatih data yang sudah dalam bentuk vektor. Setelah pelatihan selesai, dilakukan uji coba peringkasan teks. Tahap terakhir adalah evaluasi terhadap keakuratan model yang sudah dibuat.



2.1. Spesifikasi Hardware

Hardware yang digunakan sebagai pelatihan adalah sebagai berikut:

Tabel 2. Spesifikasi Hardware

Perangkat	Spesifikasi
Laptop Notebook	Acer Aspire E14 dengan tipe E5-475G
Prosesor	Intel Core i5-7200U 2.5GHz
RAM	4GB bertipe DDR4
Grafik	NVIDIA GeForce 940MX

2.2. Pra-Proses Data

Tahap awal dari pengerjaan program adalah dengan menyiapkan sejumlah data yang akan digunakan sebagai bahan pembelajaran oleh komputer. Data inilah yang menjadi acuan bagaimana model akan terbentuk. Tahap yang akan dilakukan pada pra-proses adalah tokenisasi, mengubah kata menjadi *lower case*, *stopword removal*, dan *stemming*.

2.3. Vektorisasi

Pada tahap ini, kata yang sudah dibersihkan dan siap digunakan diubah terlebih dahulu menjadi vektor dengan tujuan agar kata dapat dihitung dan

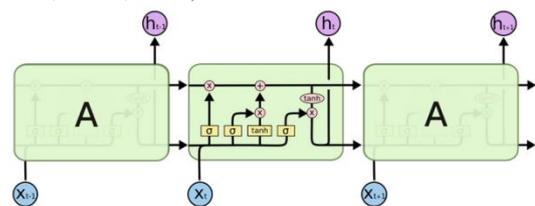
memiliki nilai persamaan antara satu kata dengan kata yang lain. Vektorisasi pada penelitian ini menggunakan vektor yang sudah diolah sebelumnya (pra-proses).

2.4. Pembangunan Model

Metode yang digunakan adalah dengan menerapkan Jaringan Syaraf Tiruan atau yang dikenal dengan istilah *Neural Network*. Jaringan syaraf tiruan memiliki sistem kerja yang menyerupai kerja syaraf di otak manusia. Setiap jaringan memiliki input layer, *hidden layer*, dan output layer (Daniel, 2013).

Salah satu jenis NN adalah *Recurrent Neural Network* atau bisa disebut RNN. Jaringan ini memiliki sedikit perbedaan daripada jaringan NN biasa. Jika pada NN, setiap input akan masuk lalu melalui *hidden layer* dan mengeluarkan output, sedangkan pada RNN terdapat perulangan. Setiap input yang masuk dan menghasilkan output, akan masuk kembali menjadi input untuk diproses di dalam *hidden layer* (Graves, 2012). Begitu seterusnya sehingga memperoleh output sesuai dengan target.

Model dibangun dengan metode RNN secara *bi-directional* atau perhitungan bobot pada RNN secara bolak-balik. Jenis variasi RNN yang digunakan adalah LSTM yang memiliki kemampuan untuk mengolah informasi lebih akurat (Zhang, 2016). LSTM memiliki kelebihan daripada sekedar RNN biasa, yaitu adanya arsitektur mengingat dan melupakan output yang akan diproses kembali menjadi input. Selain itu, kemampuan LSTM yang lain adalah dapat mempertahankan *error* yang terjadi ketika melakukan *backpropagation* sehingga tidak memungkinkan kesalahan meningkat (Zhang, 2016; Fauzi, 2016).



Gambar 2. Sel LSTM

Mekanisme *attention* digunakan untuk memfokuskan setiap hasil output LSTM terhadap target kata. Dengan *attention*, kata yang dihasilkan memiliki hubungan yang lebih baik sehingga lebih sesuai dengan konteks (Bahdanau et al, 2014).

Model memiliki beberapa *hyperparameter* yang digunakan (Greff, 2017). *Learning rate* minimal yang digunakan adalah 0.005 dengan jeda senilai 0.95. *Epoch* sebanyak 100 kali. Ukuran *batch* yang digunakan sebesar 64 dan ukuran RNN sel nya adalah 256, serta menggunakan 2 layer.

2.5. Pelatihan Data

Pada tahap *training*, data yang sudah siap untuk digunakan akan terlebih dahulu diubah menjadi vektor pada *word embedding*. Lalu input tersebut akan diolah pada *encoder layer*. Pada *encoder layer*, sel LSTM akan memproses input sehingga menghasilkan output yang akan diteruskan ke *hidden layer*. Pada *hidden layer*, data yang dilatih akan dicatat dalam bentuk logistik untuk memperbaharui *weight* dan nilai yang dibutuhkan agar mencapai target output. *Decoder layer* akan menerima hasil dari *hidden layer* untuk kemudian difokuskan menggunakan *attention* sehingga mendapatkan bobot nilai yang mendekati nilai target output. Lalu bentuk vector akan diubah kembali menjadi teks untuk kemudian ditampilkan sebagai output teks.

2.6. Peringkasan Kata

Setelah model dilatih, maka tahap terakhir adalah melakukan test terhadap kalimat yang diinput kedalam jaringan. Kalimat akan diproses menggunakan model yang sudah dihasilkan oleh tahap *training* untuk menghasilkan hasil ringkasan.

3. HASIL PENGUJIAN DAN EVALUASI

3.1. Persiapan Dataset

Dataset yang adalah dataset *news* atau berita yang berisi kumpulan dari artikel berita *Hindu*, *Indian Times*, dan *Guardian*. Dataset ini terdiri dari 4515 buah artikel beserta ringkasan teks sebagai target output. Semua jumlah dataset digunakan untuk bagian pelatihan dan test.

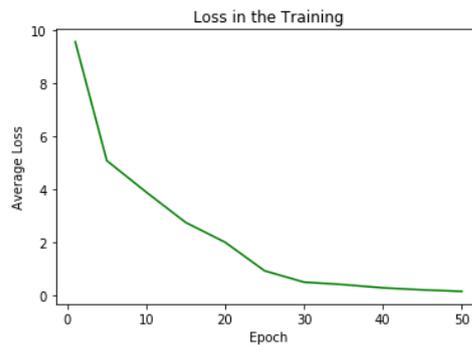
3.2. Hasil Pelatihan Data

Tahap pelatihan atau *training* dilakukan dirangkum pada Tabel 3 berikut:

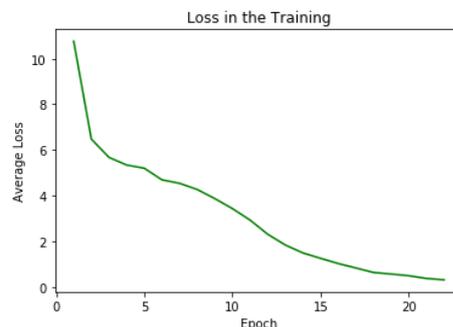
Tabel 3. Training model dengan dataset

Dataset	Banyak Data	Waktu Latih	Epoch	Loss
Non-Stem	1959	1 Jam 15 menit	51/100	0.32
Stem	729	35 menit	24/100	0.34

Dari uji coba pelatihan pada model, dataset yang menggunakan teknik *stemming* selesai lebih cepat dibandingkan dengan dataset *non-stemming*. Tetapi *loss* yang dihasilkan dataset *non-stemming* lebih sedikit dibandingkan dengan artikel berita yang menggunakan *stemming*. Jumlah data berpengaruh terhadap waktu *training*. Proses *training* disajikan dalam bentuk grafik Gambar 3 dan 4, dimana rata-rata *loss* berbanding dengan berapa kali *epoch* yang dilakukan.



Gambar 3. Grafik Training Artikel Non-Stemming



Gambar 4. Grafik Training Artikel Stemming

3.3. Hasil Ringkasan

Langkah berikutnya adalah melakukan peringkasan oleh model yang sudah dilatih. Pengujian menggunakan data sebanyak 3 buah teks berbahasa Inggris dan masing-masing artikel diuji terhadap dataset yang berbeda. Ringkasan yang dihasilkan oleh model dibandingkan dengan hasil ringkasan oleh manusia yaitu sebuah kalimat dengan jumlah maksimal 12 kata. Hasil ringkasan disajikan dalam bentuk Tabel 4, Tabel 5, dan Tabel 6 berikut:

Tabel 4. Hasil Ringkasan Teks 1

Artikel	1
Teks Asal	Pakistan-born London terrorist Khuram Butt, who masterminded last Saturday's London Bridge attack, was trying to land a job with a firm that provides security at Wimbledon and other sporting events. The 27-year-old had previously worked six months with the London Underground metro before quitting in October. Butt and his aides rammed a high-speed van into pedestrians on the London Bridge.
Artikel Berita non-stemming	pak born london terrorist trying
Artikel Berita stemming	pak born london terrorist tri

Tabel 5. Hasil Ringkasan Teks 2

Artikel	2
Teks Asal	'Harry Potter' actor Daniel Radcliffe helped a tourist who was attacked by two men on bikes in London. The victim was walking when the men stole his bag and left him with an injury. "He was a really nice bloke, a lot of stars wouldn't have stopped to help," said a former policeman who tried to intervene during the robbery.
Artikel Berita non-stemming	harry potter actor daniel play
Artikel Berita stemming	harri potter actor daniel help

Tabel 6. Hasil Ringkasan Teks 3

Artikel	3
Teks Asal	Filmmaker Christopher Nolan has revealed that he watched the Oscar-winning film 'La La Land' thrice. He added, 'I think I almost went back to see if it was as good as I had thought, and it was.' Notably, the Emma Stone and Ryan Gosling starrer 'La La Land' won six Oscar Awards and received a record fourteen Oscar nominations.
Artikel Berita non-stemming	christopher nolan reveals watched la la
Artikel Berita stemming	christoph nolan reveal watch la la

3.2. Hasil Evaluasi

Metode pengujian menggunakan metode intrinsik. Metode intrinsik yang digunakan adalah *content evaluation*. Metode ini menghitung akurasi dengan mencari nilai *precision*, *recall*, dan *F-Measure* (Steinberg, 2009); (Ramanujam, 2016). *Precision* merupakan pengukuran tingkat ketepatan hasil ringkasan yang dihasilkan oleh peringkas teks otomatis. *Precision* juga mengukur relevansi peluang teks yang terambil sebagai ringkasan. *Recall* merupakan pengukuran tingkat keberhasilan ringkasan yang dihasilkan peringkas teks otomatis. *Recall* juga dapat mengukur peluang teks relevan yang terambil sebagai ringkasan. Sedangkan *F-Measure* merupakan nilai yang digunakan untuk mengukur akurasi dari ringkasan yang dihasilkan.

$$Recall = \frac{\sum \text{Kata Relevan}}{\sum \text{Kata ringkasan Manual}} \quad (1)$$

$$Precision = \frac{\sum \text{Kata Relevan}}{\sum \text{Kata ringkasan Sistem}} \quad (2)$$

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (3)$$

Kata relevan merupakan kata-kata pada ringkasan yang dihasilkan oleh peringkas teks otomatis sama dengan kata-kata yang dihasilkan secara manual oleh manusia. Untuk mendapatkan

nilai *recall*, maka kata relevan dibagi dengan jumlah kata pada ringkasan manual. Untuk mendapatkan nilai *precision*, maka kata relevan dibagi dengan jumlah kata pada ringkasan sistem. Sedangkan untuk mendapatkan nilai akurasi dari ringkasan, dengan mencari nilai *F-Measure*, yaitu berdasarkan nilai *recall* dan nilai *precision* (Steinberg, 2009).

Ringkasan manual didapat dengan memberikan kusioner terhadap 5 orang yang memiliki kemampuan memahami teks dalam bahasa inggris dengan baik. Ringkasan manual yang dihasilkan maksimal 12 kata. Ringkasan manual yang akan digunakan sebagai ringkasan perbandingan adalah ringkasan yang memiliki kata yang sama dengan frekuensi tertinggi yang sudah dihasilkan oleh kelima responden.

Tabel 7. Uji Evaluasi Perbandingan Teks Ringkasan

Teks	\sum KRM	\sum KRS	\sum KRS 2	KR	KR 2
1	10	5	5	4	4
2	12	5	5	4	5
3	10	6	6	5	5

Keterangan:

- KRM = Kata Ringkasan Manual
- KRS = Kata Ringkasan Sistem Non-Stemming
- KRS 2 = Kata Ringkasan Sistem stemming
- KR = Kata Relevan Non-stemming
- KR 2 = Kata Relevan stemming

Hasil ringkasan sistem dibandingkan dengan hasil ringkasan manual untuk mendapatkan nilai *precision* berdasarkan persamaan (2), *recall* berdasarkan persamaan (1), dan *F-measure* berdasarkan persamaan (3). Nilai akurasi tersebut dirangkum pada tabel 6 dan 7 berikut:

Tabel 8. Hasil Evaluasi Metode Intrinsik pada Artikel Non-Stemming

Teks	Recall (%)	Precision (%)	F-Measure (%)
1	40%	80%	53,3%
2	33,3%	80%	47%
3	50%	83,3%	62,5%
Rata-Rata	41%	81%	54,27%

Tabel 9. Hasil Evaluasi Metode Intrinsik pada Artikel Stemming

Teks	Recall (%)	Precision (%)	F-Measure (%)
1	40%	80 %	53,3%
2	41,6%	100%	58,8%
3	50%	83,3%	62,5%
Rata-Rata	44%	88%	58,20%

Dari tahapan pengujian evaluasi yang telah dilakukan terhadap peringkas teks otomatis dengan menggunakan metode intrinsik, dapat diketahui bahwa nilai rata-rata *recall* artikel berita non-stemming adalah sebesar 41%, *precision* sebesar

81%, dan *F-measure* sebesar 54,27%. Sedangkan nilai rata-rata *recall* artikel berita dengan teknik *stemming* sebesar 44%, *precision* sebesar 88%, dan *F-measure* sebesar 58,20 %.

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Setelah melakukan pengujian dan evaluasi terhadap peringkasan teks otomatis, ada beberapa poin yang dapat disimpulkan. Perbedaan jumlah dataset yang digunakan sebagai bahan *training* sangat berpengaruh, karena pembendaharaan kata yang lebih banyak tentunya menjadikan program lebih mengerti dan lebih akurat dalam menciptakan ringkasan. Nilai rata-rata *recall* artikel berita non-*stemming* adalah sebesar 41%, *precision* sebesar 81%, dan *F-measure* sebesar 54,27%. Sedangkan nilai rata-rata *recall* artikel berita dengan teknik *stemming* sebesar 44%, *precision* sebesar 88%, dan *F-measure* sebesar 58,20 %.

4.2. Saran

Penambahan jumlah pada dataset yang akan digunakan dapat mempengaruhi hasil ringkasan. Semakin banyak data yang dilatih, maka semakin ‘pintar’ pula program peringkasan teks, karena program memiliki pembendaharaan yang banyak dari berbagai macam contoh ringkasan data. Disamping itu, untuk menghasilkan data yang lebih layak untuk digunakan dapat ditambah dengan menggunakan teknik *lemmatize* agar hasil ringkasan dapat dengan fokus menghasilkan kalimat yang sesuai dengan konteks.

DAFTAR PUSTAKA

- BAHDANAU, DZMITRY., CHO, KYUNGHYUN., & BENGIO, YOSHUA., 2014. *Neural Machine Translation by Jointly Learning to Align and Translate*. CoRR, abs/1409.0473. Conference ICLR.
- C. SUNITHA., JAYA, DR.A., GANESH, AMAL., 2016, *A study on Abstractive Summarization Techniques in Indian Language*, Fourth International Conference on Recent Trends in Computer Science & Engineering, Elsevier B.V, pp.25-31.
- DANIEL, GRAUPE, 2013, *Principles of Artificial Neural Network (3rd Edition)*, Word Scientific, Singapore.
- FAUZI, RAHMAT, 2016, *Implementasi Jaringan Syaraf Tiruan dengan Metode Backpropagation terhadap Bibit Tanaman Karet*. Jurnal Education and Development STKIP Tapanuli Selatan, pp. 1-11.
- GOODFELLOW, IAN., BENGIO, YOSHUA., & COURVILLE, AARON., 2016, *Deep Learning*, MIT Press Cambridge, Inggris.
- GREFF, KALUS., SRIVASTAVA, RUPESH K., KOUTNNIK, JAN., STEUNEBRINK, BAS R., SCHMIDHUBER, JURGEN., 2017. *LSTM: A Search Space Odyssey*. Transactions on Neural Network and Learning System.
- JOHNSTON, LINDSAY, 2013, *Data Mining: Concepts, Methodologies, Tools, and Applications*, Vol 1, Information Science Reference, USA.
- KHAN, ATIF & NAOMIE SALIM, 2014. A Review on Abstractive Summarization Methods. *Journal of Theoretical and Applied Information Techonlogy*, pp.64-71.
- NALLAPATI, RAMESH., ZHAI, FEIFEI., ZHOU, BOWEN, 2016. *SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents*. The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-2017)
- NALLAPATI, RAMESH., ZHOU, BOWEN., SANTOS, DOS CICERO., GULCEHRE, CAGLAR., & XIANG, BING., 2016. *Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond*. The SIGNLL Conference on Computational Natural Language Learning (CoNLL)
- PRABOWO, D.A., FHADLI, M., NAJIB, M.A., FAUZI, H.A., & CHOLISSODIN, IMAM., 2016. *TF-IDF-Enchanted Genetic Algorihm untuk Extractive Automatic Text Summarization*, Vol 3, No. 3, Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), pp.208-215.
- RAMANUJAM, N & KALIAPPAN, M, 2016, *An Automatic Multidocument Text Summarization Approach Based on Naïve Bayesian Classifier Using Timestamp Strategy*, The Scientific World Journal, Hindawi Publishing, pp.1-10.
- STEINBERG, J. & JEZEK, K., 2009, *Evaluation Measures for Text Summarization*, Computing and Informatics, vol. 28, no.2, pp. 251-275.
- ZHANG, LIN, 2016, *Theory, Methodology, Tools, and Applications for Modeling and Simulation of Complex Systems*, Springer + Business Media, Singapore.