Jurnal Riset Inovasi Bidang Informatika Dan Pendidikan Informatika (KERNEL)
Vol. 3, No. 2, Desember 2022
ISSN: 2774-4345

Penentuan Relevansi Artikel Ilmiah dengan Metode Word2Vec

*Kaisul F. Dewananda[1], Weny M. Rahmawati[2], Septiyawan R. Wardhana[1], Gusti E. Yuliastuti[1]**

[1]Institut Teknologi Adhi Tama Surabaya

[2]Politeknik Elektronika Negeri Surabaya

**Penulis Korespondensi: gustiekay@itats.ac.id*

ABSTRACT

ArXiv is a place to store electronic preprints including scientific articles and can be accessed by all internet users online. Over time, the number of scientific article documents has increased. This can reduce the level of effectiveness of grouping scientific articles based on the user's area of interest. A recommendation system is needed to determine the relevance of scientific articles that can provide users with recommendations for scientific articles that are in accordance with the user's areas of interest, so that users will find it easier to access scientific articles on the system. The author will do word embedding by applying the Word2Vec method. Word2Vec is a machine learning method for obtaining high quality vector. Based on the test results, it can be concluded that the Word2Vec method can be used to produce recommendations for scientific articles according to the user's areas of interest and produce the best parameter values n=150, epoch=100, window size=3 and learning rate=0.01.

Article History

Received 25-01-2023
Revised 27-02-2023
Accepted 27-02-2023

Key words

Artikel Ilmiah
ArXiv
Sistem Rekomendasi
Word2Vec
Word Embedding

ABSTRAK

ArXiv merupakan tempat menyimpan pracetak elektronik termasuk artikel ilmiah dan dapat diakses oleh semua pengguna internet secara *online*. Seiring berjalannya waktu, dokumen artikel ilmiah makin bertambah banyak. Hal tersebut dapat menurunkan tingkat efektifitas pengelompokkan artikel ilmiah berdasarkan bidang keminatan pengguna. Suatu sistem rekomendasi diperlukan untuk penentuan relevansi artikel ilmiah dapat memberi pengguna rekomendasi artikel ilmiah yang sesuai bidang keminatan pengguna, sehingga pengguna akan lebih mudah dalam mengakses artikel ilmiah pada sistem tersebut. Penulis akan melakukan *word embedding* dengan menerapkan metode *Word2Vec*. *Word2Vec* merupakan metode pembelajaran mesin untuk mendapatkan sebuah vektor berkualitas tinggi. Berdasarkan hasil pengujian didapatkan kesimpulan bahwa metode *Word2Vec* dapat digunakan untuk menghasilkan rekomendasi artikel ilmiah sesuai bidang keminatan pengguna dan menghasilkan nilai parameter terbaik n=150, *epoch*=100, *window size*=3 dan *learning rate*=0.01.

PENDAHULUAN

ArXiv merupakan tempat penyimpanan pracetak elektronik (biasa disebut cetakan elektronik) artikel ilmiah di bidang matematika, fisika, astronomi, ilmu komputer, biologi kuantitatif, statistika, dan keuangan kuantitatif [1]. ArXiv dapat diakses oleh semua pengguna internet secara *online*. ArXiv.org dirintis pada tanggal 14 Agustus 1991 dan berhasil mencetak artikel ke-500.000 pada 3 Oktober 2008 [2]. Pada tahun 2014, tingkat publikasinya naik menjadi 8.000 artikel setiap bulannya. Informasi tentang artikel ilmiah di *website* tersebut tumbuh semakin besar dari hari ke hari. Dengan jumlah artikel ilmiah yang sangat besar ini, sistem biasa akan kurang efektif dalam melakukan pengelompokkan bidang keminatan pengguna. Suatu sistem rekomendasi diperlukan untuk penentuan relevansi artikel ilmiah dapat memberi pengguna rekomendasi artikel ilmiah yang sesuai bidang keminatan pengguna, sehingga pengguna akan lebih mudah dalam mengakses artikel ilmiah pada sistem tersebut.

Sistem rekomendasi dapat dikategorikan ke dalam dua jenis yakni *Collaborative Filtering* dan *Content Based Filtering*. Pendekatan *Collaborative Filtering* memperkirakan faktor keminatan untuk pengguna dengan menganalisis preferensi pengguna lain yang telah mengalami *item* tersebut [3]. Sedangkan pendekatan *Content Based Filtering* menggunakan deskripsi dari profil pengguna atau deskripsi suatu *item* dalam menghasilkan suatu rekomendasi [4]. *Content Based Filtering* diperlukan untuk menganalisis deskripsi *item* dalam mengidentifikasi *item* yang menarik bagi pengguna tanpa harus mempunyai informasi sebelumnya dari pengguna.

Pada *Text Mining*, diperlukan suatu representasi angka terhadap teks atau bisa juga disebut sebagai *Word Embedding* [5]. Hal tersebut dilakukan untuk memudahkan kita melakukan proses matematis pada teks tersebut, serta mendapatkan hasil yang efisien secara cepat. Terdapat banyak metode yang dapat diterapkan untuk melakukan *Word Embedding*, salah satunya yakni *Word2Vec* [6]. Pada penelitian ini, penulis akan membahas mengenai rekomendasi karya ilmiah berdasarkan abstrak dengan menggunakan metode *Word2Vec*. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi artikel ilmiah yang sesuai dengan bidang keminatan pengguna.

TINJAUAN PUSTAKA

ArXiv

Arxiv adalah layanan distribusi gratis dan arsip terbuka untuk artikel ilmiah di bidang fisika, matematika, ilmu komputer, biologi kuantitatif, keuangan kuantitatif, statistik, teknik elektro dan ilmu sistem, dan ekonomi. Arxiv adalah sumber daya yang didukung masyarakat dan didanai secara kolaboratif yang didirikan oleh Paul Ginsparg pada tahun 1991 dan dikelola dan dioperasikan oleh Cornell University [1].

Sistem Rekomendasi

Sistem rekomendasi adalah subkelas sistem penyaringan informasi yang berupaya memprediksi "peringkat" atau "preferensi" yang akan diberikan pengguna kepada suatu *item* [4], terutama digunakan dalam aplikasi komersial. Ada juga sistem rekomendasi populer untuk topik tertentu seperti restoran dan kencan *online*. Sistem rekomendasi telah dikembangkan untuk mengeksplorasi artikel dan pakar penelitian, kolaborator, serta layanan keuangan.

Text Mining

Text Mining merupakan suatu proses pengambilan intisari dari dokumen teks sehingga didapatkan hasil yang berguna untuk tujuan tertentu [7]. Tujuan dari *Text Mining* yakni untuk memroses data tidak terstruktur dan mengekstrak pola yang terbentuk sehingga dapat diambil keputusan yang baik. Dengan kata lain, pemrosesan teks berkaitan dengan transmisi informasi secara otomatis. Tidak seperti algoritma, pemrosesan teks dapat dianggap sebagai makro yang dikelola secara berurutan dan lebih sederhana. Pemrosesan teks memiliki teknik penyaringan dan melihat ekspresi pola-aksi. Dalam *Text Mining* memiliki sub bagian yakni *Word Embedding. Word Embedding* bertujuan untuk menghasilkan kata atau frase yang dipetakan ke vektor bilangan riil. Dalam *Word Embedding* terdapat dua kegiatan utama yakni pemodelan bahasa dan teknik pembelajaran fitur. Kedua kegiatan tersebut dapat dilakukan dengan menggunakan Word2Vec yang berbasis *Neural Net.*

Word Embedding

Dalam istilah yang sangat sederhana, *Word Embedding* adalah teks yang dikonversi menjadi angka dan mungkin ada representasi numerik yang berbeda dari teks yang sama. Word Embedding digunakan sebagai representasi makna yang lebih baik karena adanya keterbatasan informasi pada teks pendek [8]. Salah satu metode yang dapat diterapkan yakni *Word2Vec*.

Word2Vec merupakan representasi dari *Word Embedding*. *Word2Vec* merupakan metode pembelajaran mesin untuk mendapatkan sebuah vektor berkualitas tinggi [5]. Vektor tersebut merepresentasikan fitur kata dengan panjang variabel tetap (*fixed-length*) dari potongan-potongan teks seperti kalimat, paragraf atau bahkan dokumen. Metode *Word2Vec* mewakili setiap dokumen

oleh vektor kata yang melalui tahap pelatihan untuk memprediksi kata yang ada dalam dokumen [9]. Pada proses prediksi, vektor suatu paragraf disimpulkan dengan memperbaiki vektor kata dan melatih vektor paragraf baru hingga mencapai konvergensi. Konsep kerja dari metode *Word2Vec* menyerupai teknik dasar dari *Neural Network* [10][11][12] dimana terdapat dua model yang dikenalkan yaitu *Continuous-Bag-of-Words* (CBOW) dan *SkipGram* [13].

METODE

Word2Vec

Pada penelitian ini, penulis menggunakan metode *Word2Vec* untuk penentuan relevansi artikel ilmiah. Adapun beberapa tahapan pada metode *Word2Vec*, antara lain:

1. Inisialisasi *window size*, *learning rate* dan jumlah *hidden neuron*.
2. Inisialisasi vektor kata dari kata target dan vektor kata konteks. Kata akan bernilai 1 apabila termasuk ke dalam kata target ataupun kata konteks sedangkan kata lainnya diberi nilai 0. Banyaknya kata konteks berdasarkan jumlah dari *window size*.
3. Inisialisasi nilai bobot awal matriks konteks (W) dan matriks *embedding* (W') dengan membangkitkan nilai secara acak.
4. Pada setiap *epoch*, lakukan proses sebanyak 5 hingga 9 kali.
5. Pada setiap kata yang menjadi kata target, lakukan proses tahap 6 hingga 9.
6. Hitung *Forward Pass*
   a. Hitung nilai bobot dari *input layer* ke *hidden layer* menggunakan Persamaan 1.
   $$h = w^T . x \quad \ldots\ldots (1)$$
   Keterangan:
   h : *hidden layer*
   x : *input* vektor dalam target
   w^T : bobot *input layer* ke *hidden layer*
   b. Hitung nilai bobot dari *hidden layer* ke *output layer* menggunakan Persamaan 2.
   $$u_j = {w_2}^T . h \quad \ldots\ldots (2)$$
   Keterangan:
   u_j : *output* baris ke-j
   h : *hidden layer*
   ${w_2}^T$: bobot *hidden layer* ke *output layer* yang sudah ditransposisi
7. Hitung *Softmax Output* menggunakan Persamaan 3.
   $$y_j = \frac{\exp(u_j)}{\sum_{j=1}^{v} \exp\,(u_j{}')} \quad \ldots\ldots (3)$$
   Keterangan:
   u_j : *output* baris ke-j
   y_j : nilai *softmax* baris ke-j
   v : jumlah kata yang unik
8. Hitung nilai kesalahan (*error*) menggunakan Persamaan 4.
   $$e = \sum_{j=1}^{c} (y_j - x_c) \quad \ldots\ldots (4)$$
   Keterangan:
   e : nilai *error*
   x_c : nilai vektor kata konteks
   y_j : nilai *softmax* baris ke-j
9. Hitung *Backpropagation*
   a. Hitung nilai bobot dari *output layer* ke *hidden layer* menggunakan Persamaan 5.
   $$dl'_{dw} = h \otimes e^t \quad \ldots\ldots (5)$$

Keterangan:
dl'_{dw} : delta bobot *output layer* ke *hidden layer*
h : nilai *hidden layer* dari hasil *forward pass*
e^T : nilai *error* yang ditransposisi
$\otimes$: operator *outer product*

b. Hitung nilai bobot dari *hidden layer* ke *input layer* menggunakan Persamaan 6.

$$dl_{dw} = X_t \otimes (W' . e) \quad \dots\dots (6)$$

Keterangan:
dl_{dw} : delta bobot *input layer* ke *hidden layer*
X_t : *input* vektor kata target
e : nilai *error*
W' : bobot *hidden layer* ke *output layer*
$\otimes$: operator *outer product*

10. Hitung pemutakhiran bobot
    a. Hitung pemutakhiran bobot dari *input layer* ke *hidden layer* menggunakan Persamaan 7.

$$W = W^{(lama)} - (\alpha \, . \, dl_{dw}) \quad \dots\dots (7)$$

Keterangan:
W : nilai bobot baru dari *input layer* ke *hidden layer*
$W^{(lama)}$: nilai bobot lama dari *input layer* ke *hidden layer*
α : *learning rate* (parameter penurunan fungsi waktu)
dl_{dw} : delta bobot *input layer* ke *hidden layer*

    b. Hitung pemutakhiran bobot dari *hidden layer* ke *output layer* menggunakan Persamaan 8.

$$W' = {W_2}^{(lama)} - (\alpha \, . \, {dl'}_{dw}) \quad \dots\dots (8)$$

Keterangan:
W' : nilai bobot baru dari *hidden layer* ke *output layer*
${W_2}^{(lama)}$: nilai bobot lama dari *hidden layer* ke *output layer*
α : *learning rate* (parameter penurunan fungsi waktu)
${dl'}_{dw}$: delta bobot *hidden layer* ke *output layer*

11. Hitung nilai *error* dengan menggunakan fungsi kerugian pada Persamaan 9.

$$e = -\sum_{j=1}^{c} u_j + C \, . \log \sum_{j=1}^{v} \exp(u_j) \quad \dots\dots (9)$$

Keterangan:
e : nilai *error*
u_j : nilai *output* indeks ke-j
C : panjang vektor kata konteks

Document Similarity

Document similarity digunakan untuk mendapatkan nilai relevansi dari sebuah dokumen. Untuk mendapatkan nilai *document similarity* diperlukan perhitungan nilai rata-rata vektor kata yang ada pada setiap dokumen dan hasil rata-rata tersebut dihitung menggunakan *cosine similarity*.

Cosine similarity merupakan salah satu metode yang diterapkan untuk suatu sistem temu kembali informasi. Pada metode ini akan diukur derajat kesamaan jarak antara dua dokumen [5]. Perhitungan jarak *cosine similarity* pada dua dokumen menggunakan Persamaan 10.

$$sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \times \sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad \dots\dots (10)$$

HASIL DAN PEMBAHASAN

Pada penelitian ini, proses pelatihan menggunakan data latih sebanyak 41.000 abstrak karya ilmiah yang berasal dari Kaggle. Sedangkan proses pengujiannya menggunakan data uji sebanyak 200 abstrak karya ilmiah.

Pengujian *Word Similarity*

Pada pengujian *word similarity* ini terdapat 6 skenario pengujian yang berbeda nilai parameternya. Adapun parameter yang digunakan, antara lain: *n, epoch, window size* dan *learning rate.* Hasil pengujian *word similarity* seperti ditunjukkan pada Tabel 1.

Tabel 1. Pengujian Word Similarity

No	*Input*	*Parameter*	*Process Time*	*Word Similarity Testing* (Kata 'Algorithms')
1	41.000 artikel ilmiah	n=5 *epoch*=100 *windows size*=5 *learning rate*=0.01	302.1 detik	[('temporaldifference', 0.9969866871833801), ('pgellaon', 0.9958505630493164), ('wang', 0.9935269355773926), ('qlearning', 0.9928880929946899), ('abc', 0.9918969869613647)]
2	41.000 artikel ilmiah	n=10 *epoch*=100 *windows size*=5 *learning rate*=0.01	354.4 detik	[('procedures', 0.9344404935836792), ('heuristics', 0.8818841576576233), ('schemes', 0.8769133687019348), ('modifications', 0.8762664198875427), ('alternatives', 0.8749602437019348)]
3	41.000 artikel ilmiah	n=100 *epoch*=100 *window size*=5 *learning rate*=0.01	633.5 detik	[('methods', 0.8426182270050049), ('techniques', 0.8013937473297119), ('approaches', 0.7577121257781982), ('schemes', 0.7043893337249756), ('procedures', 0.6822121739387512)]
4	41.000 artikel ilmiah	n=150 *epoch*=100 *window size*=3 *learning rate*=0.01	849.55 detik	[('methods', 0.8489176630973816), ('techniques', 0.7978703379631042), ('approaches', 0.7638744711875916), ('algorithm', 0.7226699590682983), ('schemes', 0.7089555263519287)]
5	41.000 artikel ilmiah	n=150, *epoch*=100 *windows size*=10 *learning rate*=0.01	1344.641 detik	[('methods', 0.8131347298622131), ('techniques', 0.7598215341567993), ('approaches', 0.7156597971916199), ('schemes', 0.6837905645370483), ('procedures', 0.651324987411499)]
6	41.000 artikel ilmiah	n=150 *epoch*=100 *windows size*=15 *learning rate*=0.01	1273.45 detik	[('methods', 0.6661826372146606), ('techniques', 0.6186638474464417), ('heuristics', 0.5684294104576111), ('variants', 0.5675731301307678), ('algorithm', 0.5607107877731323)]

Tabel 1 menunjukkan hasil pengujian untuk mencari kata yang mirip dengan kata '*algorithm'* sebanyak 5 kata paling mirip. Berdasarkan hasil pengujian parameter pada Tabel 1, didapatkan hasil yang baik setelah percobaan diatas tiga kali. Hasil pengujian cukup baik menunjukkan bahwa pembelajaran dan pengujian sesuai dengan pengertian bahasa yang digunakan manusia sehari-hari.

Pengujian *Document Similarity*

Pada pengujian *document similarity*, untuk memroses setiap dokumen tersebut memerlukan waktu yang cukup lama. Pada proses pelatihan sebanyak 41.000 artikel ilmiah memerlukan waktu sekitar 7 menit untuk menghasilkan rekomendasi.

Dokumen yang menjadi data uji yakni suatu *summary* dari salah satu artikel ilmiah data uji, sebagai berikut: *"We propose an architecture for VQA which utilizes recurrent layers to\ngenerate visual and textual attention. The memory characteristic of the\nproposed recurrent attention units offers a rich joint embedding of visual and\ntextual features and enables the model to reason relations between several\nparts of the image and question. Our single model outperforms the first place\nwinner on the VQA 1.0 dataset, performs within margin to the current\nstate-of-the-art ensemble model. We also experiment with replacing attention\nmechanisms in other state-of-the- art models with our implementation and show\nincreased accuracy. In both cases, our recurrent attention mechanism improves\nperformance in tasks requiring sequential or relational reasoning on the VQA\ndataset"*. Hasil pengujian untuk mencari kesamaan dokumen seperti ditunjukkan pada Tabel 2.

Tabel 2. Pengujian *Document Similarity*

Urutan	*Scores*	Isi Dokumen
1	0.8707674	*"Models based on deep convolutional networks have dominated recent image\ninterpretation tasks; we investigate whether models which are also recurrent temporally deep, are effective for tasks involving sequences, visual and\notherwise. We develop a novel recurrent convolutional architecture suitable for\nlarge-scale visual learning which is end-to-end trainable, and demonstrate the\nvalue of these models on benchmark video recognition tasks, image description\nand retrieval problems, and video narration challenges. In contrast to current\nmodels which assume a fixed spatio-temporal receptive field or simple temporal\naveraging for sequential processing, recurrent convolutional models are doubly\ndeep in that they can be compositional in spatial and temporal layers . Such models may have advantages when target concepts are complex and/or training\ndata are limited. Learning long-term dependencies is possible when\nnonlinearities are incorporated into the network state updates. Long- term RNN\nmodels are appealing in that they directlycan map variable-length inputs\n(e.g., video frames) to variable length outputs (e.g., natural language text)\nand can model complex temporal dynamics"*
2	0.86729705	*"In this paper, we propose a sequential neural encoder with latent structured\ndescription (SNELSD) for modeling sentences. This model introduces latent\nchunk-level representations into conventional sequential neural encoders, i.e.,\nrecurrent neural networks (RNNs) with long short-term memory (LSTM) units, to\nconsider the compositionality of languages in semantic modeling. An SNELSD\nmodel has a hierarchical structure that includes a detection layer and a\ndescription layer. The detection layer predicts the boundaries of latent word\nchunks in an input sentence and derives a chunk-level vector for each word. The\ndescription layer utilizes modified LSTM units to process these chunk-level\nvectors in a recurrent manner and produces sequential encoding outputs. These\noutput vectors are further concatenated with word vectors or the outputs of a\nchain LSTM encoder to obtain the final sentence representation. All the model\nparameters are learned in an end-to-end*

Urutan	*Scores*	Isi Dokumen
		manner without a dependency on\nadditional text chunking or syntax parsing. A natural language inference (NLI)\ntask and a sentiment analysis (SA) task are adopted to evaluate the performance\nof our proposed model. The experimental results demonstrate the effectiveness\nof the proposed SNELSD model on exploring task-dependent chunking patterns\nduring the semantic modeling of sentences. Furthermore, the proposed method\nachieves better performance than conventional chain LSTMs and tree-structured\nLSTMs on both tasks.",
3	0.86614543	*"Recurrent Neural Networks (RNNs), which are a powerful scheme for modeling\ntemporal and sequential data need to capture long-term dependencies on datasets\nand represent them in hidden layers with a powerful model to capture more\ninformation from inputs. For modeling long-term dependencies in a dataset, the\ngating mechanism concept can help RNNs remember and forget previous\ninformation. Representing the hidden layers of an RNN with more expressive\noperations (i.e., tensor products) helps it learn a more complex relationship\nbetween the current input and the previous hidden layer information. These\nideas can generally improve RNN performances. In this paper, we proposed a\nnovel RNN architecture that combine the concepts of gating mechanism and the\ntensor product into a single model. By combining these two concepts into a\nsingle RNN, our proposed models learn long-term dependencies by modeling with\ngating units and obtain more expressive and direct interaction between input\nand hidden layers using a tensor product on 3-dimensional array (tensor) weight\nparameters. We use Long Short Term Memory (LSTM) RNN and Gated Recurrent Unit\n(GRU) RNN and combine them with a tensor product inside their formulations. Our\nproposed RNNs, which are called a Long-Short Term Memory Recurrent Neural\nTensor Network (LSTMRNTN) and Gated Recurrent Unit Recurrent Neural Tensor\nNetwork (GRURNTN), are made by combining the LSTM and GRU RNN models with the\ntensor product. We conducted experiments with our proposed models on word-level\nand character- level language modeling tasks and revealed that our proposed\nmodels significantly improved their performance compared to our baseline\nmodels."*

KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, didapatkan kesimpulan bahwa metode *Word2Vec* dapat digunakan untuk menghasilkan dokumen rekomendasi artikel ilmiah sesuai bidang keminatan pengguna. Dengan menggunakan data latih sebanyak 41.000 artikel ilmiah bersumber dari arXiv.org menghasilkan nilai parameter terbaik n=150, *epoch*=100, *window size*=3 dan *learning rate*=0.01. Terkait penerapan sistem dapat digunakan untuk *preprocessing cache* dimana hasil *document similarity* sudah diproses sebelumnya dan disimpan dalam *database,* sehingga pengguna tidak harus melakukan proses tersebut. Pengguna akan menerima hasil rekomendasi karya ilmiah yang sesuai dengan bida keminatan pengguna, tanpa memerluka waktu lama untuk melakukan serangkaian prosesnya.

DAFTAR PUSTAKA

[1] G. Mckiernan, “ArXiv.org: The Los Alamos National Laboratory E-print Server,” vol. 1, no.

3, pp. 127–138, 2000.

[2] A. Boldt, "Extending ArXiv.org to Achieve Open Peer Review and Publishing," *J. Sch. Publ.*, vol. 42, no. 2, pp. 238–242, 2011, doi: 10.3138/jsp.42.2.238.

[3] E. Erlangga and H. Sutrisno, "Sistem Rekomendasi Beauty Shop Berbasis Collaborative Filtering," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 10, no. 2, p. 47, 2020, doi: 10.36448/jmsit.v10i2.1611.

[4] T. Badriyah, R. Fernando, and I. Syarif, "Sistem Rekomendasi Content Based Filtering Menggunakan Algoritma Apriori," *Konf. Nas. Sist. Inf.*, vol. 1, no. 1, pp. 554–559, 2018.

[5] N. R. Ramadhanti and S. Mariyah, "Document Similarity Detection Using Indonesian Language Word2vec Model," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, pp. 1–6, 2019, doi: 10.1109/ICICoS48119.2019.8982432.

[6] M. Sahbuddin and S. Agustian, "Support Vector Machine Method with Word2vec for Covid-19 Vaccine Sentiment Classification on Twitter," *J. Informatics Telecommun. Eng.*, vol. 6, no. 1, pp. 288–297, 2022, doi: 10.31289/jite.v6i1.7534.

[7] N. C. Wibowo, D. Satria, Y. Kartika, and S. R. Wardhana, "Pengkategorian Berita Online Secara Otomatis Menggunakan Metode PLSA," *KURVATEK*, vol. 3, no. 1, pp. 45–51, 2018.

[8] M. Z. Rahman, Y. A. Sari, and N. Yudistira, "Analisis Sentimen Tweet COVID-19 Menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 5120–5127, 2021, [Online]. Available: http://j-ptiik.ub.ac.id.

[9] S. M. N. Tannady, D. H. Setiabudi, and ..., "Penerapan Long-Short Term Memory dengan Word2Vec Model untuk Mendeteksi Hoax dan Clickbait News pada Berita Online di Indonesia," *J. Infra*, vol. 2021, 2022, [Online]. Available: https://publication.petra.ac.id/index.php/teknik-informatika/article/view/12516%0Ahttps://publication.petra.ac.id/index.php/teknik-informatika/article/download/12516/10817.

[10] A. M. Rizki, A. L. Nurlaili, F. P. Aditiawan, and G. E. Yuliastuti, "Forecasting the Inflation Rate in Indonesia Using Backpropagation Artificial Neural Network," in *2022 IEEE 8th Information Technology International Seminar (ITIS)*, 2022, pp. 1–5.

[11] A. M. Rizki, G. E. Yuliastuti, E. Y. Puspaningrum, and A. Lina, "Klasifikasi Jenis Kelamin Berdasarkan Ciri Fisik Menggunakan Algoritma Neural Network," *SCAN - J. Teknol. Inf. dan Komun.*, vol. XVII, no. 3, pp. 18–22, 2022.

[12] A. M. Rizki, H. Maulana, D. S. Y. Kartika, and G. E. Yuliastuti, "Classification Of Sexually Transmitted Infectional Diseases Using Artificial Neural Networks," *J. Mantik*, vol. 5, no. 36, pp. 1759–1765, 2021.

[13] S. A. Savittri, A. Amalia, and M. A. Budiman, "A Relevant Document Search System Model using Word2vec Approaches," *J. Phys. Conf. Ser.*, vol. 1898, no. 1, 2021, doi: 10.1088/1742-6596/1898/1/012008.