Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JACOST**

JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)

Vol. 5 No. 1 (2024) 56 – 62 | ISSN: 2723-1453 (Media Online)

Deteksi Clickbait pada Judul Berita Online Berbahasa Indonesia Menggunakan FastText

Muhaza Liebenlito[1], Arlianis Arum Yesinta[2], Muhamad Irvan Septiar Musti[3]
[1,2,3]Program Studi Matematika, Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta, Indonesia
[1]muhazaliebenlito@uinjkt.ac.id, [2]arlianis.arum18@mhs.uinjkt.ac.id, [3]muhamad.musti@uinjkt.ac.id

Abstract

The rise of people accessing news portals has created intense competition between online media to get readers or visitors to maximize their revenue. This is what triggers the development of clickbait. Clickbait can reduce the quality of the news itself, and it also has the potential to be misinformation regarding to news contents as known as fake news. Therefore, it is necessary to detect news titles that contain clickbait. This study aims to obtain an optimal clickbait news title classification model using FastText. To get the optimal model can be done by cleaning the data and optimizing the model's hyperparameters. The model was trained using 9600 training data collected from Indonesian online news. The best model obtained in this study has performance with an accuracy of 77% and an F1-Score of 69%.

Keywords: FastText Classifier, text classification, online news, clickbait

Abstrak

Maraknya masyarakat yang mengakses portal berita menimbulkan persaingan ketat antar media untuk mendapatkan pembaca atau pengunjung sebagai sarana untuk memaksimalkan pendapatan. Hal inilah yang memicu berkembangnya *clickbait*. *Clickbait* merupakan berita yang mengandung judul dengan bahasa memikat tujuan untuk menarik perhatian pembaca agar mengklik judul tersebut. Potongan informasi dari judul berita yang didapatkan dapat memicu kesalahpahaman apabila pembaca hanya sekedar membaca judulnya saja tanpa memeriksa kembali isi konten secara menyeluruh. Sehingga perlu adanya pendeteksian terhadap judul-judul berita yang mengandung clickbait. Oleh karena itu, tujuan penelitian ini adalah untuk memperoleh model klasifikasi judul berita *clickbait* yang optimal dengan menggunakan *FastText*. Untuk mendapatkan model yang optimal dapat dilakukan dengan membersihkan data dan mengatur beberapa parameter dalam model. Model dilatih dengan menggunakan 9.600 data latih yang diambil dari berita online Indonesia. Model terbaik yang didapatkan pada penelitian ini memiliki performa dengan akurasi sebesar 77% dan *F1-Score* sebesar 69%.

Kata kunci: FastText, klasifikasi teks, berita daring, Bahasa Indonesia

1. Pendahuluan

Maraknya masyarakat yang mengakses portal berita menimbulkan persaingan ketat antar media untuk mendapatkan pembaca atau pengunjung. Hal ini yang mendorong meningkatnya *clickbait*. *Clickbait* merupakan berita dengan judul melebih-lebihkan isi konten dengan tujuan untuk menarik perhatian pembaca agar mengklik judul tersebut [1]. Salah satu tujuan *clickbait* adalah untuk mencari pendapatan dengan meningkatkan *traffic* pembaca atau pengunjung. Semakin banyak pengunjung situs maka semakin banyak pula pendapatan yang didapatkan. Saat ini portal-portal berita lebih mementingkan jumlah klik dibandingkan kualitas berita itu sendiri [2]. *Clickbait* sering dijadikan sebagai daya tarik oleh para penyedia situs untuk menarik minat pembaca yang penasaran dengan judul-judul berita yang dibuat oleh penyedia situs tersebut.

Clickbait dapat berpotensi menjadi informasi yang keliru atau *hoax* [3]. Terlebih lagi, Indonesia termasuk dalam daftar negara dengan tingkat literasi yang rendah berdasarkan survey *Program for International Student Assesment* (PISA) yang dirilis *Organization for Economis Co-operation and Development* (OECD) pada tahun 2019 [4]. Potongan informasi dari judul berita yang didapatkan dapat memicu kesalahpahaman apabila pembaca hanya sekedar membaca judulnya saja tanpa memeriksa kembali isi konten secara menyeluruh.

Berdasarkan masalah *clickbait* tersebut, diperlukan pengklasifikasian judul berita *clickbait* dan *non-clickbait*. Beberapa penelitian terkait pengklasifikasian judul berita *clickbait* dan *non-clickbait* telah dilakukan sebelumnya. Pada tahun 2021, Siregar dkk melakukan klasifikasi *clickbait* dengan menggunakan metode RNN-LSTM dan berhasil mendapatkan akurasi sebesar 83%. Sedangkan modifikasi Bidirectional-LSTM

dengan metode self-attention berhasil memperoleh akurasi sebesar 81% [5]. Selain itu, Fakhruzzaman dkk [6] juga telah melakukan penelitian serupa dengan menggunakan metode M-BERT dengan menggunakan dataset yang telah disediakan secara daring oleh William dan Sari [7]. Penelitian ini berhasil mendapatkan akurasi sebesar 94%, *F1-Score* sebesar 91%, skor presisi sebesar 91%, dan skor ROC-AUC sebesar 92%.

Sedangkan pada tahun 2020, Amalia dkk melakukan klasifikasi teks terhadap artikel berita dengan menggunakan *FastText* dan berhasil mendapatkan *F1-score* sebesar 97% [8]. *FastText* adalah salah satu metode yang dapat digunakan untuk melakukan klasifikasi teks. *FastText* sendiri merupakan sebuah *library* yang dibuat oleh facebook pada tahun 2016 untuk pembelajaran yang lebih efisien terhadap representasi kata dan klasifikasi teks. Klasifikasi teks berbahasa Indonesia masih menjadi tantangan karena keterbatasan dataset yang dimiliki.

Berdasarkan uraian diatas, peneliti ingin melakukan penelitian terkait klasifikasi judul berita *clickbait* dan *non-clickbait* dengan menggunakan metode *FastText*. Pada penelitian ini *dataset* yang digunakan diambil dari CLICK-ID yang terdiri dari 15.000 judul berita yang telah dianotasi dengan label *clickbait* sebanyak 6.290 dan *non-clickbait* sebanyak 8.710 [7]. Selanjutnya, akan dilakukan EDA (*Exploratory Data Analysis*) terhadap data untuk melihat gambaran data secara keseluruhan. Kemudian, data akan melalui tahap *preprocessing* dan *splitting* hingga akhirnya dilakukan klasifikasi menggunakan metode *FastText*. Model yang telah dibuat diharapkan dapat bermanfaat dalam mendeteksi judul berita *clickbait* dan *non-clickbait* sehingga dapat dijadikan pengetahuan tambahan dalam pencegahan *clickbait* kedepannya.

2. Metode Penelitian

Pada penelitian ini data yang digunakan merupakan data sekunder yang diperoleh dari [7]. *Dataset* tersebut diambil dari 12 situs berita yaitu kompas.com, liputan6.com, okezone.com, posmetro.com, detiknews.com, fimela.com, kapanlagi.com, republika.co.id, sindonews.com, tempo.com, tribunnews.com, dan wowkeren.com dengan jumlah 15.000 judul berita yang telah dilabeli. Pelabelan dilakukan oleh 3 mahasiswa menggunakan sistem *voting* dimana *voting* terbanyak yang dijadikan sebagai label. Sehingga diperoleh judul berita dengan label *clickbait* sebanyak 6.290 dan *non-clickbait* sebanyak 8.710.

Dataset tersebut memiliki sembilan kolom yaitu *title*, *source*, *date*, *time*, *category*, *sub-category*, *content*, *url*, dan label. Namun, kolom yang digunakan dalam penelitian ini hanya empat yakni *title* kolom ini berisi judul berita, *source* kolom yang berisikan sumber berita tersebut, *category* kolom yang memuat kategori berita, *label* kolom ini berisikan label dari judul berita yakni *clickbait* atau *non-clickbait*.

2.1. *Preprocessing*

Preprocessing yang dilakukan dalam penelitian ini antara lain, *case folding*, penghapusan simbol dan angka, *lemmatizing*, dan penghapusan *stopwords*. Setelah dilakukan *preprocessing*, data akan memiliki bentuk dan format yang lebih terstruktur sehingga siap untuk dilanjutkan ke tahap selanjutnya.

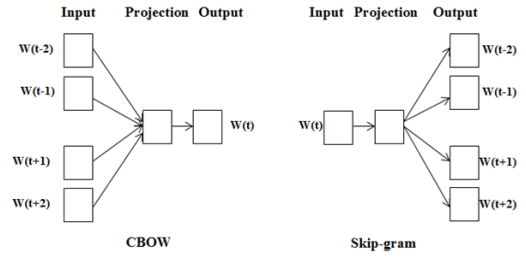
Bentuk data akan disesuaikan terlebih dahulu dengan model yang akan digunakan yaitu *FastText*. *FastText* memiliki format data tertentu yaitu *__label__*<*label kelas*> <*teks*> dimana *__label__* merupakan prefix dari label kelas, <*label kelas*> adalah label dari kalimat, dan <*teks*> merupakan kalimat itu sendiri. Setelah data memiliki format yang sesuai, kemudian data akan disimpan dalam file dengan ekstensi *.txt* yang selanjutnya siap untuk dilakukan pemodelan.

2.2. *FastText*

FastText merupakan sebuah *library* yang dikembangkan oleh tim peneliti *Facebook AI Research* (FAIR) untuk pembelajaran yang efektif terhadap representasi kata dan klasifikasi teks [9], [10]. Agar sebuah mesin dapat memahami dan melakukan pemrosesan pada data teks, maka sebuah teks harus direpresentasikan kedalam sebuah vektor. Representasi vektor kata yang tepat merupakan hal yang diperlukan untuk menghasilkan akurasi yang baik dalam teks klasifikasi [11].

FastText untuk representasi kata sendiri merupakan pengembangan dari *Word2Vec*. Namun, tidak seperti *Word2Vec* yang menggunakan kata sebagai bagian terkecil dalam menghasilkan representasi kata, *FastText* memiliki *feature bag of n-grams* yang bekerja pada level karakter untuk menghasilkan representasi kata. *N-grams* pada *FastText* merepresentasikan setiap kata menjadi bagian yang lebih kecil lagi sesuai dengan n yang dipilih dan menyertakan kata itu sendiri [10]. Misalnya, kata ‘pintu’ dengan maka akan menjadi <pi, pin, int, ntu, tu> dan <pintu> dimana ‘<’ dan ‘>’ merupakan simbol tambahan yang digunakan untuk membedakan suatu kata dengan kata itu sendiri dan memungkinkan untuk memahami prefix dan sufix dari suatu kata. Kemudian, untuk mendapatkan vektor kata, *FastText* menjumlahkan setiap vektor representasi dari karakter *n-grams*. Dalam *training unsupervised FastText* memiliki dua model yang dapat dipilih yaitu CBOW ataupun *Skipgram*. *Continous Bag of Words* (CBOW) adalah metode yang bertujuan memprediksi kata target berdasarkan konteks di sekitarnya, sementara *Skipgram* memprediksi konteks di sekitar kata target tertentu. Perbedaan CBOW dan *Skipgram* dapat dilihat pada Gambar 1.

DOI : https://doi.org/10.52158/jacost.v5i1.655



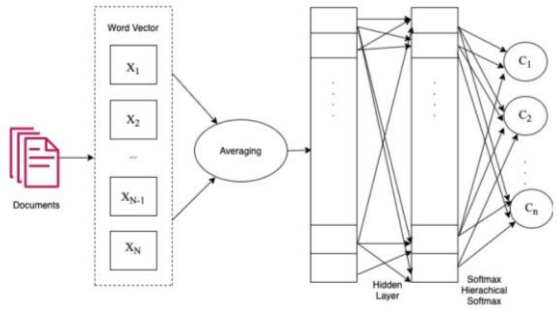
Gambar 1. Perbedaan CBOW dan *Skipgram*

Secara *default FaxtText* menggunakan *Skipgram*. Pada penelitian ini *FastText* digunakan untuk merepresentasikan kata dimana *FastText* merupakan sebuah model yang menggunakan metode *Skipgram* [12]. Ide utama di balik *Skipgram* adalah untuk memprediksi konteks kata-kata yang mungkin muncul di sekitar kata target tertentu. Formulasi *Skipgram* dapat didefinisikan sebagai $p(w_{t+j}|w_t)$menggunakan fungsi *softmax*, Persamaan 1:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^T v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_{w_O}}^T v_{w_I}\right)} \quad (1)$$

Dimana v_w dan v'_w adalah *input* dan *output* vektor representasi dari w, dan W adalah jumlah kata dalam kamus. *FastText* memiliki *feature* tambahan yaitu *character n-grams*. Namun, pada penelitian ini peneliti tidak menggunakan *feature* tersebut. Vektor representasi kata yang dihasilkan pada penelitian ini berdimensi 100.

Selain representasi kata, *FastText* juga dapat digunakan untuk melakukan klasifikasi teks. Beberapa penelitian telah membuktikan bahwa *FastText* memiliki performa yang cukup baik untuk mengklasifikasikan sebuah dokumen. Seperti penelitian [8], [12], model *FastText* memiliki performa yang paling baik dari model-model pembanding dalam penelitian tersebut.



Gambar 2. Arsitektur *FastText*

Gambar 2 merupakan arsitektur pada *FastText*. Arsitektur *FastText* terdiri dari tiga komponen utama yaitu *input layer*, *hidden layer*, dan *output layer*.

Setiap kata w_i pada dokumen direpresentasikan sebagai vektor kata x_i menggunakan representasi kata yang telah terbentuk. Selanjutnya, vektor kata yang terbentuk akan dicari rata-ratanya dengan membagi semua x_i seperti pada Persamaan 2

$$y = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2)$$

Rata-rata inilah yang disebut dengan vektor dokumen y yang kemudian akan dijadikan input di hidden layer. Pada layer ini vektor dokumen y akan dikalikan dengan matriks B untuk mendapatkan vektor klasifikasi z seperti yang dapat dilihat pada Persamaan 3, dimana y merupakan vektor berdimensi n, z adalah vektor berdimensi m, B matriks berukuran $m \times n$, dan m merupakan banyaknya label kelas. Selanjutnya, klasifikasi dilakukan dengan menghitung probabilitas kelas menggunakan fungsi softmax pada Persamaan 4

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = B.y \quad (3)$$

$$p_j = \frac{e^{z_j}}{\sum_{k=1}^{m} e^{z_k}} \quad (4)$$

dimana p_j adalah probabilitas sebuah dokumen berada dalam kelas j, z_j dan z_k merupakan komponen dari vektor z.

2.3 Pencarian *Hyperparameter* Terbaik

Penelitian ini menggunakan *k-fold cross validation* untuk mencari *hyperparamter* terbaik. Metode ini melibatkan pembagian dataset menjadi k subset yang seukuran. Selanjutnya, model dilatih dan dievaluasi sebanyak k kali. Pada setiap iterasi, satu subset diambil sebagai data uji sedangkan $k-1$ subset lainnya digunakan sebagai data latih. Model dilatih pada data latih dan diujikan pada data uji untuk mengukur kinerjanya. Proses ini diulang sebanyak k iterasi, sehingga setiap subset menjadi data uji satu kali. Hasil evaluasi dari tiap iterasi diambil rata- ratanya, memberikan gambaran komprehensif tentang kemampuan model dalam menggeneralisasi data yang tidak pernah dilihat sebelumnya [13].

Pendekatan ini membantu dalam mengidentifikasi *overfitting* atau *underfitting*, serta memaksimalkan penggunaan dataset untuk pelatihan dan pengujian. Meskipun memakan waktu lebih lama, *k-fold cross validation* memberikan evaluasi yang lebih kredibel dan dapat diandalkan terhadap performa model *machine learning*.

2.4. Evaluasi Model

Setelah mendapatkan model terbaik dengan melakukan konfigurasi terhadap *hyperparameter*, tahap selanjutnya adalah mengevaluasi model tersebut dengan data test. Metode evaluasi yang digunakan adalah *Confusion Matrix*. Metode ini merupakan metode evaluasi yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi. *Confusion Matrix* adalah tabel dengan empat kombinasi berbeda dari nilai prediksi dan nilai

DOI : https://doi.org/10.52158/jacost.v5i1.655

aktual. Pada Tabel 1, TP (*True Positive*) yaitu dimana model memprediksi data ada dikelas positif dan yang sebenarnya data berada dikelas positif, TN (*True Negative*) dimana model memprediksi data ada dikelas negatif dan yang sebenarnya data berada dikelas negatif, FP (*False Positive*) yaitu model memprediksi data ada di kelas positif namun sebenarnya data berada di kelas negatif, FN (*False Negative*) yaitu model memprediksi data berada di kelas negatif, namun sebenarnya data berada di kelas positif.

Tabel 1.*Confusion Matrix*

Aktual	Prediksi	
	Nilai Positif	Nilai Negatif
Nilai Positif	TP	FN
Nilai Negatif	FP	TN

Evaluasi model klasifikasi dapat dilihat dengan menghitung nilai akurasi, presisi, *recall*, dan *F1-Score*. Akurasi adalah persentase data yang terprediksi benar yang dapat dihitung dengan Persamaan 5.

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \quad (5)$$

Presisi digunakan untuk menghitung perbandingan prediksi positif yang benar dengan jumlah keseluruhan prediksi positif yang dapat dihitung dengan Persamaan 6.

$$Presisi = \frac{TP}{TP + FP} \quad (6)$$

Sedangkan *recall* merupakan perbandingan prediksi positif yang benar dengan jumlah keseluruhan data positif yang dapat dihitung menggunakan Persamaan 7.

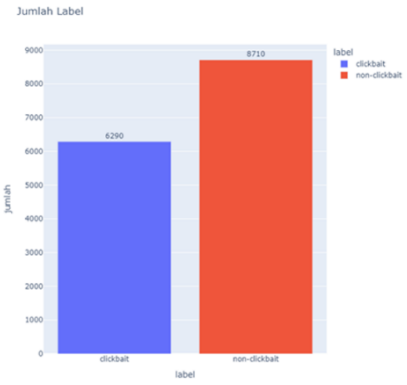
$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Persamaan 8 digunakan untuk menghitung *F1-Score*.

$$F1Score = 2 \times \frac{presisi \times recall}{presisi + recall} \quad (8)$$

3. Hasil dan Pembahasan

3.1. Analisa Data Eksploratif



Gambar 3. Jumlah label pada data

Gambar 4. (a) Wordcloud unigram keseluruhan data dan (b) Wordcloud bigram keseluruhan data

Berdasarkan Gambar 4 dapat dilihat bahwa sebagian besar judul berita yang terdapat didalam keseluruhan dataset memiliki topik yang berkaitan dengan ‘kpk’, ‘habibie’, ‘indonesia’, ‘jokowi’, ‘anak’, dan ‘polisi’. Jika menggunakan bigram, maka kata yang sering muncul adalah ‘bj habibie’, ‘revisi uu’, ‘uu kpk’, dan ‘kabut asap’. Kata yang muncul mengerucut pada satu topik yaitu pemerintahan. Sehingga terindikasi bahwa topik berita yang sering dibuat oleh media massa berkenaan dengan topik pemerintahan dan politik. Terlihat juga pembicaraan tentang presiden ketiga Republik Indonesia yaitu B.J. Habibie menjadi salah satu tokoh yang banyak dijadikan berita hal ini dikarenakan dataset yang diambil merupakan judul berita pada tahun 2019 dimana B.J. Habibie mengalami sakit dan meninggal dunia pada tahun tersebut tepatnya tanggal 11 September 2019.

Gambar 5. (a) Wordcloud unigram pada data clickbait dan (b) Wordcloud bigram pada data clickbait

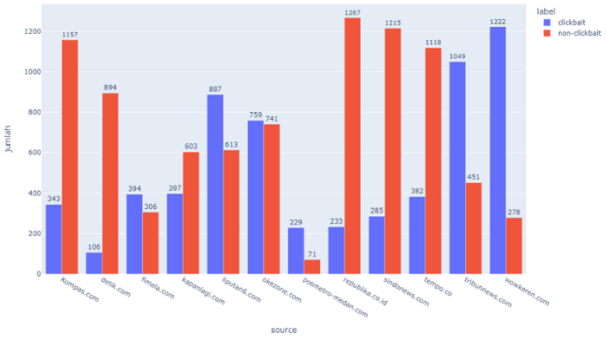
Jika dilihat berdasarkan label *clickbait* pada Gambar 5, kata yang sering muncul jika menggunakan unigram adalah ‘habibie’, ‘indonesia’, ‘kpk’, ‘bikin’, dan ‘anak’. Namun, jika menggunakan bigram kata yang mendominasi adalah ‘bj habibie’, ‘timnas indonesia’, ‘revisi uu’, dan terdapat juga nama-nama selebriti Indonesia seperti ‘bebby fey’ dan ‘barbie kumalasari’. Kata-kata tersebut masih berkaitan dengan topik-topik pemerintahan, bahkan judul berita terkait B.J. Habibie pun juga masih menjadi perbincangan yang sering menjadi topik utama. Selain itu, terdapat pula nama-nama selebritis yang mengindikasikan bahwa judul berita berlabel *clickbait* juga banyak membahas seputar dunia hiburan.

Jika dilihat berdasarkan label *non-clickbait* pada Gambar 6, kata yang sering muncul jika menggunaka unigram adalah ‘indonesia’, ‘kpk’, ‘habibie’, dan ‘jokowi’. Namun, jika menggunakan bigram kata yang mendominasi adalah ‘bj habibie’, ‘uu kpk’, ‘revisi uu’, dan ‘kabut asap’.

DOI : https://doi.org/10.52158/jacost.v5i1.655

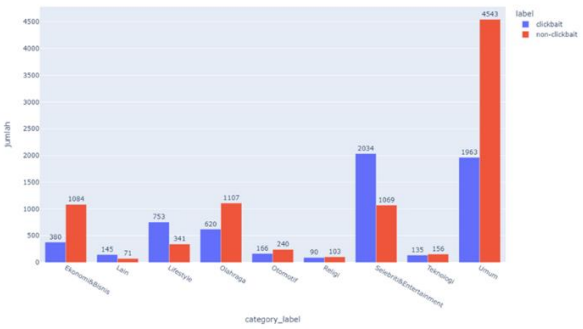
Gambar 6. (a) Wordcloud unigram pada data non-clickbait dan (b) Wordcloud bigram pada data non-clickbait

Dalam judul berita berlabel non-clickbait juga topik-topik yang sering dibicarakan adalah topik-topik terkait pemerintahan. Selain B.J Habibie, topik lain yang menjadi perbincangan diantaranya ada isu-isu terkait revisi UU dan kabut asap. Hal ini dikarenakan pada tahun 2019, terjadi kebakaran hutan di Provinsi Riau.



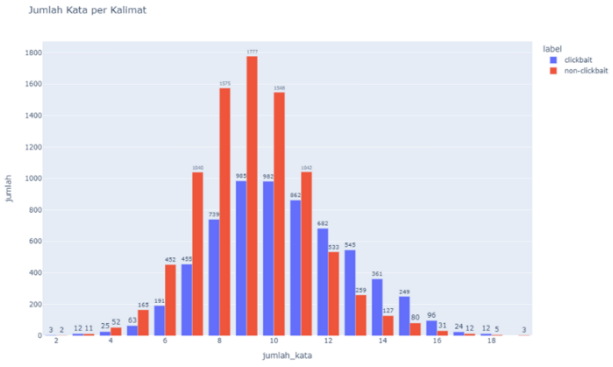
Gambar 7. Jumlah label berdasarkan sumber berita

Berdasarkan Gambar 7 dapat dilihat dari 12 sumber berita pada penelitian ini, ada lima sumber berita yang didominasi oleh judul yang mengandung *clickbait* yaitu fimela.com, liputan6.com, okezone.com, tribunnews.com, dan wowkeren.com.



Gambar 8. Jumlah label berdasarkan kategori berita

Penelitian ini menggunakan 12 sumber berita dimana masing-masing sumber memiliki kategori berita yang berbeda. Oleh karena itu, kategori-kategori tersebut dikelompokkan kembali menjadi 9 kategori yaitu 'Teknologi' kategori ini berisi judul-judul berita yang berkaitan dengan teknologi, 'Selebriti&Entertainment' kategori memuat judul-judul berita yang berkaitan dengan artis dan dunia hiburan, 'Religi' mengkategorikan judul berita yang berkaitan dengan keagamaan, 'Otomotif' kategori yang memuat judul berita terkait dunia otomotif, 'Olahraga' kategori yang berisi judul-judul berita terkait dengan olahraga seperti sepak bola dan lain sebagainya, 'Lifestyle' mengkategorikan judul berita yang terkait dengan gaya hidup, 'Ekonomi&Bisnis' kategori yang memuat judul berita terkait ekonomi, keuangan, dan sebagainya, 'Umum' mengkategorikan judul berita terkait isu-isu pemerintahan, peristiwa dalam negeri ataupun luar negeri dan lain sebagainya, 'Lain' kategori yang berisi judul berita diluar 8 kategori sebelumnya seperti kisah-kisah. Berdasarkan Gambar 8, kategori yang banyak mengandung *clickbait* adalah 'lifestyle', 'selebriti&entertainment', dan kategori 'lain'.



Gambar 9. Jumlah kata yang membentuk setiap judul

Pada Gambar 9 minimal kata yang membentuk sebuah judul adalah 2. Judul dengan jumlah kata 12 keatas lebih didominasi oleh judul berita dengan label *clickbait*. Hal ini berarti judul berita berlabel *clickbait* cenderung menggunakan jumlah kata yang lebih banyak dibandingkan dengan label *non-clickbait*.

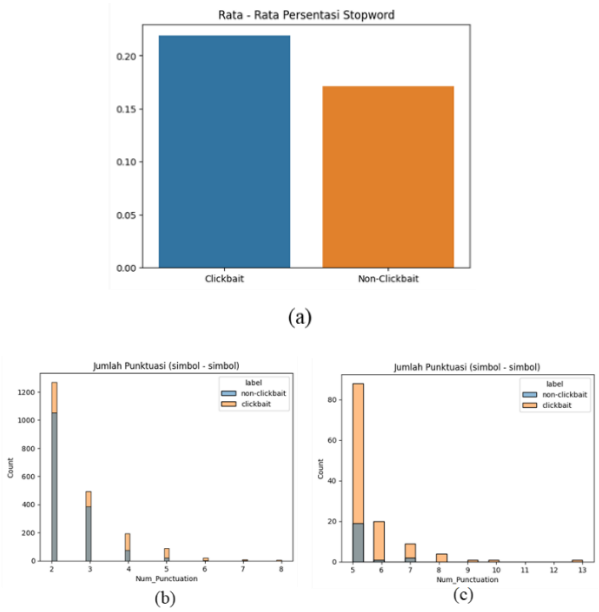
Selain itu, dari Gambar 10 (a) judul berita dengan label *clickbait* juga memiliki persentase *stopwords* yang lebih banyak jika dibandingkan dengan label *non-clcikbait*. Oleh karena itu, *stopwords* diduga memiliki pengaruh terhadap performa model. Sehingga, akan dilakukan pelatihan model dengan menghilangkan *stopwords* dan tanpa menghilangkan *stopwords*. Gambar 10 (b) menunjukkan jumlah punktuasi dalam judul dengan rentang 2-8 dimana untuk jumlah punktuasi 2 dan 3 didominasi oleh judul berlabel *non-clickbait* sedangkan punktuasi berjumlah 4 didominasi oleh judul berlabel *clickbait*. Gambar 10 (b) menunjukkan jumlah punktuasi pada rentang 5-13 dimana pada rentang ini didominasi oleh judul berita berlabel *clickbait*. Hal ini menunjukkan bahwa judul berita berlabel *clickbait* menggunakan punktuasi yang lebih banyak jika dibandingkan judul berita berlabel *non-clickbait*.

3.2. Hasil Pelatihan Model dan Tuning Hyperparameter

Salah satu cara untuk mendapatkan model yang optimal adalah dengan *hyperparameter tuning*. Pada penelitian ini, optimasi *hyperparatemer* dilakukan terhadap ukuran *learning rate*, *wordNgrams*, dan *epoch*. Optimasi *hyperparameter* pada penelitian ini dilakukan dengan memilih ukuran *learning rate* dari nilai yang

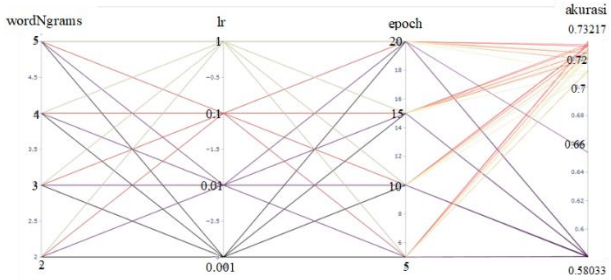
DOI : https://doi.org/10.52158/jacost.v5i1.655

termasuk dalam himpunan {0.001, 0.01, 0.1, 1}, *wordNgrams* diantara nilai yang termasuk dalam himpunan {2, 3, 4, 5}, dan *epoch* diantara nilai yang termasuk dalam himpunan {5, 10, 15, 20}. Selanjutnya, pemodelan dilakukan dengan menggunakan dua data hasil *preprocessing* yang berbeda. *Preprocessing* yang pertama dilakukan dengan menghilangkan *stopwords* dan *preprocessing* kedua dilakukan tanpa menghilangkan *stopwords*.



Gambar 10. (a) Perbandingan *stopwords* (b) Jumlah punktuasi rentang 2-8 (c) Jumlah punktuasi rentang 5-13

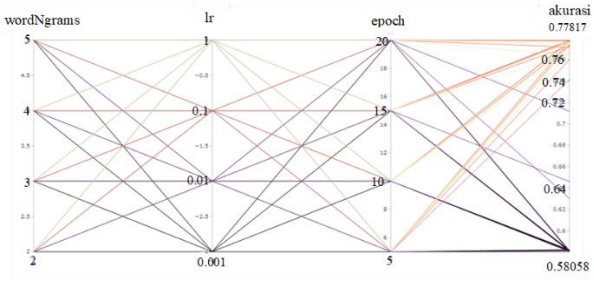
Model pertama adalah model yang dilatih dengan data tanpa *stopwords.* Akurasi terbaik yang diperoleh pada model ini adalah 73,22% ketika *learning rate* bernilai 0.1, *wordNgram* berukuran 2, dan *epoch* berukuran 5. Hasil optimasi pada model ini dapat dilihat pada Gambar 11.



Gambar 11. *Parallel Coordinate Plot* Model 1

Dari Gambar 11 dapat dilihat bahwa ketika nilai *learning rate* 0.001 dan 0.01 akurasi dari model berada dibawah 70%. Namun, ketika nilai learning rate 0.1 dan 1 performa model lebih baik. Ini menunjukkan bahwa semakin besar nilai *learning rate* maka akurasi model semakin baik. Hal ini sejalan dengan nilai *learning rate* yang direkomendasikan oleh *FastText* yaitu berkisar antara $0.1 - 1$. Sedangkan *wordNgrams* dan *epoch* memengaruhi nilai akurasi model secara bebas (tidak memiliki keterkaitan tertentu seperti learning rate).

Sedangkan untuk model kedua yaitu model yang dilatih dengan data yang mengandung *stopwords.* Akurasi terbaik yang diperoleh pada model ini adalah 77,82% ketika *learning rate* bernilai 0.1, *wordNgram* berukuran 4, dan *epoch* berukuran 15. Hasil optimasi pada model ini dapat dilihat pada Gambar 12.



Gambar 12. *Parallel Coordinate Plot* Model 2

Sama seperti model sebelumnya bahwa semakin besar *learning rate* maka performa model semakin baik, sedangkan *wordNgrams* dan *epoch* memengaruhi nilai akurasi model secara bebas.

3.3. Hasil Evaluasi Model Terbaik

Setelah mendapatkan *hyperparameter* terbaik, selanjutnya model akan dievaluasi. Pada tahap evaluasi, model akan di uji untuk memprediksi kelas data. Tujuan evaluasi uji data untuk melihat seberapa besar model dapat dipercaya dalam memprediksi kelas.

Tabel 2.Confusion Matrix Model 1

Aktual	Prediksi	
	Non-clickbait	Clickbait
Non-clickbait	1.427	315
Clickbait	525	733

Hasil pengujian model pertama (tanpa *stopwords*) ditunjukkan pada Tabel 2 dimana judul berita berlabel *non-clickbait* benar diprediksi oleh model sebanyak 1.427 dan model salah memprediksi sebanyak 315. Sedangkan judul berita berlabel *clickbait* benar diprediksi oleh model sebanyak 733 dan model salah memprediksi sebanyak 525.

Tabel 3. Confusion Matrix Model 2

Aktual	Prediksi	
	Non-clickbait	Clickbait
Non-clickbait	1.509	233
Clickbait	472	786

Hasil pengujian model kedua (dengan *stopwords*) ditunjukkan pada Tabel 3 dimana judul berita berlabel *non-clickbait* benar diprediksi oleh model sebanyak 1.509 dan model salah memprediksi sebanyak 233. Sedangkan judul berita berlabel *clickbait* benar diprediksi oleh model sebanyak 786 dan model salah memprediksi sebanyak 472.

Berdasarkan Tabel 4 perbandingan model tanpa menghilangkan *stopwords* menaikkan nilai presisi, *recall*, *F1-Score*, dan akurasi pada model. Hal ini

DOI : https://doi.org/10.52158/jacost.v5i1.655

menunjukkan bahwa keberadaan *stopwords* memiliki pengaruh terhadap performa model klasifikasi *clickbait*.

Tabel 4.Hasil Pengujian Setiap Model

Preprocessing	Label	Presisi	*Recall*	*F1-Score*	Akurasi
Tanpa *Stopwords*	Non-clickbait	73%	82%	77%	72%
	Clickbait	70%	58%	64%	
Dengan *Stopwords*	Non-clickbait	76%	87%	81%	77%
	Clickbait	77%	62%	69%	

Sehingga, didapatkan model terbaik adalah model yang masih mengandung *stopwords* dengan nilai akurasi sebesar 77%, nilai presisi sebesar 77%, nilai *recall* sebesar 62%, dan *F1-Score* sebesar 69%. Hal ini berarti bahwa model dapat mengklasifikasikan 77% data dengan benar. Nilai presisi pada kelas *clickbait* sebesar 77% artinya model mampu memprediksi 77% judul berita berlabel *clickbait* dengan benar dari keseluruhan judul berita *clickbait* yang diprediksi, nilai recall sebesar 62% artinya model mampu memprediksi 62% judul berita *clickbait* dengan benar dari keseluruhan judul berita yang benar *clickbait*. Nilai presisi pada kelas *non-clickbait* sebesar 76% artinya model mampu memprediksi 76% judul berita *non-clcikbait* dengan benar dari keseluruhan judul berita *non-clickbait* yang diprediksi, nilai recall sebesar 87% artinya model dapat memprediksi 87% judul berita *non-clickbait* dengan benar dari keseluruhan judul berita yang benar *non-clickbait*.

4. Kesimpulan

Judul berita yang digunakan dalam penelitian ini berjumlah 15.000 dengan jumlah judul berlabel *non-clickbait* sebanyak 8.710 data dan *clickbait* sebanyak 6.290. Penelitian ini menggunakan *FastText* dengan dua *preprocessing* yang berbeda yaitu dengan menghilangkan *stopwords* dan tanpa menghilangkan *stopwords*. Model terbaik didapatkan dengan melatih model menggunakan *dataset* tanpa menghilangkan *stopwords* karena keberadaan *stopwords* terbukti memiliki pengaruh yang cukup signifikan terhadap performa model. Selain itu penggunaan nilai *hyperparameter* yang menghasilkan model terbaik adalah *learning rate* dengan nilai 0.1, *wordNgrams* dengan ukuran 4, dan *epoch* berukuran 15. Kombinasi ini menghasilkan model terbaik dengan akurasi sebesar 77% dan *F1-Score* sebesar 69%, yang berarti bahwa model berhasil mengklasifikasikan 77% data dengan benar, dengan rata-rata klasifikasi benar pada label *clickbait* adalah 69% data dari label tersebut. Untuk penelitian selanjutnya dapat dikembangkan beberapa hal yaitu menggunakan fitur *character n-gram* pada *FastText* untuk menangani OOV (*Out of Vocabulary*) dan menerapkan *multiword expression* untuk mengatasi fenomena *stopwords*. Selain itu, penelitian selanjutnya diharapkan dapat menambahkan data judul berita berlabel *clickbait*.

Daftar Rujukan

[1] N. Rahmatika and G. Prisanto, "Pengaruh Berita Clickbait Terhadap Kepercayaan pada Media di Era Attention Economy," *Avant Garde*, vol. 10, no. 2, Art. no. 2, Dec. 2022, doi: 10.36080/ag.v10i2.1947.

[2] Y. Chen, N. K. Conroy, and V. L. Rubin, "News in an online world: The need for an 'automatic crap detector,'" *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015, doi: 10.1002/pra2.2015.145052010081.

[3] Y. Chen, N. J. Conroy, and V. L. Rubin, "Misleading Online Content: Recognizing Clickbait as 'False News,'" in *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, in WMDD '15. New York, NY, USA: Association for Computing Machinery, Nov. 2015, pp. 15–19. doi: 10.1145/2823465.2823467.

[4] "Tingkat Literasi Indonesia di Dunia Rendah, Ranking 62 Dari 70 Negara." Accessed: Sep. 09, 2023. [Online]. Available: https://perpustakaan.kemendagri.go.id/2021/03/tingkat-literasi-indonesia-di-dunia-rendah-ranking-62-dari-70-negara/

[5] M. Liebenlito, Ivansyah, M. Y. Wijaya, and R. F. Suwarman, "Modified self-attentive bi-directional long-short term memory for detecting clickbait in Indonesian news headline," *AIP Conference Proceedings*, vol. 3049, no. 1, p. 020016, Feb. 2024, doi: 10.1063/5.0194623.

[6] B. Siregar, I. Habibie, E. B. Nababan, and Fahmi, "Identification of Indonesian clickbait news headlines with long short-term memory recurrent neural network algorithm," *J. Phys.: Conf. Ser.*, vol. 1882, no. 1, p. 012129, May 2021, doi: 10.1088/1742-6596/1882/1/012129.

[7] A. William and Y. Sari, "CLICK-ID: A novel dataset for Indonesian clickbait headlines," *Data in Brief*, vol. 32, p. 106231, Oct. 2020, doi: 10.1016/j.dib.2020.106231.

[8] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, Jul. 2020, pp. 69–75. doi: 10.1109/DATABIA50434.2020.9190447.

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[10] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431. Accessed: Dec. 21, 2023. [Online]. Available: https://aclanthology.org/E17-2068

[11] S. Martinčić-Ipšić, T. Miličić, and L. Todorovski, "The Influence of Feature Representation of Text on the Performance of Document Classification," *Applied Sciences*, vol. 9, no. 4, Art. no. 4, Jan. 2019, doi: 10.3390/app9040743.

[12] B. Kuyumcu, C. Aksakalli, and S. Delil, "An automated new approach in fast text classification (fastText): A case study for Turkish text classification without pre-processing," in *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval*, in NLPIR '19. New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 1–4. doi: 10.1145/3342827.3342828.

[13] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, Sep. 2015, doi: 10.1016/j.patcog.2015.03.009.

DOI : https://doi.org/10.52158/jacost.v5i1.655