Metode Pembobotan Hibrida untuk Ekstraksi Frasa Kunci Bahasa Arab

Evan K. Susanto[1], M. Bahrul Subkhi[2], Agus Z. Arifin[2], Maryamah[2], Rizka W. Sholikah[3], dan Rarasmaya Indraswari[4]
[1]Departemen Informatika, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia
[2]Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[3]Departemen Teknologi Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[4]Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Corresponding author: Evan K. Susanto (e-mail: evanks@stts.edu).

ABSTRACT A large amount of information makes indexing and finding the intent of documents a challenging task. Most of the documents are also not associated with any unique keyphrases. Readers are therefore forced to read the entire document to get a whole picture of its contents. The automatic keyphrase extraction using YAKE Algorithm provides a fast solution of keyphrase extraction using a single document feature. However, using only local features results in the extraction results being less relevant because they require significant terms mentioned in other documents. Another problem that rises is that some local features can't be used for Arabic language, like capital letters. In this paper, we propose a hybrid term weighting method that integrates local statistical features from a single document and external documents for unsupervised arabic keyphrase extraction systems. This keyphrase extraction system can be effectively used in Arabic. and other languages that do not use capital letters and unstructured documents such as news or scientific papers. We show that our method performs better with experimental results than the baseline methods, which are YAKE and TF-IDF.

KEYWORDS Arabic Keyphrase Extraction, Hybrid Term Weighting Method, Information Retrieval, Unsupervised Algorithm.

ABSTRAK Banyaknya informasi membuat proses pengindeksan dan pencarian inti dari dokumen menjadi permasalahan yang rumit. Sebagian besar dokumen yang tersedia tidak dilengkapi dengan kata kunci terkait. Hal ini sehingga memaksa pembaca untuk membaca seluruh dokumen untuk mendapat gambaran penuh dari konten seluruh dokumen. Ekstraksi frasa kunci otomatis yang menggunakan Algoritma YAKE memberi solusi cepat ekstraksi frasa kunci menggunakan fitur lokal dari sebuah dokumen. Namun, penggunaan fitur lokal saja membuat hasil ekstraksi menjadi kurang relevan karena diperlukan istilah signifikan yang muncul di dokumen lain. Masalah lain yang muncul adalah terdapat beberapa fitur lokal yang tidak dapat digunakan untuk bahasa Arab, misalnya huruf kapital. Pada penelitian ini, diusulkan metode pembobotan kata yang mengintegrasikan fitur statistik lokal dari sebuah dokumen dan fitur eksternal dari dokumen lain untuk sistem ekstraksi kata kunci. Metode ini dapat digunakan secara efektif pada bahasa Arab dan dapat digunakan pada bahasa lain yang tidak memiliki huruf kapital serta untuk dokumen-dokumen yang tidak terstruktur seperti berita atau karya ilmiah. Dari hasil uji coba telah dibuktikan bahwa performansi metode ini lebih baik daripada metode pembanding yaitu YAKE dan TF-IDF.

KATA KUNCI Algoritma Unsupervised, Ekstraksi Frasa Kunci Bahasa Arab, Pembobotan Kata Metode Hibrida, Temu Kembali Informasi

I. PENDAHULUAN

Jumlah informasi yang banyak pada era digital ini membuat pengindeksan dan pencarian makna dari sebuah dokumen menjadi hal yang sulit dilakukan. Hampir semua dokumen tidak dilengkapi dengan frasa kunci sehingga pembaca perlu membaca seluruh isi dokumen agar dapat diperoleh informasi kunci yang ada di dalamnya. Proses pencarian frasa kunci ini memerlukan waktu yang sangat panjang dan usaha yang luar biasa apabila dilakukan secara manual. Jumlah dokumen yang semakin banyak menyebabkan ekstraksi manual frasa kunci dari sebuah dokumen menjadi tidak efisien.

Untuk mengatasi masalah tersebut, dikembangkanlah metode ekstraksi frasa kunci otomatis. Metode-metode ini biasanya dapat mencari 5 sampai 10 frasa kunci (yang masing-masing terdiri dari satu atau lebih kata) dari sebuah dokumen. Ekstraksi frasa kunci otomatis dapat digunakan untuk memberikan gambaran singkat tentang konten dari sebuah dokumen. Selain itu, frasa kunci ini juga sangat membantu untuk proses sistem temu kembali informasi.

Salah satu masalah yang dihadapi dalam proses ekstraksi frasa otomatis sebuah dokumen adalah pemrosesan bahasa yang digunakan dalam dokumen tersebut. Setiap bahasa memiliki karakteristik masing-masing. Tidak semua bahasa dapat dinormalisasi dengan mudah menjadi sebuah representasi yang universal untuk dapat diproses dengan menggunakan satu sistem yang sama.

Ekstraksi frasa kunci otomatis untuk dokumen dalam Bahasa Arab termasuk salah satu proses yang cukup menantang. Bahasa Arab digunakan oleh sebagian besar penduduk dunia sehingga jumlah publikasi dan dokumen yang menggunakan Bahasa Arab semakin meningkat dengan cepat hari demi hari. Namun, dataset dokumen Bahasa Arab terlabel yang tersedia masih sedikit. Bahasa Arab juga memiliki karakteristik unik seperti tidak adanya huruf kapital, penulisan dari kanan ke kiri, dan perlunya proses normalisasi khusus. Hal ini membuat proses normalisasi yang biasa dilakukan dalam dokumen lain tidak dapat diterapkan secara langsung pada dokumen berbahasa Arab.

II. *TEORI DASAR*

A. TF-IDF

TF-IDF [1] merupakan salah satu metode paling umum yang digunakan untuk mengambil konteks dari suatu teks [2] dengan cara merepresentasikan dokumen dalam bentuk angka. Sesuai dengan namanya, TF-IDF mendapatkan frasa dengan cara mengkalikan Term Frequency (TF) dengan Inverse Document Frequency (IDF). Frasa yang memiliki hasil nilai yang tinggi memiliki kemungkinan tinggi bahwa frasa tersebut dapat merepresentasikan dokumen tersebut, atau frasa tersebut merupakan frasa kunci.

Term Frequency (TF) menghitung berapa kali suatu frasa muncul pada dokumen tersebut. Cara ini dipakai karena pada umumnya seberapa sering suatu frasa muncul pada suatu kata berbanding lurus dengan relevansi frasa tersebut pada teks [3]. Nilai TF dapat didapatkan hanya dengan sekedar menghitung frekuensi suatu frasa, atau dapat menggunakan cara yang lebih kompleks [4] seperti melakukan normalisasi setelah mendapatkan frekuensi kemunculan frasa untuk memperhitungkan panjang dari suatu dokumen itu sendiri.

Inverse Document Frequency (IDF) adalah sebuah metrik yang menghitung invers dari frekuensi kemunculan sebuah kata pada sekumpulan dokumen[5]. Metrik ini digunakan untuk membuat kata-kata yang umumnya sering muncul pada teks, tetapi tidak memiliki relevansi pada teks untuk disaring keluar. Contoh pada kata Bahasa Inggris adalah kata-kata: "the", "and", dan "of". Kata-kata ini sering muncul di banyak dokumen sehingga kata-kata ini memiliki nilai IDF yang rendah. Sebaliknya, sebuah dokumen yang misalnya membahas tentang sebuah topik, misalnya “river”, akan banyak mengandung kata “river” tetapi kata “river” tidak banyak muncul di dokumen lain. Hal ini menyebabkan nilai IDF dari “river” akn menjadi tinggi. IDF membantu kata-kata relevan yang lebih jarang muncul dapat memiliki nilai TF-IDF lebih tinggi.

B. YAKE

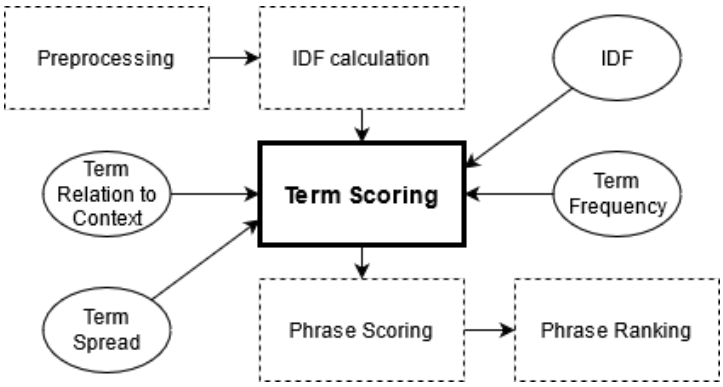
Berbeda dengan TF-IDF, YAKE [6] menghitung bobot untuk suatu kata menggunakan lima fitur dari dokumen dan perhitungan bobot yang dapat menggabungkan bobot dari beberapa kata untuk menghitung bobot suatu frasa. Kelima fitur ini merupakan: Term Casing, Term Position, Normalized Term Frequency, Term Relatedness to Context, dan Term Different Sentences. Persamaan untuk menghitung bobot dapat dilihat pada persamaan 1.

$$\mathbf{S}(\boldsymbol{t}) = \frac{\boldsymbol{T_{rel}} \times \boldsymbol{T_{position}}}{\boldsymbol{T_{case}} + \frac{\boldsymbol{TF_{norm}}}{\boldsymbol{T_{rel}}} + \frac{\boldsymbol{T_{sentence}}}{\boldsymbol{T_{rel}}}} \quad (1)$$

T_{rel} merupakan representasi seberapa dekat suatu kata dengan konteks teks tersebut. $T_{position}$ merupakan nilai posisi kata terhadap dokumen. T_{case} menghitung frekuensi kata tersebut muncul jika frasa tersebut muncul dengan huruf besar didepannya. TF_{norm} serupa dengan TF yang digunakan oleh TF-IDF, tetapi TF yang digunakan pada rumus adalah TF yang telah dinormalisasi. $T_{sentence}$ merupakan jumlah kemunculan relatif kata pada kalimat yang berbeda-beda.

T_{case} digunakan dikarenakan dalam beberapa bahasa, akronim atau kata-kata penting biasanya direpresentasikan dengan huruf besar atau huruf paling pertama dari kata tersebut merupakan huruf besar. Pada YAKE, T_{case} dihitung dengan cara menghitung berapa kali sebuah kata muncul diawali dengan huruf kapital atau semua kata-katanya terdiri dari huruf kapital. Semakin sering kata tersebut muncul dengan huruf besar semakin tinggi bobotnya[7].

$T_{position}$ mengasumsikan kata-kata yang muncul pada awal teks lebih relevan daripada jika kata-kata tersebut muncul



GAMBAR 1. **Diagram yang menggambarkan proses ekstraksi frasa kunci**

ditengah ataupun akhir teks [8]–[10]. Dikarenakan biasanya awal teks digunakan penulis untuk menarik perhatian pembaca, untuk menginformasikan kepada pembaca, tentang apa teks tersebut [3]. Selain itu, dokumen resmi seperti berita dan laporan ilmiah biasanya meletakkan kalimat utamanya di awal paragraf. Karena itu, tidak digunakan distribusi normal pada YAKE, tetapi menggunakan formula TF yang sudah dimodifikasi [4] untuk memperhitungkan bobot pada kalimat yang ada pada awal teks lebih besar daripada ditengah ataupun dibelakang teks [11].

Peran T_{rel} dalam YAKE adalah untuk menyaring stopwords didalam suatu teks. Untuk mengukur sebagaimana relevan suatu kata dalam teks tersebut. Apakah kata-kata yang sering muncul tersebut relevan terhadap teks [12] atau kata-kata yang lebih jarang muncul lebih relevan.

TF_{norm} adalah mengukur nilai frekuensi kemunculan kata dalam sebuah dokumen. Nilai ini mirip dengan nilai TF yang ada pada TF-IDF. Penggunaan metrik ini didasari dari asusmsi yang sama dengan TF-IDF bahwa sebuah kata yang muncul berulang kali pada sebuah dokumen (selain *stopwords*) kemungkinan besar merupakan frasa yang dapat mendeskripsikan dokumen tersebut [3]. Bedanya adalah pada rumus ini nilai TF dinormalisasi untuk mengurangi pengaruh dari panjang dokumen. Nilai dari TF_{norm} dihitung menggunakan rumus perhitungan frekuensi kata yang dimodifikasi [4], yaitu dari frekuensi kemunculan kata dibagi rata-rata kemunculan dari semua kata yang ada di dalam dokumen ditambah faktor normalisasi berupa satu standar deviasi.

$T_{sentence}$ menangani asumsi bahwa frasa kunci akan muncul dikalimat berbeda-beda didalam suatu teks. Asumsi ini didasari dari hal atau topik yang dibahas akan disebutkan berkali-kali pada teks dan merupakan frasa kunci. Nilai dari fitur ini dihitung dengan cara menghitung banyaknya kalimat yang mengandung kata ini dibagi dengan banyaknya kalimat yang ada pada dokumen

III. *METODOLOGI*

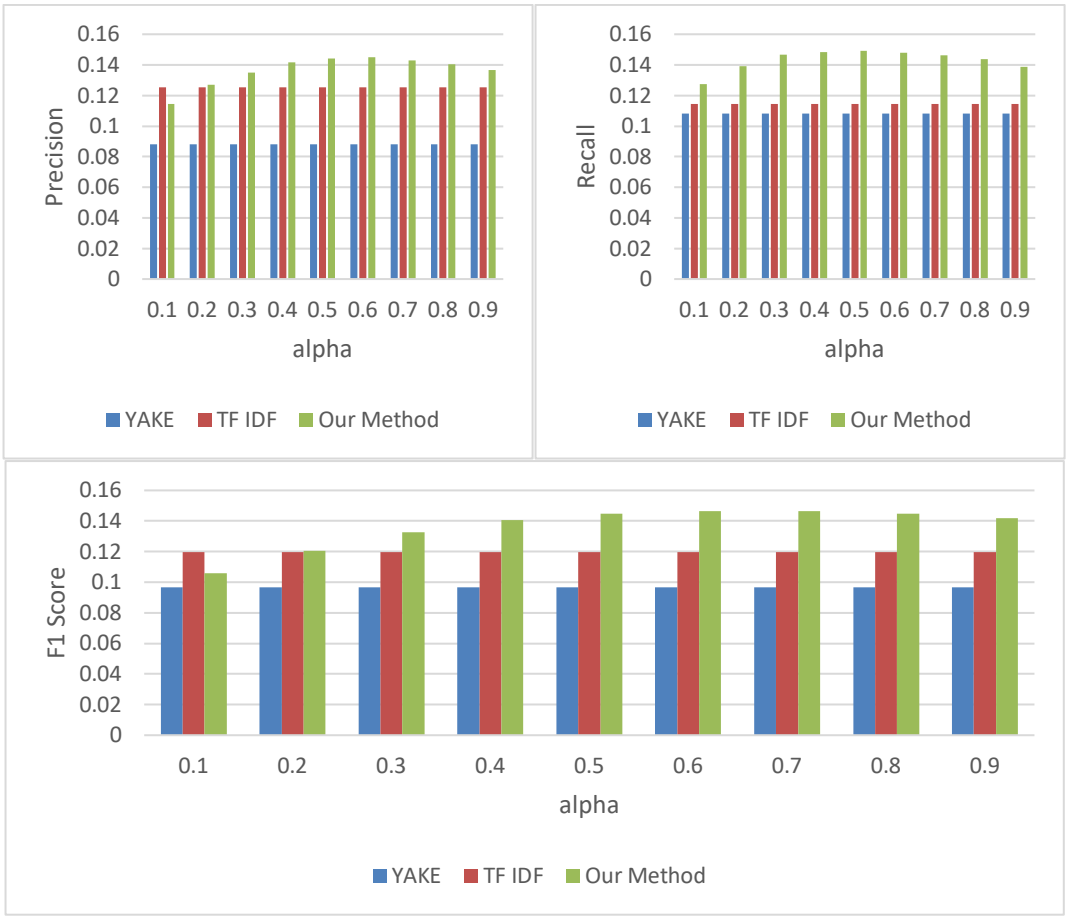
Pada bab ini, akan didiskusikan metode yang digunakan dalam penelitian ini. Akan dijelaskan langkah per langkah proses dari metode yang diajukan. Akan diperkenalkan juga metode pembobotan gabungan untuk menentukan sebagaimana mungkin suatu frasa merupakan frasa kunci dari suatu dokumen, yang didapatkan dari menggabungkan dua metode yang sudah dikenal yang menggunakan fitur lokal dan fitur eksternal dari sekumpulan dokumen. Ringkasan dari metode yang digunakan pada penelitian ini dapat dilihat pada Gambar 1.

Metode yang diajukan mengikuti alur dari algoritma YAKE, yang terdiri dari 5 langkah utama. Lima langkah utama tersebut adalah: preprocessing, term listng, term scoring, phrase scoring, dan phrase ranking. Kontribusi penelitian ini adalah melakukan modifikasi dalam langkah term scoring dengan cara melakukan integrasi pembobotan IDF yang didapat dari metode TF-IDF untuk meningkatkan kinerja pembobotan dalam YAKE. Di bab ini akan dijelaskan modifikasi yang dilakukan.

A. PREPROCESSING

Langkah pertama dalam penelitian ini adalah preprocessing. Pada langkah ini, dihapus semua angka dan tanda baca yang tidak menandakan akhir kalimat dari semua dokumen. Setelah itu, semua dokumen dipecah menjadi kalimat-kalimat. Pada penelitian ini tidak dilakukan preprocessing lain yang biasanya dilakukan pada algoritma lain seperti stemming atau melakukan normalisasi pada huruf Arab, karena hal ini dapat memberikan dampak negatif pada kualitas frasa kunci yang diekstraksi. Algoritma ini juga di rancang untuk dapat digunakan lintas bahasa, dapat digunakan walaupun hanya terdapat sedikit dokumen. Stemming atau normalisasi tidak digunakan karena mungkin saja tidak dapat diaplikasikan terdapat bahasa - bahasa lain.

Proses lain dalam preprocessing adalah mendaftar semua kata yang mungkin menjadi kata kunci atau bagian dari frasa kunci. Hal ini dilakukan dengan cara membuat set dari setiap kata (hanya satu per kata) yang muncul pada dataset. Pada langkah ini, dibutuhkan juga input yang terdiri dari kata-kata stopword yang digunakan pada bahasa target. Lalu setiap kata stopword yang muncul pada set kata-kata diberikan label. Setelah semua kata sudah didaftar dan semua stopword sudah diberikan label, proses akan dilanjutkan ke langkah perhitungan IDF.



GAMBAR 2. **Perbandingan performansi antara YAKE, TF-IDF, dan metode yang diusulkan menggunakan semua dokumen pada dataset**

B. KALKULASI IDF

Langkah berikutnya dari metode pada penelitian ini adalah menghitung nilai IDF untuk setiap kata-kata. Nilai IDF dihitung dari kata-kata menggunakan metode standar untuk menghitung IDF, yaitu menghitung log dari jumlah dokumen dibagi dengan jumlah dokumen yang mengandung kata-kata tersebut. Pertama, didaftar semua kata-kata pada dokumen. Lalu dihitung jumlah dokumen pada corpus, yang disimbolkan juga sebagai N. Terakhir, untuk setiap kata t, dihitung jumlah dokumen yang mengandung kata-kata t tersebut, yang disimbolkan sebagai n_t. Persamaan untuk perhitungan nilai IDF untuk suatu kata t (IDF_t) dapat dilihat pada Persamaan 2.

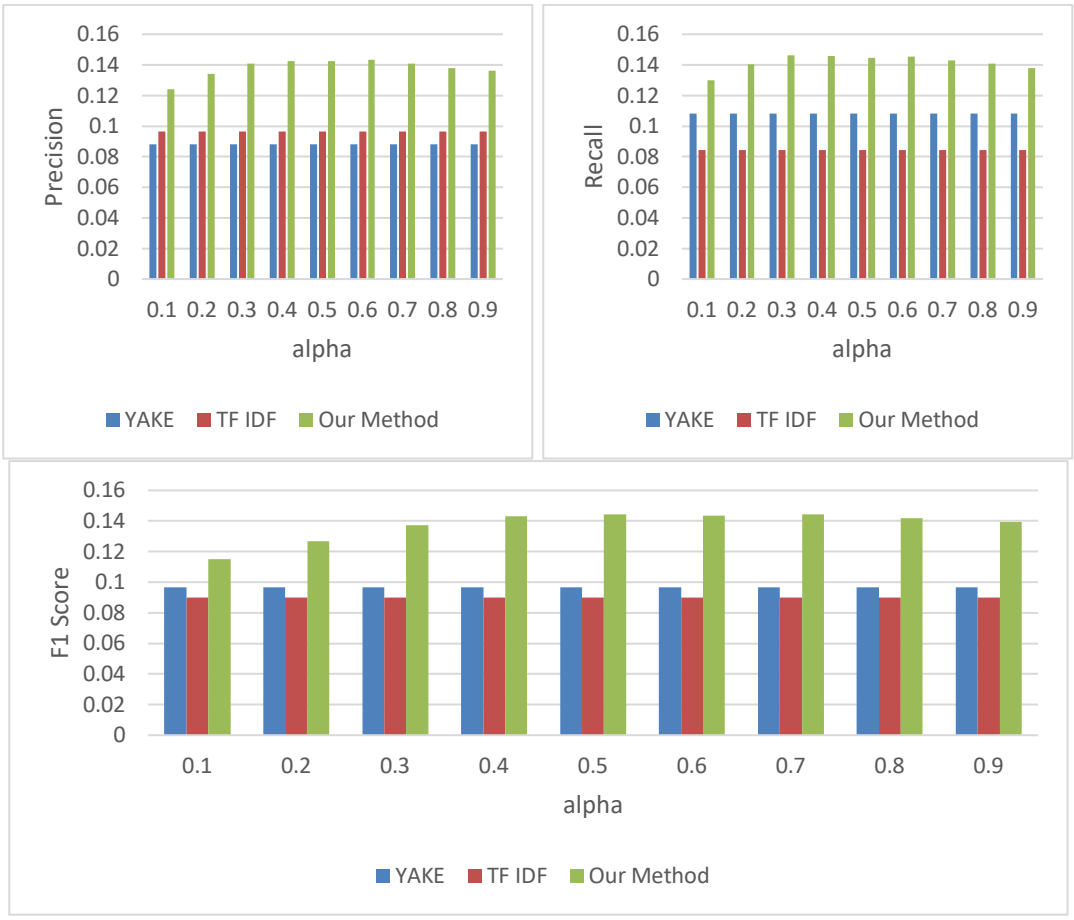
$$\boldsymbol{IDF_t = \log\left(\frac{N}{n_t}\right)} \quad (2)$$

C. TERM SCORING WITH HYBRID TERM WEIGHTING METHOD

Langkah ketiga dari proses ekstraksi frasa kunci pada penelitian ini adalah penilaian kata-kata. Pada penelitian ini diusulkan 3 modifikasi dari formula yang digunakan pada algoritma YAKE: (1) menghapus term casing dan term position dari formula YAKE, (2) menambahkan kalkulasi untuk menentukan seberapa penting suatu kata bedasarkan dokumen eksternal, dan (3) menambahkan hyperparameter yang dapat menyesuaikan pengaruh penilaian kata-kata terhadap dokumen target dan dokumen eksternal.

Modifikasi pertama adalah menghapuskan term casing dan term position dari formula YAKE. Term casing dihapus karena Bahasa Arab tidak memiliki huruf besar. Modifikasi ini tidak hanya berguna untuk Bahasa Arab, tetapi juga berguna untuk bahasa lain yang tidak memiliki huruf besar seperti Bahasa Mandarin, Bahasa Jepang, ataupun Bahasa Korea. Pada saat uji coba menggunakan YAKE, fitur ini menyebabkan YAKE selalu menganggap bahwa named entity merupakan frasa kunci, walaupun kenyataannya tidak demikian.

Term position juga dihapus karena term position mengasumsikan kata-kata pada awal dokumen lebih besar kemungkinannya kata-kata tersebut merupakan frasa kunci. Asumsi ini benar jika frasa kunci yang berusaha diekstrak dari dokumen formal seperti berita atau jurnal ilmiah. Tetapi, asumsi ini tidak benar jika pada dokumen yang tidak formal atau dokumen yang lebih tidak terstruktur seperti teks religius



GAMBAR 3. **Perbandingan performansi antara YAKE, TF-IDF, dan metode yang diusulkan menggunakan hanya sebagian dokumen pada dataset**

ataupun artikel di internet. Bobot ini dihapus untuk membuat pembobotan menjadi lebih umum dan dapat digunakan pada dokumen yang kurang terstruktur.

Modifikasi kedua memperkenalkan fitur baru yang merepresentasikan bobot dari suatu kata bedasarkan dokumen eksternal. Fitur ini dapat dianggap sebagai pengganti dari fitur casing karena keduanya dapat mendeteksi kata-kata penting. Banyak metode yang dapat digunakan untuk menghitung seberapa penting kata-kata bedasarkan dokumen eksternal, seperti Inverse Document Frequency (IDF), Inverse Class Frequency (ICF) [13], Inverse Book Frequency (IBF) [14], dan Inverse Preference Frequency (IPF) [15]. Dalam penelitian ini digunakan pembobotan IDF.

Modifikasi terakhir ada hyperparameter α yang dapat menyesuaikan pengaruh bobot yang dihitung dari dokumen target dan dokumen eksternal. Ide utama dari hyperparameter ini adalah, jika terdapat 2 algoritma berbeda untuk ekstraksi frasa kunci, satu menggunakan fitur dari dokumen target seperti YAKE, dan satu lagi menggunakan dokumen ekternal seperti TF-IDF, performa YAKE tetap sama tidak peduli seberapa banyak dokumen yang tersedia pada dataset. Tetapi sebaliknya, peforma TF-IDF membaik semakin banyak dokumen yang tersedia pada dataset. Tetapi peforma TF-IDF juga memburuk semakin sedikit dokumen yang tersedia pada dataset. Dengan menambahkan hyperparameter yang dapat mengatur pengaruh dari kedua cara ini bedasarkan jumlah dokumen pada dataset yang tersedia.

Secara total, terdapat 4 fitur yang diekstraksi untuk setiap kata untuk menghitung nilai dari kata tersebut, 4 fitur tersebut merupakan: IDF, frekuensi yang dinormalisasi, sebagaimana dekat kata tersebut dengan konteks, dan kemunculan kata dikalimat-kalimat yang berbeda. Nilai IDF telah dihitung pada langkah sebelumnya. Untuk tiga fitur berikutnya, akan dipinjam metode perhitungan algoritma YAKE. Frekuensi yang telah dinormalisasi didapatkan dari jumlah kemunculan kata didokumen dibagi dengan rata-rata jumlah kemunculan setiap kata ditambakan dengan satu standar deviasi.

$$W_{freq_t} = \frac{freq_t}{\overline{freq} + \sigma} \tag{3}$$

Untuk mendapatkan bobot ini untuk setiap kata, dikalkulasikan jumlah kemunculan tiap kata di dokumen tersebut. Setelah itu, dihitung rata-rata kemunculan kata dalam dokumen dan deviasi standarnya, yang disebut juga sebagai $\overline{freq}$ dan σ. Langkah terakhir, untuk setiap kata-kata t di dokumen, dihitung bobot frekuensi W_{freq_t}. Rumus yang

digunakan untuk menormalisasi term frequency dapat dilihat pada persamaan 3.

$$\boldsymbol{D} = \frac{|\boldsymbol{T_t}|}{\boldsymbol{\Sigma_{k \in T_t} Co(t,k)}} \quad (4)$$

$$\boldsymbol{W_{rel}} = \boldsymbol{1} + (\boldsymbol{D_L} + \boldsymbol{D_R}) \times \frac{\boldsymbol{freq_t}}{\boldsymbol{freq_{max}}} \quad (5)$$

D merepresentasikan faktor dispersi dari sebuah kata, T_t adalah set dari kata-kata yang muncul pada satu sisi (baik kanan maupun kiri), dan $Co(t,k)$ adalah jumlah kata t dan k muncul pada saat yang bersamaan. W_{rel} merepresentasikan seberapa mendekati konteks suatu kata, D_L dan D_R ada perhitungan faktor dispersi dari sisi kiri kata maupun sisi kanan kata, $freq_t$ adalah jumlah kemunculan kata t pada dokumen dan $freq_{max}$ adalah jumlah kemunculan maksimal dari suatu kata. Rumus perhitungan term relatedness to context dapat dilihat pada rumus 4 dan 5.

Perhitungan bobot terakhir yang digunakan pada kalkulasi ini ada spread weight dari suatu kata, yang dimana dihitung dengan cara membagi jumlah kalimat dalam dokumen yang mengandung kata target dengan jumlah kalimat dalam dokumen. Untuk menghitung spread weight (W_{spread_t}) dari kata t, pertama-tama dibuat set yang mengandung semua kalimat dalam dokumen yang disebut sebagai $sent$. Lalu dibuat set yang mengandung kata t, yang disebut juga sebagai $sent_t$. Berikutnya dihitung jumlah elemen pada $sent_t$ dan $sent$ untuk menghitung spread weight dari kata t (W_{spread_t}). Persamaan dari W_{spread_t}dapat dilihat pada rumus 6.

$$\boldsymbol{W_{spread_t}} = \frac{|\boldsymbol{sent_t}|}{|\boldsymbol{sent}|} \quad (6)$$

Keempat fitur ini digabungkan dengan hyperparameter α untuk membentuk rumus pembobotan yang dapat dilihat pada persamaan 7.

$$\boldsymbol{Score_t} = \frac{\boldsymbol{W_{rel}}}{\boldsymbol{\alpha(W_{freq}+W_{spread})+(1-\alpha)*W_{IDF}}} \quad (7)$$

$Score_t$ merepresentasikan nilai kata, W_{rel_t} merepresentasikan sebagaimana dekat kata dengan konteks, W_{freq_t} adalah bobot frekuensi yang telah di normalisasi, W_{spread_t} mereperesentasikan spread weight dari kata, W_{IDF_t} adalah nilai IDF dari kata, dan alpha adalah nilai hyperparameter penyesuaian dimana $0 < \alpha < 1$. Hyperparameter α dapat digunakan untuk mengatur fitur apa yang memiliki efek lebih tinggi terhadap metode penilaian. Semakin kecil α berarti fitur dari dokumen eksternal akan memiliki efek lebih tinggi, dan semakin tinggi α berarti fitur dokumen internal akan memiliki efek lebih tinggi terhadap pembobotan. Ini dikarenakan α akan dikalikan dengan $(W_{freq} + W_{spread})$, yang merupakan fitur dokumen internal dan $(1 - \alpha)$ akan dikalikan dengan W_{IDF}, yang merupakan fitur dokumen eksternal.

D. PHRASE SCORING

Setelah setiap bobot kata dalam dokumen dihitung, mulai dihitung nilai dari frasa kunci kandidat yang terdiri dari lebih dari satu kata. Dalam penelitian ini, frasa kunci yang dicari terdiri tidak lebih dari tiga kata tetapi bisa kurang dari tiga kata. Langkah pertama dalam proses ini adalah membuat daftar setiap frasa yang terdiri dari dua atau tiga kata yang muncul bersamaan. Lalu dihapus frasa yang dimulai atau diakhiri dengan stopword, dikarenakan frasa kunci jarang dimulai atau diakhiri dengan stopword [16]–[18].

Langkah sebelumnya akan menghasilkan tiga jenis frasa kunci: frasa yang terdiri dari tiga kata yang bukan merupakan stopword, frasa yang terdiri dari dua kata yang bukan merupakan stopword, dan frasa yang teridiri dari satu kata stopword diantara dua kata yang bukan merupakan stopword. Ketiga jenis frasa kunci ini kemudian digabung menjadi satu daftar frasa kunci dan menghitung nilai akhir setiap frasa kunci kandidat. Untuk frasa yang tidak mengandung stopword dapat dihitung nilai akhirnya menggunakan fumus 8.

$$\boldsymbol{Score_{kp}} = \frac{\boldsymbol{\Pi_{t \in kp}\, Score_t}}{\boldsymbol{freq_{kp} \times (1 + \Sigma_{t \in kp} Score_t)}} \quad (8)$$

Langkah pertama untuk menghitung nilai akhir adalah menghitung jumlah nilai $Score_t$ untuk setiap kata dalam frasa kunci kandidat kp. Lalu dihitung jumlah kemunculan frasa kunci kp pada dokumen ($freq_{kp}$). Terakhir, jumlah nilai dari setiap kata pada frasa kunci $\Pi_{t \in kp}\, Score_t$ dibagi dengan jumlah kemunculan frasa kunci $freq_{kp}$ dan 1 ditambahkan dengan jumlah nilai dari setiap kata pada frasa kunci $\Sigma_{t \in kp}\, Score_t$.

Digunakan rumus berbeda untuk menghidung kata kunci yang mengandung stopword ditengah. Nilai kata dari stopword diganti dengan dengan probabilitas kata pertama muncul sebelum stopword dan probabilitas kata terakhir muncul setelah stopword. Rumus yang digunakan untuk menghitung nilai frasa kunci kandidat yang mengandung stopword dapat dilihat pada rumus 9 dan 10.

$$\boldsymbol{S_{t_2}} = \boldsymbol{1 - P(t_2|t_1) \times P(t_3|t_2)} \quad (9)$$

$$\boldsymbol{Score_{kp}} = \frac{\boldsymbol{Score_{t_1} \times (1 + S_{t_2}) \times Score_{t_3}}}{\boldsymbol{freq_{kp} \times (1 + Score_{t_1} - S_{t_2} + Score_{t_3})}} \quad (10)$$

Pertama dihitung nilai dari stopword t_2 (S_{t_2}) dengan mengkalikan $P(t_2|t_1)$, probabilitas stopword t_2 muncul setelah kata pertama t_1, dengan $P(t_3|t_2)$, probabilitas kata

terakhir t_3 muncul setelah stopword t_2. Lalu digunakan rumus yang serupa dengan rumus 8 untuk menghitung nilai akhir, dengan beberapa perbedaan. Nilai $Score_{t_1}$ kata pertama dikalikan dengan nilai $Score_{t_2}$ kata kedua dan nilai $1 + S_{t_2}$ stopword. Kemudian hasil yang diperoleh dibagi dengan hasil perkalian dari jumlah kemunculan frasa kunci kandidat dengan hasil pertambahan nilai $Score_{t_1}$ kata pertama, nilai $Score_{t_3}$ kata terakhir dan 1, lalu mengurangi nya dengan nilai stopword S_{t_2}.

E. PHRASE RANKING

Langkah terakhir dari proses yang diusulkan pada penelitian ini adalah meranking setiap frasa kunci kandidat bedasarkan nilai mereka. Frasa kunci kandidat diurutkan bedasarkan nilai yang didapatkan dari 2 langkah sebelum ini dari kecil ke besar. Frasa kunci yang memiliki nilai lebih rendah akan memiliki ranking lebih tinggi dari yang memiliki nilai yang lebih tinggi. Lalu akan diambil frasa kunci tertinggi N sebagai frasa kunci.

Sebelum frasa kunci tertinggi N diambil, akan dilakukan penghapusan duplikat. Pada tahap ini terkadang masih ada frasa kunci yang mirip dengan frasa kunci lain pada daftar. Ini dikarenakan tidak dilakukannya proses stemming pada tahap preprocessing yang akan menggabungkan frasa kunci yang mirip menjadi satu. Digunakan metode Levenshtein distance untuk menghitung tingkat kemiripan dari dua frasa kunci yang ada di daftar. Jika kedua kata kunci memiliki tingkat kemiripan diatas nilai batas, frasa kunci yang memiliki nilai yang lebih tinggi (ranking lebih rendah) akan dihapus. Proses ini akan diulangi sampai tidak ada frasa kunci yang serupa pada frasa kunci kandidat tertinggi N. Setelah semua duplikasi telah dihapus, makan frasa kunci kandidat yang masuk dalam batas N akan menjadi frasa kunci yang berhasil di ekstraksi.

IV. HASIL EKSPERIMEN DAN DISKUSI

A. HASIL EKSPERIMEN

Terdapat 2 dataset yang digunakan untuk penelitian ini. Pertama adalah Arabic Keyphrase Extraction Corpus (AKEC) [19]. Dataset ini memiliki 160 dokumen dalam Bahasa arab dari 4 corpora dan dokumen-dokumen tersebut dapat dibagi menjadi 9 topik berbeda. Dataset kedua diambil dari [20]. Dataset ini memiliki 400 dokumen yang dapat dibagi menjadi 18 topik berbeda. Pada penelitian ini, kedua dataset ini digabung membentuk 1 dataset. Contoh data yang digunakan dapat dilihat pada Tabel 1.

Dilakukan perbandingan metode yang diajukan, YAKE, dan TF-IDF. Performa masing-masing metode dilihat dari menghitung penilaian F1 [21]. Nilai F1 didapatkan dengan cara menghitung akurasi rata-rata dan frasa yang dapat ditebak (recall). Persamaan untuk menghitung presisi, recall, dan F1 dapat dilihat di persamaan 11, 12, 13.

Agar dapat mendapatkan penilaian lebih akurat, preprocessing dilakukan juga kepada frasa kunci yang dibuat secara manual. Stopword, tanda baca, huruf yang berulang, dan angka dihapus dari frasa kunci yang dibuat secara manual. Lalu frasa kunci tersebut distemming menggunakan ISRI Arabic stemmer [22]. Selain itu frasa kunci yang diambil hanyalah frasa kunci yang terdiri dari maximal 3 kata.

TABEL I
CONTOH POTONGAN DOKUMEN DAN FRASA KUNCINYA

Potongan Dokumen	Frasa Kunci
جغرافيا ليسوتو. أبرز حقيقة جغرافية عن ليسوتو أنها الدولة الوحيدة المستقلة في العالم التي تقع كليًا فوق متر في الارتفاع. أخفض نقطة فيها ترتفع متر عن سطح البحر ما يجعلها أعلى أخفض نقطة في العالم لدولة ما. مناخ ليسوتو أكثر برودة من المناطق الأخرى في نفس خط العرض بسب ارتفاعها ويمكن تصنيفه في المناخات القارية.=الموقع=.ليسوتو بلد في أفريقيا الجنوبية ° وتقع عند خط عرض حوالي ° جنوبًا وبخط طول شرقًا...	'أعلى أخفض نقطة في العالم', 'جنوب أفريقيا', 'المناخ', 'التقسيم الإداري', 'جغرافيا ليسوتو', 'الأخطار البيئية', 'الموارد الطبيعية', 'دولة مغلقة', 'المناخات القارية'

$$\textbf{precision} = \frac{|\{\textbf{relevant kp}\} \cap \{\textbf{retrieved kp}\}|}{|\{\textbf{retrieved kp}\}|} \quad (11)$$

$$\textbf{recall} = \frac{|\{\textbf{relevant kp}\} \cap \{\textbf{retrieved kp}\}|}{|\{\textbf{relevant kp}\}|} \quad (12)$$

$$\textbf{F1 Score} = \frac{2 \times \textbf{precision} \times \textbf{recall}}{\textbf{precision} + \textbf{recall}} \quad (13)$$

Evaluasi dilakukan dengan menggunakan 2 skenario. Skenario pertama menggunakan semua dokumen training untuk menghitung IDF dan sebagai data testing. Skenario kedua, 25% dari total dataset digunakan untuk traning IDF dan 75% sisanya sebagai data testing. Skenario kedua digunakan untuk melakukan evaluasi jika hanya terdapat sedikit data yang tersedia untuk training IDF. Pada kedua skenario alpha yang digunakan adalah: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.

Pada Gambar 2 dapat dilihat algoritma yang diajukan bekerja lebih baik daripada YAKE dan TF-IDF pada hampir semua nilai alpha. Hal ini membuktikan dengan mengintegrasikan dokumen diluar dokumen target dapat meningkatkan kemampuan algoritma untuk ekstraksi frasa kunci dari suatu dokumen. Pada Gambar 3, dapat dilihat pada kasus dimana data training sedikit, performa TF-IDF memburuk, tetapi algoritma yang diajukan masih dapat bekerja lebih baik dari TF-IDF maupun YAKE. Hal ini

membuktikan dapat melakukan ekstraksi bergantung dari hanya dokumen target juga merupakan hal penting terutama pada kondisi dimana data training sedikit.

Dapat dilihat juga untuk setiap nilai α, performa algoritma yang diajukan lebih baik daripada YAKE. Maka dapat disimpulkan bahwa formula yang diajukan dapat melakukan ekstraksi lebih baik daripada YAKE untuk Bahasa Arab. Dikarenakan formula yang diajukan tidak lagi menggunakan T_{case} yang tidak relevan diakibatkan oleh sifat Bahasa Arab yang tidak memiliki kapitalisasi dan juga menggunakan IDF.

Dari kedua skenario dapat dilihat juga bahwa algoritma yang diajukan bekerja paling baik menggunakan α 0.5 dan 0.6. Hal ini disebabkan dengan nilai α demikian, maka akan dipertimbangkan dengan sama rata dokumen target dan dokumen lain. Penelitian lebih lanjut dapat dilakukan untuk memastikan apakah α yang optimal dapat berubah bedasarkan jumlah dokumen didalam corpus.

B. DISKUSI

Penelitian ini berusaha menggabungkan 2 metode penilaian untuk extraksi kata frasa yang bersifat unsupervised. Metode tersebut menggukan fitur lokal yang berasal dari dokumen input dan menggunakan fitur eksternal yang didapatkan dari kumpulan dokumen diluar dokumen input. Kedua metode mempunyai kelebihan yang membuat mereka populer, tetapi juga mempunyai kekurangan yang membuat mereka tidak cocok untuk diaplikasikan dalam kasus-kasus tertentu.

Algoritma yang menggunakan fitur lokal seperti YAKE dapat bekerja lebih baik dalam kasus dimana sangat sedikit atau bahkan tidak ada dokumen yang tersedia. Algoritma seperti ini biasanya universal, tidak terkunci disatu bahasa tertentu tanpa perlu modifikasi yang signifikan. Tetapi algoritma ini akan gagal pada kasus dimana sebagai mana penting frasa tidak dapat ditentukan hanya dari fitur lokal saja. Dikarenakan satu dokumen tidak dapat menyimpan semua informasi dan konteks dari suatu kata. Algoritma ini juga bisa gagal jika fitur yang diperlukan tidak ada (seperti pada kasus dimana YAKE tidak dapat ekstraksi penilaian yang didapat dari casing, dikarenakan Bahasa Arab tidak memiliki huruf besar).

Disisi lain, algoritma yang menggunakan informasi dari banyak dokumen dapat bekerja lebih baik dalam menentukan pembobotan kata dikarenakan algoritma tersebut dapat mendapatkan informasi dan konteks dari suatu kata lebih baik. Tetapi, algoritma ini akan gagal jika jumlah dokumen pendukung sedikit, dikarenakan algoritma ini tidak dapat mendapatkan cukup informasi mengenai suatu kata. Metode ini juga tidak dapat digunakan lintas bahasa dengan mudah seperti algoritma yang hanya menggunakan fitur lokal. Dilihat bahwa kedua algoritma ini memiliki kelebihan dan kekurangan yang dapat digabungkan untuk menutupi kekurangan yang lainnya.

Penelitian ini membuktikan dengan menggabungkan kedua metode ini dapat menghasilkan hasil yang lebih baik daripada kedua metode tersebut. Hal ini sangat menarik dikarenakan akan ada banyaknya kemungkinan area pengembangan yang dapat dieksplorasi lebih lanjut, seperti banyaknya jenis berbeda dari fitur lokal dan eksternal yang dapat digunakan untuk memodifikasi cara pembobotan. Posibilitas untuk mencari pengaturan terbaik pembobotan terbaik dari kedua jenis fitur juga dapat menjadi area penelitian yang menarik.

V. KESIMPULAN

Pada penelitian ini diusulkan sebuah metode pembobotan gabungan untuk ekstraksi frasa kunci. Pada penelitian ini diusulkan formula yang mengintegrasikan fitur statistik lokal yang didapatkan dari dokumen target dan fitur yang didapatkan dari dokumen pendukung lain. Pada penelitian ini juga diperkenalkan hyperparameter baru pada formula yang dapat mengatur pembobotan antara dokumen target dan dokumen pendukung untuk mengakomodasi jumlah dokumen pendukung yang digunakan untuk training. Pada penelitian ini juga ditunjukkan bahwa metode yang diusulkan memiliki peforma yang lebih baik daripada YAKE maupun TF-IDF. Pada penelitian ini juga ditunjukkan bahwa hyperparameter dapat meningkatkan peforma metode yang diusulkan bedasarkan jumlah dokumen pendukung. Pada penelitian ini juga dapat disimpulkan dari percobaan yang dilakukan bahwa nilai optimum untuk hyperparameter adalah diantara 0.5-0.6.

Terdapat beberapa pengembangan yang mungkin dapat diteliti kedepannya. Salah satunya adalah untuk mencoba metode pembobotan untuk fitur luar lain, seperti ICF [13], IBF [14], atau menggabungkan metrik-metrik lain [23]–[27]. Kemungkinan lain yang bisa dicoba adalah untuk mengaplikasikan metode yang diusulkan ini kepada bahasa lain atau jenis dokumen lain seperti artikel yang diambil dari internet atau teks religius. Untuk penelitian selanjutnya juga dapat mencoba untuk mendapatkan nilai hyperparameter yang optimal bedasarkan jumlah dokumen yang tersedia untuk training dan seberapa relevan dokumen training dan dokumen testing.

PERAN PENULIS

Evan Kusuma Susanto: Analisa permasalahan, investigasi, perumusan solusi, implementasi program, uji coba, penyusunan draf manuskrip,
M. Bahrul Subkhi: Analisa permasalahan, investigasi, perumusan solusi, penyusunan dataset, persiapan skenario uji coba, penyusunan draf manuskrip.
Agus Zainal Arifin, Maryamah, Rizka W. Sholikah, dan Rarasmaya Indraswari: Memiliki peran yang sama dalam membimbing investigasi, penyusunan solusi, dan penyusunan manuskrip.

COPYRIGHT

DAFTAR PUSTAKA

[1] G. Salton, "Automatic text processing: the transformation," *Anal. Retr. Inf. by Comput.*, 1989.
[2] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," *Int. J. Digit. Libr.*, 2016, doi: 10.1007/s00799-015-0156-0.
[3] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM J. Res. Dev.*, 2010, doi: 10.1147/rd.14.0309.
[4] C. D. Manning, P. Raghavan, H. Schutze, C. D. Manning, P. Raghavan, and H. Schutze, "Scoring, term weighting, and the vector space model," in *Introduction to Information Retrieval*, 2012.
[5] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Doc.*, 2004, doi: 10.1108/00220410410560573.
[6] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "YAKE! Keyword extraction from single documents using multiple local features," *Inf. Sci. (Ny).*, 2020, doi: 10.1016/j.ins.2019.09.013.
[7] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, "A text feature based automatic keyword extraction method for single documents," 2018, doi: 10.1007/978-3-319-76941-7_63.
[8] F. Boudin, "Unsupervised keyphrase extraction with multipartite graphs," 2018, doi: 10.18653/v1/n18-2105.
[9] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," 2003, doi: 10.3115/1119355.1119383.
[10] C. Florescu and C. Caragea, "PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents," 2017, doi: 10.18653/v1/P17-1102.
[11] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, "YAKE! collection-independent automatic keyword extractor," 2018, doi: 10.1007/978-3-319-76941-7_80.
[12] D. MacHado, T. Barbosa, S. Pais, B. Martins, and G. Dias, "Universal mobile information retrieval," 2009, doi: 10.1007/978-3-642-02710-9_38.
[13] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Inf. Sci. (Ny).*, 2013, doi: 10.1016/j.ins.2013.02.029.
[14] M. A. Fauzi, A. Z. Arifin, and A. Yuniarti, "Arabic Book Retrieval using Class and Book Index Based Term Weighting," *Int. J. Electr. Comput. Eng.*, 2017, doi: 10.11591/ijece.v7i6.pp3705-3710.
[15] K. F. H. Holle, A. Z. Arifin, and D. Purwitasari, "Preference Based Term Weighting for Arabic Fiqh Document Ranking," *J. Ilmu Komput. dan Inf. (Journal Comput. Sci. Information)*, vol. 151, pp. 45–52, 2015, doi: http://dx.doi.org/10.21609/jiki.v8i1.283.
[16] S. Das Gollapalli, X. L. Li, and P. Yang, "Incorporating expert knowledge into keyphrase extraction," 2017.
[17] D. Mahata, J. Kuriakose, R. R. Shah, and R. Zimmermann, "Key2Vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings," 2018, doi: 10.18653/v1/n18-2100.
[18] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," in *Text Mining: Applications and Theory*, 2010.
[19] M. Helmy, M. Basaldella, E. Maddalena, S. Mizzaro, and G. Demartini, "Towards building a standard dataset for Arabic keyphrase extraction evaluation," 2017, doi: 10.1109/IALP.2016.7875927.
[20] M. Al Logmani and H. Al Muhtaseb, "Arabic Dataset for Automatic Keyphrase Extraction," 2017, doi: 10.5121/csit.2017.70121.
[21] Y. Sasaki, "The truth of the F-measure," *Teach Tutor mater*, 2007.
[22] M. G. Syarief, O. T. Kurahman, A. F. Huda, and W. Darmalaksana, "Improving Arabic Stemmer: ISRI Stemmer," 2019, doi: 10.1109/ICWT47785.2019.8978248.
[23] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 10, no. 2, p. e1339, 2020.
[24] Z. A. Merrouni, B. Frikh, and B. Ouhbi, "Automatic keyphrase extraction: a survey and trends," *J. Intell. Inf. Syst.*, vol. 54, no. 2, pp. 391–424, 2020.
[25] G. Rabby, S. Azad, M. Mahmud, K. Z. Zamli, and M. M. Rahman, "Teket: a tree-based unsupervised keyphrase extraction technique," *Cognit. Comput.*, vol. 12, no. 4, pp. 811–833, 2020.
[26] Y. Zhang, Y. Chang, X. Liu, S. Das Gollapalli, X. Li, and C. Xiao, "Mike: keyphrase extraction by integrating multidimensional information," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1349–1358.
[27] E. Papagiannopoulou and G. Tsoumakas, "Local word vectors guiding keyphrase extraction," *Inf. Process. \& Manag.*, vol. 54, no. 6, pp. 888–902, 2018.