Perbandingan Algoritma Sentencepiece BPE dan Unigram Pada Tokenisasi Artikel Bahasa Indonesia

Triwidyastuti Jamaluddin[1], Moch Arif Bijaksana[2], Ibnu Asror[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]triwidyastuti@student.telkomuniversity.ac.id, [2]arifbijaksana@telkomuniversity.ac.id,
[3]iasror@telkomuniversity.ac.id

Abstrak

Tokenisasi merupakan sebuah konsep yang mencakup proses sederhana dimana urutan teks dipecah menjadi bagian-bagian yang lebih kecil atau token dan kemudian dimasukkan sebagai input ke dalam model Natural language processing (NLP), atau proses model yang lebih kompleks seperti menerapkan pengetahuan dunia Deep Learning (DL). Tokenisasi akan lebih rumit ketika berhadapan dengan kasus semua kata dikelompokkan menjadi satu token atau tanpa pemisah dan kesalahan dalam tipografi. Paper ini mengusulkan model unsupervised tokenization menggunakan subword unit tokenizer dan detokenizer representasi oleh neural network, implementasi algoritma Byte Pair Encoding (BPE) dan Unigram Language Model. Selain itu, mengeksploitasi sentencePiece, model segmentasi pada kalimat dapat dilatih tanpa spasi. Ekpsperimen menggunakan bahasa Indonesia menghasilkan akurasi 54.6% dan 87.0% untuk Byte Pair Encoding (BPE) dan Unigram Languange Model, masing-masing.

Kata kunci: ***SentencePiece, Subword Tokenizer, Byte Pair Encoding* (BPE),** ***Unigram Language Model.***

Abstract

Tokenization is a concept that includes a simple process where the sequence of the text is split up into smaller parts or tokens and then entered as input into the model of natural language processing (NLP), or more complex process models such as applying the world knowledge of Deep Learning (DL). Tokenization will be more complicated when dealing with cases where all words are grouped into a single token or without separators and errors in typography. This paper proposes a model unsupervised tokenization using subword tokenizer and detokenize representation by neural networks, implementation of algorithm Byte Pair Encoding (BPE) and Unigram Language Model. Moreover, exploiting sentencePiece, the segmentation model of sentences can be trained without spaces. Experiments using the Indonesian language resulted in 54.6% and 87.0% in accuracy for Byte Pair Encoding (BPE) and Unigram Language Model, respectively.

Keywords: ***SentencePiece, Subword Tokenizer, Byte Pair Encoding* (BPE),** ***Unigram Language Model.***

1. Pendahuluan

Latar Belakang

Tokenisasi adalah hal yang sangat penting dalam pemrosesan text seperti sentimen analisis, deteksi topik, dan spam filtering[6]. Dalam klasifikasi text, representasi kalimat dapat diperhitungkan berdasarkan token penyusun kalimat yang secara khusus, sebuah kalimat terlebih dahulu diubah menjadi unit-unit yang lebih kecil bermak- na seperti karakter, kata-kata dan sub kata[11]. Kemudian token dapat dimodelkan pada neural network seperti Convolutional Neural Network (CNN) [3], atau Neural Machine Translation (NMT)[10].

Bahasa Indonesia memiliki kekayaan kosakata yang memadai sebagai sarana pikir, ekspresi, dan komunikasi di berbagai bidang kehidupan, terdiri dari sekitar 85.000 jumlah entri, 41.250 lema, 48.250 sublema, serta peribahasa sebanyak 2.036[1]. Dalam bahasa Indonesia menggunakan alfabet latin[2] sama seperti bahasa Inggris dapat di lihat dengan white space yang merupakan indikator yang baik pada segmentasi kata. Namun, tokenisasi akan menjadi masalah apabila dalam bahasa yang tidak tersegmentasi seperti pada kalimat yang tidak memiliki spasi dan kesalahan dalam tipografi. Berbagai kesalahan dalam penulisan bahasa Indonesia yang paling sering terjadi adalah penggunaan kata depan dan kata penghubung. Misalnya, kata ”di jalan” sering kali ditulis dengan kata ”dijalan” atau pada kata ”ke rumah” sering kali ditulis menjadi kata ”kerumah”. Padahal, untuk setiap keterangan yang merujuk tempat ataupun waktu, penulisannya dipisahkan dengan kata induknya.

Pada penelitian ini, kami mengeksplorasi tokenisasi yang mengacu pada pemisahan tanda baca dan mem- bagi token menjadi kata atau subword menggunakan model *SentencePiece* sebagai unsupervised tokenizer dan detokenizer pada neural network-based text generation systems. Implementasi *SentencePiece* menggunakan dua algoritma yaitu, Byte Pair Encoding (BPE) dan Unigram Languange Model[5]. Oleh karena itu, tokenisasi yang benar dapat membantu meningkatkan kualitas pada pemrosesan text khususnya pada segmentasi.

Topik dan Batasannya

Dari latar belakang yang sudah dipaparkan, topik serta batasan yang diangkat dalam penelitian ini sebagai berikut :

1. Kamus Data

   Batasan kamus data yang diangkat berupa kumpulan kosakata berbagai artikel berita dalam bahasa Indonesia di susun menjadi sebuah dataset. Diambil dari riset Information and Language Processing Systems (ILPS) [9]. Membutuhkan setidaknya lebih dari 3 juta kata dalam dokumen agar dapat meningkatkan kualitas proses pelatihan dalam tokenisasi.

2. Input dan Output

   Inputan dari sistem berupa kalimat bahasa Indonesia menggunakan huruf kecil tanpa spasi yang akan diprediksi peluang penebakan kata yang benar berdasarkan dataset yang sudah ada. Output dari sistem berupa hasil subword tokenisasi pada kalimat. Contoh input dan output dapat dilihat pada tabel 1:

Tabel 1. Contoh Input dan Output Sistem

Input Kata	Gold Standar	Output
pemerintaharusbertindak	[’pemerintah’,’harus’,’bertindak’]	’pemerintah’,’harus’,’bertindak’
penanganansangatlambat	[’penanganan’,’sangat’,’lambat’]	’penanganan’,’sangat’,’lambat
sedangmelakukanrapat	[’sedang’,’melakukan’,’rapat’]	’sedang’, ’melakukan’,’rapat’

3. Rencana pengujian Sistem

   Proses pengujian menggunakan model implementasi SentencePiece berbasis neural network menggunakan algoritma Byte Pair Encoding (BPE) dan Unigram Language Model pada bahasa Indonesia. Adapun hasil evaluasi menggunakan akurasi measure diantaranya adalah Precision, Recall, F-Measure untuk mencapai hasil yang optimal.

Tujuan

Adapun tujuan dari tugas akhir ini adalah untuk mengimplementasikan model SentencePiece pada bahasa Indonesia. Menggunakan sebuah algoritma Byte Pair Encoding (BPE) dan Unigram Language Model pada kalimat tanpa spasi kemudian memisahkan tanda baca dan mambagi token menjadi unit subword untuk mendapatkan hasil tokenisasi yang optimal. Mengevaluasi algoritma BPE dan Unigram LM dalam pelatihan ini.

Organisasi Tulisan

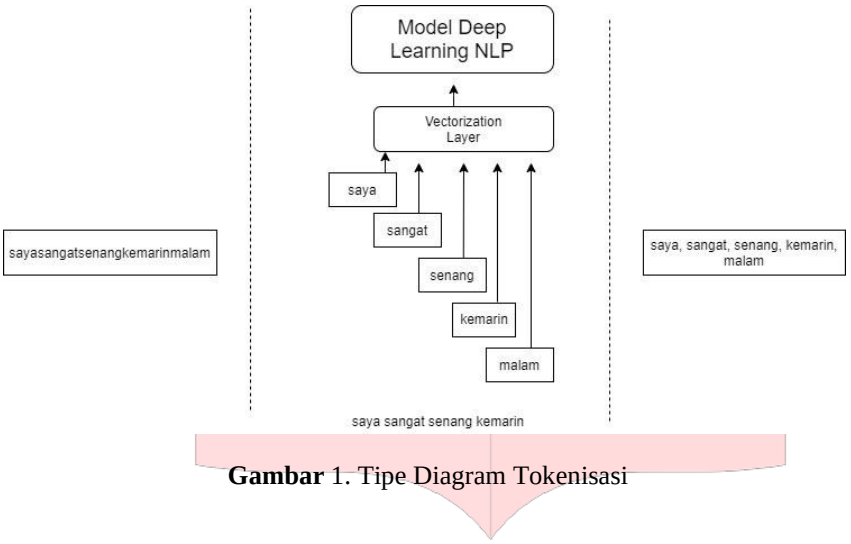
Susunan dari tulisan ini adalah sebagai berikut : Bagian 2 membahas tentang teori, studi dan literatur terkait dengan tulisan ini. Bagian 3 akan dijelaskan sistem yang akan dibangun. Bagian 4 akan terdapat hasil dan evaluasi dari sistem yang di bangun. Dan Bagian 5 akan dijelaskan kesimpulan.

2. Studi Terkait

Bagian ini berisikan hasil kajian teori yang telah dilakukan. Teori yang dicantumkan merupakan daftar referensi dalam pengerjaan penelitian. Teori-teori yang digunkan pada penelitian ini seperti subword tokenisasi, model sistem sentencePiece, algoritma Byte Pair Encoding (BPE), Unigram Languange Model serta akurasimeasure.

2.1 Subword Tokenisasi

Subword tokenisasi adalah proses pemisahan kata pada frasa, kalimat atau seluruh teks dokumen menjadi unit yang lebih kecil dan leksam. Seperti bahasa Indonesia dapat di lakukan dengan memisahkan suatu kalimat pada teks menjadi kata dan tanda baca hanya dengan memecahnya dengan *white space* , tetapi pada teks atau bahasa



Gambar 1. Tipe Diagram Tokenisasi

yang tidak diselingi dengan *white space* situasinya akan lebih rumit.

Pada gambar 1 menunjukkan bahwa jenis token pada teks dapat di identifikasi dengan cara yang berbeda. Pilihan pertama dengan semua kata dikelompokkan menjadi satu token. Pada pilihan kedua yaitu memecah urutan inputan kata menjadi token yang terpisah. Pilihan ketiga menggunakan satu token dengan menambahkan simbol ”,” sebagai pembeda antara tiap kata. Dalam paper (Kudo et al., 2018)[5], Subword tokenisasi mengimplementasikan fitur model dari sentencePiece, subword-nmt, dan wordpiece. Namun, performansi model sentencePiece yang paling baik. Kosakata subword dibuat dengan melatih model dari tokenisasi sentencePiece menggunakan algoritma segmentasi BPE [8] dan unigram [4].

2.2 Byte Pair Encoding (BPE)

Pada bagian ini akan dijelaskan mengenai Byte Pair Encoding (BPE). Pada awalnya digunakan untuk membantu kompresi data dengan menemukan kombinasi pasangan yang paling sering muncul dari byte data yang mana digantikan sebagai satu byte data yang baru berupa simbol atau karakter tertentu. Ini juga dapat diterapkan pada bidang Natural Language Processing (NLP) untuk menemukan cara yang paling efesien dalam merepresentasikan teks. Penerapan pada algoritma BPE yaitu, menginisialisasi simbol kosakata dengan karakter kosakata dan mewakili kata sebagai urutan karakter. Kemudian menghitung semua pasangan simbol secara iteratif dan mengganti setiap kemunculan pasangan yang paling sering (X, Y) dengan simbol XY [8]. Setiap operasi penggabungan menghasilkan simbol baru yang mewakili karakter n-gram. Berikut contoh implementasi dari BPE pada tabel 2:

Tabel 2. Contoh bagaimana BPE memperoleh kosakata diberikan urutan huruf ”aaabaaabac”

Iteration	Sequence	Penggantian
0	aaabaaabac	...
1	ZabZabac	{$Z \leftarrow$aa}
2	ZYZYac	{$Y \leftarrow$ab}
3	XXac	{$X \leftarrow$ZY}

Pada tabel 2 menjelaskan bahwa pasangan byte dari sequence yang paling sering muncul adalah ”aa”, sehingga pasangan byte ”aa” digantikan oleh suatu byte yang tidak digunakan pada dalam data, misalnya dengan karakter ”Z”. Setelah dilakukan pergantian tabel ”aa” oleh ”Z”, maka data menjadi:”*ZabZabac*”. Kemudian dalam hal ini byte data ”ab” juga paling sering muncul, maka dilakukan pergantian dengan suatu byte data yang juga tidak dipakai, misalnya pada karakter ”Y”. Sehingga di peroleh hasil: ”*ZYZYac*”.

Masih memungkinkan kemunculan paling sering pada pasangan byte ”ZY”, yang akan digantikan juga dengan suatu byte yang belum pernah digunakan sebelumnya, misalnya pada karakter ”X”, sehingga diperoleh hasil

"*XXac*". Perlu diperhatikan bahwa algoritma BPE itu adalah algoritma yang mana data pasangan byte yang paling sering muncul akan digantikan dengan suatu byte yang baru sampai data tidak bisa di kompresi lagi karena tidak terdapat pasangan byte yang paling sering muncul. Berikut ini adalah langkah dalam proses trainya yaitu [8]:

1. Mempersiapkan data pelatihan yang cukup besar
2. Menentukan ukuran kosakata subword
3. Split kata menjadi urutan karakter dan menabahkan akhiran simbol ke akhiran kata dengan frekuensi kata.
4. Generate subword baru sesuai dengan kemunculan frekuensi tertinggi
5. Temukan dan ganti pasangan byte yang paling sering kemunculannya pada vocab
6. Mengulangi langkah 4 dan 5 hingga mencapai ukuran kosakata subword pada pasangan frekuensi tertinggi berikutnya.

2.3 Unigram Language Model

Dalam pemrosesan bahasa alami, kami memperkenalkan model sederhana yaitu n-gram yang mana urutan dari n kata. Misalnya pada kata "*bermain*" adalah unigram (n = 1), "*bermain bola*" adalah bigram (n = 2), "*bermain bola bersama*" adalah trigram (n = 3), dan seterusnya. pada proyek ini, hanya fokus pada Unigram LM yaitu kata singular. Model ini bagus untuk melakukan tugas-tugas sederhana seperti indentifikasi bahasa. Metode Unigram LM berbeda dengan BPE[4], model Unigram LM lebih fleksibel karena didasarkan pada model bahasa probabilistik dan dapat menghasilkan beberapa segmentasi berdasarkan probabilitasnya. Unigram diperoleh dengan mencari frekuensi kemunculan huruf dalam suatu dokumen, kemudian frekuensi kemunculan tersebut dapat ditentukan pada probabilitas kemunculan abjad pada suatu dokumen. Berikut ini persamaan berlaku:

$$P(\mathrm{x}) = \prod_{i=1}^{N} p(x_i), \qquad (1)$$

$$\forall i\, x_i \in v,\ \sum_{x \in v} p(x) = 1,$$

Dimana, probabilitas (x_i) adalah bentuk dari $\mathrm{x} = (x_1, \ldots, x_n)$ diberi kamus subword v, untuk setiap subword $x \in v$ memiliki probabilitas $p(x)$. Misalkan $S(X)$ adalah himpunan hasil tokenisasi dari X, hasil tokenisasi yang paling mungkin x^* dipilih sebagai

$$\mathrm{x}^* = \mathrm{argmax}_{x \in S(X)} P(\mathrm{x}) \qquad (2)$$

Dan karena probabilitas kejadian subword tersebut adalah variabel yang tersembunyi, maka digunakan suatu algoritma Expectation Maximization (EM) [4]. Probabilitas setiap kata pada Unigram LM tidak bergantung pada kata apapun sebelumnya. Hanya bergantung pada fraksi waktu kata yang muncul di antara semua kata dalam pelatihan. Kita bisa melangkah lebih jauh dari memperkirakan sebuah probabilitas setiap teks. Bahwa setiap kalimat dalam teks tidak bergantung dengan kalimat lain. Berikut ini adalah langkah dalam proses trainnya [4]:
Ulangi langkah berikut hingga $|V|$ mencapai vocab size yang diinginkan.

1. Mempersiapkan data pelatihan yang cukup besar.
2. menentukan ukuran kosakata.
3. Optimalkan probabilitas kemunculan $(P(x)$ dengan memberikan urutan kata (EM algoritma).
4. Menghitung $loss_i$ untuk setiap subword x_i, dimana $loss_i$ menunjukkan seberapa besar likelihood L berkurang ketika subword x_i dihapus dari kosakata.
5. Urutkan simbol berdasarkan $loss_i$.

2.4 Model Sistem SentencePiece

Dalam paper (Kudo et al., 2018)[5], SentencePiece merupakan sebuah model yang menerapkan metode Unsupervised Text Tokenizer dan Detokenizer khususnya pada neural network-based text generation sistem yang terdiri dari empat komponen yaitu *normalisasi, tranier, encoder dan decoder*[5]. Dimana ukuran kosakata telah ditentukan sebelum pelatihan. Prosesnya sangat cepat, kecepatan segmentasi sekitar 50k kalimat/detik dan penggunaan

memori sekitar 6MB [5]. Karena sentencePiece merupakan multiple subword algorithms, model ini support pada algoritma segmentasi unit subword yaitu byte-pairencoding (BPE) dan Unigram Language Model[5]. SentencePiece membutuhkan kosakata kamus dengan jumlah yang besar dalam proses pelatihan dan segmentasi.

2.5 Akurasi Measure

Untuk mengetahui akurasi terhadap hasil pengujian pada tokenisasi, diperlukan beberapa metode dalam mengevaluasi performansi algoritma dari Machine Learning (ML) khususnya unsupervised learning yaitu Gold Standard yang melibatkan penelitian dari Pedoman Umum Ejaan Bahasa Indonesia (PUEBI). Serta melibatkan perhitungan measure diantaranya adalah Precision, Recall dan F-Measure .

Precision adalah keakuratan hasil klasifikasi dari seluruh dokumen oleh sistem, sehingga dapat diketahui apakah kategori data yang diklasifikasi sesuai dengan kategori yang sebenarnya [7]. Precision dihitung dari jumlah pengenalan data yang bernilai benar oleh sistem dibagi dengan jumlah keseluruhan pengenalan data yang dilakukan pada sistem yang ditunjukkan dengan rumus 3:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Dimana,

TP = True Positive, mengacu pada kata-kata yang cocok dengan hasil dari sistem dan gold standar.
FP = False Positive, mengacu pada kata-kata yang tidak ditemukan dalam hasil sistem tetapi ditemukan dalam gold standar

Recall menunjukkan tingkat keberhasilan sistem dalam mengenali suatu kategori. Recall dihitung dari jumlah pengenalan data yang bernilai benar oleh sistem dibagi dengan jumlah data yang seharusnya dapat dikenali sistem [7]. Ditunjukkan pada rumus 4:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Dimana,

FN = False Negative, merujuk pada kata-kata yang ditemukan dalam hasil sistem tetapi tidak ditemukan dalam gold standart.

F-measure merupakan gambaran pengaruh relatif antara precision dan recall atau disebut harmonic mean. Peformansi algoritma yang digunakan dapat disimpulkan dari nilai F-measure [7]. F-measure dapat dihitung seperti yang ditunjukkan dengan rumus 5:

$$F\ measure = 2\ x \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

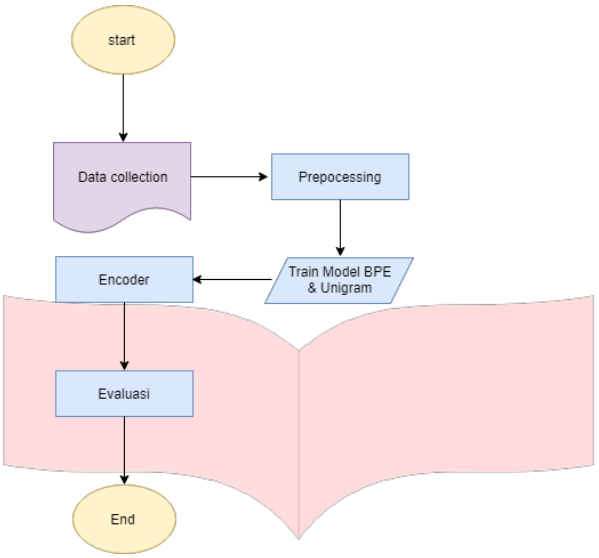
3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

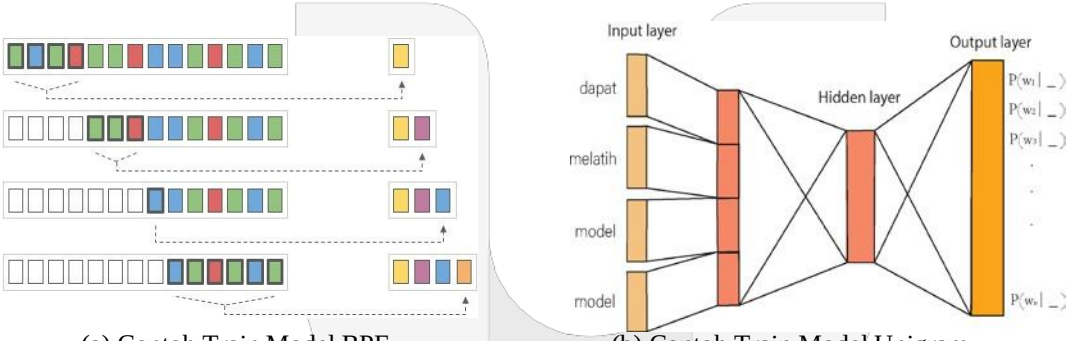
Sistem yang dibangun dengan mengimplementasikan model SentencePiece menggunakan dua algoritma yaitu Byte Pair Encoding (BPE) dan Unigram Language model untuk menghasilkan subword token pada sebuah kalimat. Ditunjukkan Pada gambar 2 adalah alur diagram proses tokenisasi:

Alur pada gambar 2 menunjukkan langkah dalam proses segmentasi. Implementasi pengujian model sentencePiece membutuhkan dataset dalam pelatihan. Kemudian di preprocessing meliputi case folding, menghapus tanda baca, whitepace (karakter kosong), simbol-simbol dan lain sebagainya untuk menghasilkan dataset yang bagus. Setelah preprocessing data, kami dapat melatih model BPE dan Unigram dengan melakukan teknik encode dimana merubah teks dalam bentuk encoder. Hasil terakhir akan dilakukan evaluasi terhadap proses pelatihan algoritma BPE dan Unigram menggunakan teknik measure yaitu precision, recall dan f-measure.

Gambar 3 pada bagian (a).Contoh train model bpe menunjukkan proses train model dengan menemukan substring terpanjang yang tidak tertangani yang ada di vocab dan keluarkan sebagai token, hingga seluruh string telah ditangani. Pada bagian (b) contoh train model unigram menunjukkan contoh train model pada unigram:Sistem A



Gambar 2. Model Sistem Tokenisasi



(a) Contoh Train Model BPE (b) Contoh Train Model Unigram

Gambar 3. Contoh Train Model

menghasilkan *dapat*, sistem B menghasilkan *melatih*, sistem C menghasilkan *model*, sistem D menghasilkan *model*, yang jadi acuannya adalah *model*. Pengodean 1-of-n diterapkan untuk memetakan kata-kata ke inputjaringan saraf yang sesuai.

3.2 Eksperimen Dataset

Kami melakukan eksperimen dataset memproses data teks lebih dari 3 juta kata dalam bahasa Indonesia yang diambil dari riset Information and Language Processing Systems (ILPS) [9]. Berisi berbagai kumpulan teks artikel berita bahasa Indonesia, kata-kata umum, nama, entitas dan sebagainya. Dataset ini akan dijadikan sebagai bahan untuk melatih proses tokenisasi implementasi dari Unsuvervised Learning pada BEP dan Unigram. Sistem tersebut dapat mengenali dan membaca teks sedemikian rupa sehingga dapat belajar dari aksi ini sendiri. Dengan kata lain semakin banyak membaca semakin banyak belajar meningkatkan kemampuan dalam memecah teks menjadi standar unit untuk memprosesnya.

3.3 Eksperimen model BPE

Setelah membuat dataset, kita dapat melatih model BPE sehingga kita berakhir dengan list BPE yang digunakan sebagai encoding. Pada eksperimen ini, kita dapat melatih dengan cukup mudah menggunakan implementasi SentencePiece. Parameter utama yang perlu diperhatikan adalah ukuran kosakata. Semakin besar kosakata yang dimiliki, semakin banyak kata-kata umum yang dapat disimpan. Ukuran kosakata adalah jumlah dari operasi penggabungan BPE dan jumlah karakter dalam data pelatihan. Jumlah operasi penggabungan BPE menentukan apakah urutan simbol yang dihasilkan hanya terdiri dari beberapa operasi penggabungan saja atau lebih banyak operasi penggabungan.

3.4 Eksperimen model Unigram

Sekarang hanya perlu melatih model Unigram dan membandingkan hasil keduanya. Parameter utama hampir sama dengan model BPE yaitu mempertimbangkan ukuran kosakata. Namun model unigram lebih fleksibel karena didasarkan pada bahasa probabilistik dalam regularisasi subword.

4. Evaluasi

4.1 Hasil Pengujian

Hasil dari penelitian ini adalah perbandingan akurasi antara algoritma sentancePiece BPE dan Unigram dengan parameter vocab size yang digunakan untuk proses tokenisasi. Dataset yang digunakan dalam pelatihan ini mencapai lebih dari 3 juta kata. Pengujian ini dilakukan dengan membandingkan kalimat yang tertokenisasi oleh sistem dengan kalimat sebenarnya yang kemudian mendapatkan hasil akurasi. Tabel 3 dan tabel 4 menunjukkan contoh tokenisasi subword kalimat dengan ukuran kosakata yang berbeda.

Tabel 3. Contoh Hasil Output dengan Vocab size pada teks ”yang akanterkenadampak adalahpemerintahdaerah”

Vocab Size	**Segmentasi BPE**
1000	’yang’, ’akan’, ’ter’, ’k’, ’en’, ’ad’, ’amp’, ’ak’, ’adalah’, ’p’, ’em’, ’er’, ’intah’, ’da’, ’erah’
5000	’yang’, ’akan’, ’ter’, ’k’, ’en’, ’ad’, ’ampak’, ’adalah’, ’pemerintah’, ’daerah’
10000	’yang’, ’akan’, ’ter’, ’k’, ’en’, ’ad’, ’ampak’, ’adalah’, ’pemerintah’, ’daerah’
30000	’yang’, ’akan’, ’ter’, ’ken’, ’ad’, ’ampak’, ’adalah’, ’pemerintah’, ’daerah’
40000	’yang’, ’akan’, ’ter’, ’ken’, ’ad’, ’ampak’, ’adalah’, ’pemerintah’, ’daerah’
50000	’yang’, ’akan’, ’ter’, ’ken’, ’ad’, ’ampak’, ’adalah’, ’pemerintah’, ’daerah’
Inputan	yang akanterkenadampak adalahpemerintahdaerah
Vocab Size	**Segmentasi Unigram**
1000	’yang’, ’akan’, ’ter’, ’ke’, ’na’, ’da’, ’mp’, ’ak’, ’adalah’, ’p’, ’em’, ’er’, ’in’, ’ta’, ’h’, ’da’, ’er’, ’ah’
5000	’yang’, ’akan’, ’ter’, ’ke’, ’na’, ’dampak’, ’adalah’, ’pemerintah’, ’daerah’
10000	’yang’, ’akan’, ’ter’, ’kena’, ’dampak’, ’adalah’, ’pemerintah’, ’daerah’
30000	’yang’, ’akan’, ’ter’, ’kena’, ’dampak’, ’adalah’, ’pemerintah’, ’daerah’
40000	’yang’, ’akan’, ’ter’, ’kena’, ’dampak’, ’adalah’, ’pemerintah’, ’daerah’
50000	’yang’, ’akan’, ’terkena’, ’dampak’, ’adalah’, ’pemerintah’, ’daerah’
Inputan	yang akanterkenadampak adalahpemerintahdaerah

Tabel 4. Contoh Hasil Output dengan Vocab size pada teks ”dilarangmemindahkanbukuinidaritempatnya”

Vocab Size	**Segmentasi BPE**
1000	’dil’, ’arang’, ’m’, ’em’, ’ind’, ’ah’, ’kan’, ’b’, ’uku’, ’in’, ’id’, ’ari’, ’t’, ’empat’, ’nya’
5000	’dil’, ’arang’, ’mem’, ’indahkan’, ’b’, ’uku’, ’in’, ’id’, ’ari’, ’t’, ’empat’, ’nya’
10000	’dilarang’, ’mem’, ’indahkan’, ’buku’, ’in’, ’id’, ’ari’, ’tempat’, ’nya’
30000	’dilarang’, ’mem’, ’indahkan’, ’buku’, ’in’, ’id’, ’ari’, ’tempat’, ’nya’
40000	’dilarang’, ’mem’, ’indahkan’, ’buku’, ’inid’, ’ari’, ’tempat’, ’nya’
50000	’dilarang’, ’mem’, ’indahkan’, ’buku’, ’inid’, ’ari’, ’tempat’, ’nya’
Inputan	dilarangmemindahkanbukuinidaritempatnya
Vocab Size	**Segmentasi Unigram**
1000	’di’, ’la’, ’rang’, ’me’, ’min’, ’dah’, ’kan’, ’bu’, ’ku’, ’in’, ’i’, ’dar’, ’it’, ’empat’, ’nya’
5000	’dilarang’, ’me’, ’min’, ’dah’, ’kan’, ’bu’, ’ku’, ’ini’, ’dari’, ’tempat’, ’nya’
10000	’dilarang’, ’mem’, ’indah’, ’kan’, ’buku’, ’ini’, ’dari’, ’tempat’, ’nya’
30000	’dilarang’, ’mem’, ’indah’, ’kan’, ’buku’, ’ini’, ’dari’, ’tempat’, ’nya’
40000	’dilarang’, ’mem’, ’indah’, ’kan’, ’buku’, ’ini’, ’dari’, ’tempat’, ’nya’
50000	’dilarang’, ’me’, ’mindahkan’, ’buku’, ’ini’, ’dari’, ’tempat’, ’nya’
Inputan	dilarangmemindahkanbukuinidaritempatnya

4.2 Analisis Hasil Pengujian

Hasil akurasi yang diperoleh dari pengujian dapat dilihat pada tabel 5 Pada tabel tersebut terlihat perbedaan akurasi yang diperoleh untuk setiap algoritma. Algoritma BPE mendapatkan akurasi tertinggi sebesar 52.9% pada saat vocab size sebesar 50000 sedangkan untuk algoritma Unigram diperoleh akurasi tertinggi 84.7% pada saat vocab size 50000. Dapat ditarik kesimpulan bahwa ukuran vocab size dapat mempengaruhi hasil akurasi dari setiap algoritma dan pada penelitian agoritma Unigram lebih bagus dibandingkan dengan algoritma BPE terlihat dari hasil pengujian pada tabel 5.

Tabel 5. Hasil Pengujian

Model	Vocab Size	Precision	Recall	F1 Score
BPE	1000	0.138	0.287	0.187
	5000	0.282	0.452	0.347
	10000	0.352	0.512	0.417
	20000	0.407	0.553	0.469
	30000	0.446	0.581	0.504
	40000	0.466	0.596	0.523
	50000	0.493	0.614	0.546
Unigram	1000	0.198	0.405	0.266
	5000	0.435	0.656	0.523
	10000	0.553	0.742	0.634
	20000	0.674	0.817	0.739
	30000	0.752	0.861	0.803
	40000	0.790	0.880	0.832
	50000	0.841	0.901	0.870

5. Kesimpulan

Berdasarkan dari hasil penelitian menggunakan model sentencePiece pada algoritma BPE dan Unigram (LM). Dapat disimpulkan bahwa algoritma Unigram lebih bagus dibandingkan algoritma BPE dengan menggunakan parameter ukuran vocab size. Dengan menggunakan teknik measure untuk hasil evaluasi akurasi maksimum 54.6% pada algoritma BPE dengan nilai recall maksimum 61.4% dan nilai precision maksimum 49.3%. Sedangkan algoritma Unigram (LM) menghasilkan akurasi maksimum 87.0% dengan nilai recall maksimum 90.1% dan nilai precision maksimum 84.1%. Sehingga dapat di simpulkan bahwa ukuran vocab size terhadap algoritma BPE dan Unigram dapat mempengaruhi hasil akurasi, masing-masing.

Daftar Pustaka

[1] N. Darheni. Dinamika perkembangan kosakata bahasa indonesia ditinjau dari aspek pemaknaan jurnal sosio-teknologi edisi 23 tahun 10, agustus 2011 1117 dinamika perkembangan kosakata bahasa indonesia ditinjau dari aspek pemaknaan. *Jurnal Sosioteknologi*, 10(23):1117–1128, 2011.

[2] T. P. P. B. Indonesia. Pedoman umum ejaan bahasa indonesia. *Jakarta: Badan Pengembangan dan Pembinaan Bahasa*, 2016.

[3] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014. [4]

T. Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[5] T. Kudo and J. Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[6] L. Liu and K. Jia. Detecting spam in chinese microblogs-a study on sina weibo. In *2012 Eighth International Conference on Computational Intelligence and Security*, pages 578–581. IEEE, 2012.

[7] A. A. Puspitasari. *Klasifikasi Dokumen Tumbuhan Obat Menggunakan Metode Improved K-Nearest Neighbor*. PhD thesis, Universitas Brawijaya, 2017.

[8] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[9] F. Tala. The impact of stemming on information retrieval in bahasa indonesia. *Proc. CLIN, the Netherlands, 2003*, 2003.

[10] Y. Wang, M. Huang, X. Zhu, and L. Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[11] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.