Journal of Information System, Graphics, Hospitality and Technology
Volume 03, Nomor 02, Oktober 2021
P-ISSN : 2622-1594
E-ISSN : 2685-449X

Ekstraksi Relasi Antar Entitas di Bahasa Indonesia Menggunakan Neural Network

Ananta Tio Putra[a], Eunike A. Kardinata[b], Hartarto Junaedi[b], Francisca H. Chandra[a], Joan Santoso[a]
[a]*Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
[b]*Departemen Sistem Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
E-mail: anantatio.01@mail.com, eunike@stts.edu, aikawa@stts.edu, fhc@stts.edu, joan@stts.edu

***Abstrak*—Dengan perkembangan zaman yang begitu pesat, berdampak pada perkembangan data pula. Salah satu bentuk data yang paling banyak saat ini berupa data tekstual seperti artikel sederhana maupun dokumen lain yang terdapat di internet. Agar data tekstual tersebut dapat dimengerti dan dimanfaatkan dengan baik oleh manusia, maka perlu di proses dan disederhanakan agar menjadi informasi yang ringkas dan jelas. Oleh karena itu, semakin berkembang pula penelitian dalam bidang *Information Extraction* (IE) dan salah satu contoh penelitian di IE adalah *Relation Extraction* (RE). Penelitian RE sudah banyak dilakukan terutama pada Bahasa Inggris dimana *resourcenya* sudah termasuk banyak. Metode yang digunakan pun bermacam-macam seperti *kernel, tree kernel, support vector machine, long short-term memory, convulution recurrent neural network*, dan lain sebagainya. Pada penelitian kali ini adalah penelitian RE pada Bahasa Indonesia dengan menggunakan metode *convulution recurrent neural network* yang sudah dipergunakan untuk RE Bahasa Inggris. Dataset yang digunakan pada penelitian ini adalah dataset Bahasa Indonesia yang berasal dari file xml wikipedia. File xml wikipedia ini kemudian diproses sehingga menghasilkan dataset seperti yang digunakan pada CRNN dalam Bahasa inggris yaitu dalam *format SemEval-2 Task 8*. Uji coba dilakukan dengan berbagai macam perbandingan data training dan testing yaitu 80:20, 70:30, dan 60:40. Selain itu, parameter pooling untuk CRNN yang digunakan ada dua macam yaitu '*att*' dan '*max*'. Dari uji coba yang dilakukan, hasil yang didapatkan adalah bervariasi mulai dari mendekati maupun lebih baik bila dibandingkan dengan CRNN dengan menggunakan dataset Bahasa inggris sehingga dapat disimpulkan bahwa dengan CRNN ini bisa digunakan untuk proses RE pada Bahasa Indonesia apabila dataset yang digunakan sesuai dengan penelitian sebelumnya.**

***Kata Kunci*—Convulution Recurrent Neural Network, Information Extraction, Relation Extraction, Bahasa Indonesia.**

I. PENDAHULUAN

Pada zaman sekarang ini, perkembangan data begitu pesat. Salah satu bentuk data yang paling banyak kita jumpai adalah berupa data tekstual yang terdapat pada artikel sederhana, maupun juga data teks lain yang terdapat pada internet [1]. Bisa dikatakan hampir semua data tekstual yang mudah dijumpai merupakan data tekstual yang tidak terstruktur. Agar data tekstual yang begitu besar tersebut dapat dimengerti dan dimanfaat oleh manusia, harus diproses dan disederhanakan terlebih dahulu menjadi informasi yang lebih ringkas dan jelas. Untuk memproses data teks yang begitu besar, diperlukan proses komputerisasi, oleh karena itu semakin banyak penelitian yang dilakukan dalam bidang *text mining* yang biasa dikenal dengan *information extraction* (IE). Salah satu metode penelitian IE yang terkenal adalah *TextRunner* [2]. IE terdiri dari beberapa macam task dan salah satunya adalah *Relation Extraction* (RE). RE ini sendiri bertujuan untuk mengidentifikasi relasi apa saja yang ada pada pasangan entitas yang terdapat pada sebuah kalimat [3]. Pada penelitian RE, metode yang paling populer atau paling sering digunakan adalah metode *rule based* atau *feature based* dimana disediakan *pattern* terlebih dahulu dan kemudian pattern tersebut digunakan untuk melakukan ekstraksi relasi pasangan antar entitas seperti relasi hipernim dan hiponim yang terdapat pada sebuah kalimat [4]. Selain untuk mengetahui relasi apa saja yang terdapat pada sebuah kalimat, RE juga dapat digunakan dibidang kesehatan untuk mengetahui relasi apa dapat diperoleh antar artikel penelitian, ringkasan penelitian, catatan kesehatan dari seorang pasien, penanganan apa saja yang dapat diberikan kepada pasien, dan lain sebagainya [3], [5].

Selain secara *supervised learning* dengan metode rule based, RE dapat dilakukan secara *semi-supervised* dimana hanya memerlukan *pattern* yang lebih sedikit dan kemudian pattern tersebut akan ditraining sehingga mendapatkan pattern-pattern baru. Salah satu contoh penelitian lain dalam RE adalah dengan metode *dependency kernel* dimana relasi yang muncul pada satu kalimat antara dua entitas pertama, masih berhubungan dengan relasi yang muncul antar dua entitas lain yang masih dalam satu kalimat karena apabila direpresentasikan dalam grafik *dependency*, terhubung dengan jarak terpendek [6]–[8]. Selain dengan *dependency kernel* dan menghitung jarak terpendek sehingga dapat menemukan relasi yang lain dalam satu kalimat, penelitian dengan metode *Unrestricted Relation Discovery* juga berperan sama yaitu dapat menemukan relasi-relasi lain dalam teks yang belum diketahui dan masih relevan [9]. Setelah penelitian dengan menggunakan jarak terpendek antar relasi yang terbentuk dalam satu kalimat, terdapat juga penelitian dengan menggunakan *composite kernel,* dimana menggabungkan dua buah *kernel* yang bertujuan untuk memanfaatkan property yang ada pada *kernel* gabungan

Naskah Masuk : 10 Juni 2021
Naskah Direvisi : 3 Agustus 2021
Naskah Diterima : 20 Agustus 2021

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

DOI : 10.37823/insight.v3i02.156

sehingga dapat dihasilkan *knowledge* yang baru pada proses RE [10].

Karena sebagian besar RE memerlukan data dalam jumlah yang besar, terdapat penelitian yang menggunakan metode *distant supervision* dengan memanfaatkan database yang berisi data relasi dalam skala besar yang dikenal dengan *Freebase* [11]. Dengan bantuan *Freebase*, dapat membuat data training dalam jumlah besar pula secara otomatis sehingga menghemat biaya. Selain dengan bantuan Freebase, juga terdapat penelitian yang memperkenalkan metode *snowball* untuk mengekstrak *pattern* dari sebuah teks bahkan tanpa bantuan manusia [12].

Dari berbagai macam penelitian yang telah disebutkan diatas, sebagian besar menggunakan Bahasa Inggris. Pada penelitian kali ini akan membahas RE dengan menggunakan dataset Bahasa Indonesia dimana Bahasa Indonesia sendiri termasuk kategori *low resource language* yang berarti sumber data yang ada untuk penelitian RE masih sangat sedikit. Oleh karena itu, pada penelitian ini akan dilakukan akuisisi atau pembuatan dataset Bahasa Indonesia dalam bentuk format *SemEval-2 Task 8* [13]. Setelah dataset Bahasa Indonesia terbentuk, selanjutnya akan diujicobakan dengan menggunakan metode CRNN [3].

II. Related Works

Penelitian *relation extraction* telah banyak dilakukan guna untuk mendapatkan inti informasi atau rangkuman, hubungan antar dua entitas, dan lain sebagainya yang terdapat dalam sebuah artikel. Untuk mendapatkan hasil yang maksimal, berbagai metode telah digunakan dalam penelitian untuk RE maupun untuk mengklasifikasikan hasil yang didapatkan seperti dengan menggunakan *support vector machine* (SVM), *long short-term memory* (LSTM), kernel, distant supervision dan banyak lagi lainnya. Selain itu, terdapat penelitian RE dengan menggunakan *convulutional neural network* (CNN) maupun modifikasi dari CNN seperti *convulution recurrent neural network* (CRNN) yang bertujuan untuk mengoptimasi hasil yang didapatkan.

Penelitian RE yang menggunakan *kernel* seperti memadukan dua buah individual kernel yaitu *entity kernel* dan *convulution parse tree kernel*, memberikan hasil yang lebih unggul serta bisa menemukan fitur-fitur lain tanpa harus dilakukan secara manual [10]. Pada penelitian sebelumnya, menggunakan penelitian *tree kernel* untuk melakukan komputasi *single kernel* pada contoh-contoh relasi yang ada dimana contoh relasi mengandung *dependency tree* dan terdapat dua entitas. Penelitian ini memanfaatkan jarak terpendek antar dua entitas pada grafik *dependency* juga dapat mengekstrak relasi lain yang tidak diketahui atau terlihat sebelumnya [6]. Selain itu, terdapat juga penelitian RE berbasil *structured parse tree* kernel dimana bisa memberikan hasil yang lebih baik dibandingkan dengan *linier kernel* [14], [15]. Pada penelitian RE untuk Bahasa Indonesia, dilakukan dengan menggunakan *tree kernel* dengan menghitung nilai *similarity* antar *tree* dan secara intuitif, penggunaan tree dengan saling ketergantungan antar struktur dapat memberikan hasil yang akurat karena antar entitas masih saling terkait satu sama lain [7].

Metode selain *kernel* untuk melakukan RE seperti menggunakan model *maximum entropy* dengan mengkombinasikan fitur *lexical*, *syntactic*, dan *semantic* juga dilakukan dan dapat mengekstrak relasi yang lebih banyak meskipun jumlah data yang teranotasi di awal hanya sedikit [16]. Selain itu, penelitian menggunakan *distant supervision* yang dioptimasi dengan cara proses learning pada *multi-instance* dan *multi-label* memberikan hasil yang optimal karena dapat digunakan pada *task* dengan relasi yang belum diketahui atau belum terdapat pada *freebase* [19]. CNN juga digunakan dalam penelitian RE dimana dengan CNN dan RNN dapat menangkap struktur-struktur yang tersembunyi yang tidak dapat diketahui secara langsung. CNN memiliki keunggulan untuk *mengenerate k-gram* secara berturut-turut pada sebuah kalimat sedangkan RNN berfungsi untuk *mengencode* konteks kalimat. Pada penelitian ini mengkombinasikan *feature-based* dengan model *log-linear* dengan CNN-RNN ini [17]. Penelitian lain yang memanfaatkan CNN adalah dengan menambahkan ukuran dari *window-filter* dan *word embedding* yang belum di training sebagai inisialisasi awal memberikan hasil yang baik bila dibandingkan dengan *baseline* system untuk RE dan *state-of-the-art system* untuk RE [18].

Selain penelitian untuk mengekstrak relasi itu sendiri, terdapat pula penelitian yang bertujuan untuk melakukan pengklasifikasian pada hasil yang terekstrak agar dapat digunakan semaksimal mungkin. Pengklasifikasian dapat digunakan dengan menggunakan metode *bootstrapping* [20] dan juga SVM yang dapat mengidentifikasi hubungan antar entitas dan kemudian menetapkan jenis atau tipe semantik mereka dimana matrix dari kemiripan relasi menunjukkan sebuah klasifikasi atau kelompok tertentu [21]. Pada penelitian lain, kombinasi *bootstrapping* pada SVM juga dapat meningkatkan tingkat akurasi pada saat pengklasifikasian [22]. Penelitian berbasis *Attention-Based Bidirectional Long Short-Term Memory* (Att-BLSTM) dilakukan untuk mengatasi masalah informasi dapat muncul di posisi manapun dalam sebuah kalimat [23]. *Convulutional neural network* (CNN) yang dioptimasi menjadi CRNN, CR-CNN, DNN [3], [24], [25].

III. Pembuatan Dataset

Pada penelitian kali ini, akuisisi dataset yang akan digunakan sebagai data training dan testing berasal dari file xml Wikipedia berupa *unlabeled corpus*. File tersebut bisa didapatkan dari Wikipedia[1]. Contoh file dapat dilihat pada tampilan di Gambar 1.

File yang telah didapatkan akan dilakukan proses ekstraksi sehingga terbagi menjadi dua kumpulan yaitu kumpulan *taxobox-infobox* dan kumpulan teks wikipedia. Teks yang didapatkan akan melalui proses *cleaning* serta *preprocessing*. Dari banyak data yang terdapat pada file xml tersebut, yang digunakan adalah data dalam domain hewan, tumbuhan, data personal orang, tempat, dan data organisasi. Proses *cleaning* dilakukan dengan tujuan agar teks terbebas dari noise seperti kurung kotak, kurung kurawal, dan lainnya karena banyak sedikitnya noise dapat mempengaruhi hasil.

[1] https://dumps.wikimedia.org/idwiki. Website penyedia backup file Wikipedia Bahasa Indonesia.(diakses tanggal 1 Maret 2021).

 DOI : 10.37823/insight.v3i02.156

```
<page>
    <title>Anwar Sadat</title>
    <ns>0</ns>
    <id>3</id>
    <revision>
      <id>14714345</id>
      <parentid>13898322</parentid>
      <timestamp>2019-01-30T00:40:33Z</timestamp>
      <contributor>
        <ip>36.72.72.72</ip>
      </contributor>
      <comment>penambahan kata penghubung yang sesuai</comment>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">{{Infobox_President
        |name = Mohammed Anwar Al Sadat &lt;br /&gt; محمد أنور السادات
        |nationality = Al Menofeia, Mesir
        |image = Sadat_Camp_David.jpg
        |order = [[Presiden Mesir]] ke-3
        |term_start = [[20 Oktober]] [[1970]]
        |term_end = [[6 Oktober]] [[1981]]
        |predecessor = [[Gamal Abdel Nasser]]
        |successor = [[Hosni Mubarak]]
        ...
      </text>
      <sha1>mrbudpnyncanur9fk0haxazk52z0yx6</sha1>
    </revision>
</page>
```

Gambar. 1. Contoh Data XML Wikipedia

Setelah *cleaning*, dilakukan *preprocessing* dimana yang dilakukan adalah *sentence extraction* dan *tokenization*. *Sentence extraction* adalah proses ekstraksi kalimat pada suatu dokumen, dan *tokenization* adalah proses untuk memisah setiap kata yang terdapat pada suatu kalimat. Contoh hasil output dapat dilihat di gambar 2.

```
shantaram rajaram vankudre lahir di maharashtra dalam sebuah
keluarga maharashtra

                    ↓

shantaram, rajaram, vankudre, lahir, di, maharashtra, dalam,
sebuah, keluarga, maharashtra
```

Gambar. 2. Contoh Hasil Preprocessing XML Wikipedia

Proses selanjutnya adalah *word embedding* dengan menggunakan *Glove* [26] yang merepresentasikan kata kedalam vektor angka. Selain itu juga dilakukan proses NER (*Named Entity Reconition*) serta *POS Tagging*. Setelah NER dan *POSTag* dilakukan pada kumpulan teks, selanjutnya dilakukan *distant supervision* pada kumpulan taxobox-infobox dengan hasil berupa pasangan entitas yang terdapat pada *taxobox-infobox* tersebut. Kemudian dari pasangan entitas itu akan dicari pada setiap kalimat pada kumpulan teks yang telah diolah sebelumnya sehingga mendapatkan relasi yang ada dalam setiap kalimat tersebut.

Misalnya data wikipedia Anwar Sadat dengan url https://id.wikipedia.org/wiki/Anwar_Sadat. Dari *taxobox-infobox* yang terdapat pada url berikut, jika dicocokkan pada kalimat

"*Jenderal Besar Mohammed Anwar Al Sadat; lahir di Mit Abu Al-Kum, Al-Minufiyah, Mesir, 25 Desember 1918 – meninggal di Kairo, Mesir, 6 Oktober 1981 pada umur 62 tahun) adalah seorang tentara dan politikus Mesir.*"

maka akan ditemukan relasi seperti *'is-a'*, *'birth-date'*, *"birth-place"*, *'death-date'*, *'death-place"*. Data pasangan entitas dan relasi yang ditemukan kemudian disimpan kedalam file tersendiri. Apabila dalam satu kalimat ditemukan beberapa relasi, maka pencatatan akan dilakukan berulang kali karena yang disimpan adalah satu kalimat mewakili satu kalimat dan satu relasi. Contoh salah satu hasil dari proses pembentukan dataset dapat dilihat di Gambar 3.

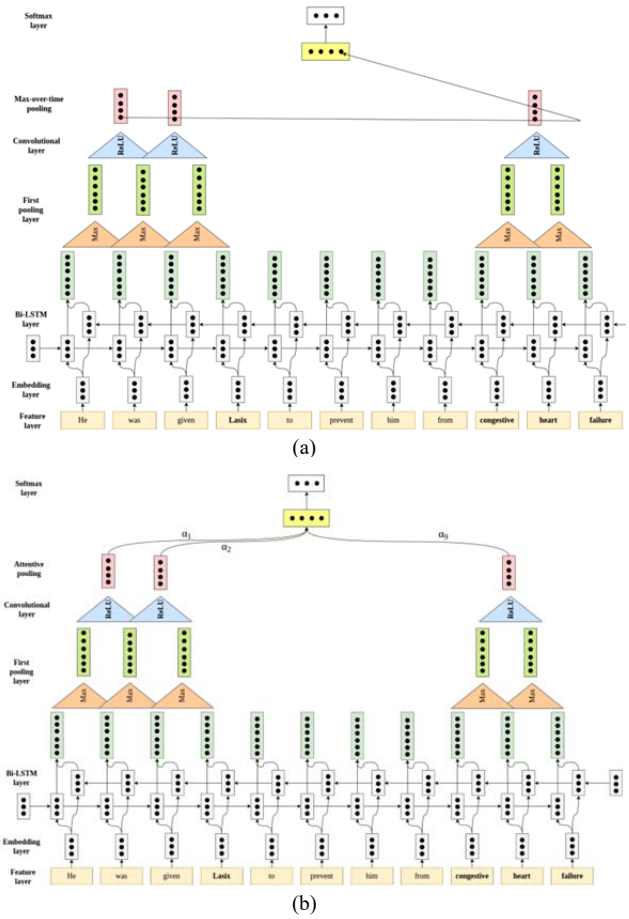
```
<None>  Aaaaba  adalah genus kumbang dari familia  Buprestidae .  Aaaaba adalah genus
kumbang dari familia Buprestidae .  https://id.wikipedia.org/wiki?curid=2519896
/misc/misc/Is-A Is-A  e2,e1
```

Gambar. 3. Contoh Data Preprocessing

Setelah terbentuk file teks yang berisi data kalimat dan relasi yang terbentuk, kemudian dari file tersebut dibentuk sesuai dengan format data *SemEval-2010 Task 8* dan akan diujikan pada model CRNN-Max dan CRNN-Att [3] sebagai usulan metodologi yang dada di penelitian ini.

IV. METODOLOGI PENELITIAN

Metode yang diusulkan dalam penelitian ini menggunakan *Neural Network* yang diusulkan di [3]. Terdapat dua model, yaitu CRNN-Max yang dapat dilihat di Gambar 4 (a) dan CRNN-Att yang dapat dilihat di Gambar 4 (b).



Gambar. 4. (a) Model CRNN-Max dan (b) Model CRNN-Att

Model yang diusulkan terdiri dari beberapa bagian, yaitu *Embedding Layer*, *Recurrent Layer*, *First Pooling Layer*, *Convolutional Layer*, *Second Pooling Layer*, serta *Fully Connected Layer*. Pada *Second Pooling Layer* inilah terdapat dua model yang diusulkan, yaitu *Max Pooling*

DOI : 10.37823/insight.v3i02.156

overtime dan *Attention-based Pooling*. Dari kedua jenis model *pooling layer* ini nantinya yang akan digunakan sebagai pembeda dari masing-masing model. Penjelasan detail dari masing-masing layer akan dijabarkan sebagai berikut.

A. Embedding Layer

Pada *embedding layer*, setiap input kata akan diubah menjadi vector kata yang didapatkan dari model *Word Embedding Glove*[26]. Seluruh kata yang akan digunakan sebagai dari input dari model *Neural Network* akan dilatih dan dilakukan *fine tuning* bersamaan dengan proses pelatihan dari model *Neural Network* yang diusulkan.

Jika ditemukan kata yang tidak ada di model dari Word Embedding, maka akan digunakan sebuah vector random. Vector random ini diinisialisasi secara random dan diupdate secara bersamaan dengan proses training.

B. Recurrent Layer

RNN atau *Recurent Neural Network* telah banyak digunakan dalam proses pelabelan secara *sequential*. Terdapat dua jenis RNN *cell* yang sering digunakan, yaitu GRU dan LSTM. LSTM[27] merupakan *cell* yang digunakan di recurrent layer dalam penelitian ini.

Model RNN yang digunakan dalam penelitian ini adalah menggunakan *Bi-directional* agar diperoleh informasi dari konteks di model yang diusulkan. Jika $h_l^{(t)}$ dan $h_r^{(t)}$ merupakan output dari dari *forward* LSTM dan *backward* LSTM dari *timestep t* maka output dari model Bi-LSTM yang digunakan adalah :

$$z^{(t)} = h_t^{(t)} : h_r^{(t)}, z^{(t)} \in \mathbb{R}^{n_o} \quad (1)$$

Di mana tanda ":" merupakan tanda untuk melakukan penggabungan dari dua buah vector. Setelah didapatkan setiap output LSTM dari masing-masing kata, maka proses akan dilanjutkan untuk digunakan sebagai input dari *First Pooling Layer*.

C. First Pooling Layer

First Pooling Layer menghasilkan *word embedding* di tingkat kata yang menggabungkan informasi dari konteks masa lalu dan masa depan. Terkadang sebuah kata itu sendiri mungkin tidak terlalu memberikan kontribusi yang signifikan untuk merepresentasikan kalimat, sehingga dalam kasus ini, solusinya adalah dengan melakukan ekstraksi fitur terpenting dari frasa pendek menggunakan *pooling*. Jika f1 menyatakan panjang filter yang digunakan untuk *pooling*, dan ($z_1, \ldots, z_m$) adalah barisan vektor yang diperoleh dari lapisan sebelumnya, maka

$$p = (p_1, p_2, \ldots, p_{m-f_{1+1}}) \quad (2)$$

Di mana pi ∈ R^{no} didefinisikan sebagai:

$$p_i = \max_{i \le j \le f_1} [z_{i+j}] \quad (3)$$

Yaitu untuk mendapatkan maximum dari seluruh vektor z_{i+1} s.d. z_{i+f1}.

D. Convolutional Layer

Convolutional digunakan pada *p* untuk mendapatkan *local features* dari masing-masing kalimat. Jika sebuah *convolutional filter* diberikan parameter w_c dengan ukuran n * f2, dimana f2 adalah panjang dari filter, maka output convolution dari sequence masing-masing *convolutional layer* adalah:

$$h_c^i = f(w_c . p^{i:i+f_2-1} + b_c) \quad (4)$$

Dimana i = 1, 2,…, m – f1 – f2 + 2. Fungsi aktivasi yang digunakan adalah *ReLU* dimana f(x) = max({0,x}), dan b_c adalah bias. Parameter w_c dan b_c merupakan *shared parameter* untuk seluruh proses *convolusi* i = 1, 2,…, m – f1 – f2 + 2.

E. Second Pooling Layer

Output dari *convolution layer* terdiri dari berbagai macam panjang (m-f1-f2+2) karena tergantung dari panjang kalimat sebesar *m*. Untuk mendapatkan fixed length dari *global features* untuk seluruh *sequence*. Pada percobaan ini diusulkan dua buah mekanisme, yaitu *max pooling* atau *attention based pooling*. Detail dari penjelasan masing-masing *pooling* layer adalah:

- *Max Pooling Over Time*

Max pooling over time ini diusulkan oleh Collobert, dkk [28]. Layer ini berguna untuk mencari nilai maksimal dari seluruh kalimat dengan asumsi bahwa seluruh informasi yang relevant akan diakumulasikan pada posisi tersebut.

Karena input layer ini adalah *local convolved vectors*, maka strategi yang digunakan adalah melakukan ekstraksi fitur yang sangat penting dari beberapa frasa singkat. Output dari layer ini dapat dilihat di :

$$z_{pool} = \max_{1 \le i \le (m - f_1 - f_2 + 2)} [h_c^i] \quad (5)$$

Dimana z_{pool} adalah *dimension-wise maximum* dari seluruh h^i_c.

- *Attention-based Pooling*

Max Pooling dapat mengalami kegagalan jika informasi yang penting terdistribusi di beberapa klausa lain yang ada di dalam sebuah kalimat. Untuk mengatasi hal ini digunakan sebuah skema *attention-based pooling* dengan mendapatkan fitur optimal dengan melakukan kombinasi bobot liner dari sebuah *vector attention*. Bobot dilatih dengan menggunakan *mekanisme attention* dimana fitur-fitur yang dianggap penting akan memiliki nilai bobot yang lebih besar. *Mekanisme attention* menghasilkan vector α dengan ukuran m – f1 – f2 + 2 dan nilai dari vector untuk setiap frasa diperoleh dari vector fitur hasil convolutional layer. Model matematis dari layer ini adalah:

$$H_{att} = \tanh(W_1^{\propto} H_c)$$
$$\propto = Softmax(W_2^{\propto} H_{att})$$
$$z_{att} = \propto H_c^T \quad (6)$$

Dimana, H_c adalah matriks dari output vector CNN, $W_1\,\alpha$, $W_2\,\alpha$ dengan ukuran n_c x n_c adalah matriks parameter, dan vektor α adalah bobot *attention* dan z_{att} adalah output dari *pooling layer*. Bobot *attention* merupakan bobot pada level kalimat, sehingga nilai α berbeda pada setiap kalimat.

F. Fully Connected dan Softmax

Untuk melakukan klasifikasi dari penentuan label jenis relasi yang akan dikenali, maka akan digunakan sebuah *fully connected layer* dengan jumlah sebesar |C| nodes, dimana C

DOI : 10.37823/insight.v3i02.156

adalah himpunan dari seluruh label relasi yang akan dikenali. Pada *output layer* akan digunakan fungsi *softmax* untuk mendapatkan distribusi *probability* dari sekumpulan label kelas yang mungkin. Output final dapat didefinisikan sebagai:

$$p(c_i|x) = Softmax(W_i^o z + b_i^o) \quad (7)$$

Dimana W^o dan b^o merupakan parameter bobot dan bias dan z adalah hasil dari *second pooling layer*, yaitu apakah z_{pool} atau z_{att} tergantung dari jenis skema *second pooling layer* yang dipilih. Untuk hasil prediksi y' yang diperoleh menggunakan:

$$y' = \underset{c_i \in C}{\arg\max}\, p(c_i|x) \quad (8)$$

V. PERCOBAAN

Dataset Bahasa Indonesia yang telah dibentuk dalam format *SemEval-2010 Task 8* akan diuji coba pada model CRNN dengan perbandingan persentase training dan testing sebesar 80:20, 70:30, dan 60:40. Selain itu, parameter lain yang digunakan adalah filter1 (*f1*) yang bernilai 2, filter1 (*f2*) yang bernilai 5, layer yang berjumlah 100, *learning rate* 0.001, jumlah *epoch* 15, dan tipe *pooling* yang terdiri dari *Max* dan *Att*. Sedangkan untuk pengukuran keberhasilan digunakan *F1-Score* yang telah digunakan di [29].

Percobaan pertama dilakukan dengan menggunakan model dari CRNN dengan menggunakan *attention-based pooling* pada *second pooling layer*. Hasil dari uji coba akan disajikan pada tabel 1.

TABEL I
UJI COBA CRNN-ATT

DATA	CRNN-ATT		
	PRECISION	RECALL	F1 SCORE
Base	64.42	62.14	62.45
80:20	69.48	53.30	55.32
70:30	68.75	71.14	69.53
60:40	70:20	62.27	64.41

Dari hasil ujicoba dengan menggunakan model CRNN-Att diperoleh hasil terbaik ada di *F1-Score* pada pembagian data 70-30. Sedangkan nilai performa terburuk diperoleh pada pembagian distribusi data 80:20.

Percobaan kedua dilakukan percobaan pada model CRNN dengan model *pooling* pada *second pool layer* adalah *Maximum Pooling across time*. Hasil dari ujicoba dapat dilihat di tabel 2.

TABEL 2
UJI COBA CRNN-MAX

DATA	CRNN-MAX		
	PRECISION	RECALL	F1 SCORE
Base	67.91	67.91	64.38
80:20	81.15	81.15	67.28
70:30	82.07	82.07	80.57
60:40	73.61	73.61	74.21

Pada percobaan kedua percobaan terbaik diperoleh pada pembagian data 70-30. Sedangkan untuk percobaan teburuk diperoleh pada pembagian data 80-20.

Dari uji coba yang dilakukan pada model CRNN dengan menggunakan parameter standard seperti yang digunakan pada penelitian sebelumnya, menunjukkan hasil yang baik dari segi *precission*, *recall*, dan *F1 score*. Nilai *F1-Score* terbaik diperoleh pada pembagian data 70-30. Sedangkan nilai *F1-Score* terburuk diperoleh pada pembagian data 80-20. Dari percobaan ini penambahan jumlah training data sebesar 80% dari total keseluruhan data menyebabkan model menjadi *overfitting* dan gagal mengenali relasi yang ada di dalam data test.

VI. KESIMPULAN

Penelitian RE dengan menggunakan model Convulutional Recurrent Neural Network (CRNN) dengan dataset Bahasa Inggris dalam format *SemEval 2010 task 8*, bisa juga dimanfaatkan untuk penelitian RE pada bahasa Indonesia. Performa terbaik diperoleh pada model CRNN-Max dengan nilai *F1-Score* sebesar 80.57. Namun yang perlu diperhatikan adalah bentuk dataset yang digunakan sebagai input untuk model CRNN ini. Dataset bahasa Indonesia yang digunakan pada model CRNN ini bersumber pada file xml wikipedia yang diproses terlebih dahulu dan dibentuk ke dalam format *SemEval 2010 task 8*. Harapan selanjutnya adalah penelitian dapat dikembangkan dengan melakukan optimasi pada CRNN sehingga memberikan hasil lebih baik

UCAPAN TERIMA KASIH / ACKNOLEDGMENT

Ucapan terima kasih diberikan kepada seluruh pihak yang membantu pengerjaan penelitian ini, khususnya Pusat Studi Natural Language Processing ISTTS

DAFTAR PUSTAKA

[1] J S. Brin, "Extracting Patterns and Relations," *World Wide Web Databases*, pp. 172–183, 1999.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 2670–2676, 2007.

[3] D. Raj, S. K. Sahu, and A. Anand, "Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text," *CoNLL 2017 - 21st Conf. Comput. Nat. Lang. Learn. Proc.*, no. CoNLL, pp. 311–321, 2017, doi: 10.18653/v1/k17-1032.

[4] M. N. Nityasya, R. Mahendra, and M. Adriani, "Hypernym-Hyponym Relation Extraction from Indonesian Wikipedia Text," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 285–289, 2019, doi: 10.1109/IALP.2018.8629216.

[5] B. Rink, S. Harabagiu, and K. Roberts, "Automatic extraction of relations between medical concepts in clinical texts," *J. Am. Med. Informatics Assoc.*, vol. 18, no. 5, pp. 594–600, 2011, doi: 10.1136/amiajnl-2011-000153.

[6] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," *HLT/EMNLP 2005 - Hum. Lang. Technol. Conf. Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, no. October, pp. 724–731, 2005, doi: 10.3115/1220575.1220666.

[7] D. S. Esperanti and A. Purwarianti, "Relation extraction using dependency tree kernel for Bahasa Indonesia," *4th IGNITE Conf. 2016 Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2016*, no. 1, 2016, doi: 10.1109/ICAICTA.2016.7803105.

[8] M. Wang, "A Re-examination of Dependency Path Kernels for Relation Extraction," *Proc. Third Int. Jt. Conf. Nat. Lang. Process.*, vol. 2, pp. 841–846, 2008, [Online]. Available: http://www.aclweb.org/anthology-new/I/I08/I08-2119.pdf.

[9] Y. Shinyama and S. Sekine, "Preemptive Information Extraction using Unrestricted Relation Discovery," *HLT-NAACL 2006 - Hum. Lang. Technol. Conf. North Am. Chapter Assoc. Comput. Linguist. Proc. Main Conf.*, no. June, pp. 304–311, 2006, doi: 10.3115/1220835.1220874.

 DOI : 10.37823/insight.v3i02.156

[10] M. Zhang, J. Zhang, J. Su, and G. Zhou, "A composite kernel to extract relations between entities with both flat and structured features," *COLING/ACL 2006 - 21st Int. Conf. Comput. Linguist. 44th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, vol. 1, no. July, pp. 825–832, 2006, doi: 10.3115/1220175.1220279.

[11] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," no. August, p. 1003, 2009, doi: 10.3115/1690219.1690287.

[12] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," *Proc. ACM Int. Conf. Digit. Libr.*, pp. 85–94, 2000.

[13] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, L. Romano, and S. Szpakowicz, "SemEval-2010 Task 8 : Multi-Way Classification of Semantic Relations Between Pairs of Nominals," no. July, pp. 33–38, 2010.

[14] G. D. Zhou, M. Zhang, D. H. Ji, and Q. M. Zhu, "Tree kernel-based relation extraction with context-sensitive structured parse tree information," *EMNLP-CoNLL 2007 - Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn.*, no. June, pp. 728–736, 2007.

[15] R. J. Mooney and R. C. Bunescu, "Subsequence Kernels for Relation Extraction," *Adv. Neural Inf. Process. Syst.*, pp. 171–178, 2006, [Online]. Available: http://papers.nips.cc/paper/2787-subsequence-kernels-for-relation-extraction.

[16] K. Watanabe and D. Bollegala, "A two-step approach to extracting attributes for people on the web," *Proc. WWW 2009 2nd …*, pp. 22-es, 2009, [Online]. Available: http://portal.acm.org/citation.cfm?doid=1219044.1219066%5Cnhttp://www.miv.t.u-tokyo.ac.jp/papers/watanabe-WEPS2009.pdf.

[17] T. H. Nguyen and R. Grishman, "Combining Neural Networks and Log-linear Models to Improve Relation Extraction," 2015, [Online]. Available: http://arxiv.org/abs/1511.05926.

[18] T. H. Nguyen and R. Grishman, "Relation Extraction: Perspective from Convolutional Neural Networks," *Assoc. Comput. Linguist.*, pp. 39–48, 2015, doi: W15-1506.

[19] M. Surdeanu†, J. Tibshirani†, R. Nallapati?, and C. D. Manning†, "Multi-instance Multi-label Learning for Relation Extraction," *Solid State Sci.*, vol. 10, no. 12, pp. 1794–1799, 2008, doi: 10.1016/j.solidstatesciences.2008.01.031.

[20] L. Qian, G. Zhou, F. Kong, and Q. Zhu, "Semi-supervised learning for semantic relation classification using stratified sampling strategy," *EMNLP 2009 - Proc. 2009 Conf. Empir. Methods Nat. Lang. Process. A Meet. SIGDAT, a Spec. Interes. Gr. ACL, Held Conjunction with ACL-IJCNLP 2009*, no. August, pp. 1437–1445, 2009, doi: 10.3115/1699648.1699690.

[21] B. Rink and S. Harabagiu, "UTD: Classifying semantic relations by combining lexical and semantic resources," *ACL 2010 - SemEval 2010 - 5th Int. Work. Semant. Eval. Proc.*, no. July, pp. 256–259, 2010.

[22] Z. Zhang, "Weakly-supervised relation classification for information extraction," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 581–588, 2004, doi: 10.1145/1031171.1031279.

[23] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," *54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Short Pap.*, pp. 207–212, 2016, doi: 10.18653/v1/p16-2034.

[24] C. N. Dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with Convolutional neural networks," *ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf.*, vol. 1, no. 1, pp. 626–634, 2015, doi: 10.3115/v1/p15-1061.

[25] L. Zhang and D. Moldovan, "Chinese relation classification via convolutional neural networks," *Proc. 31st Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2018*, no. 2011, pp. 225–228, 2018.

[26] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In EMNLP. volume 14, pages 1532–1543.

[27] Sepp Hochreiter and J¨urgen Schmidhuber. 1997. Long short-term memory. Neural computation 9(8):1735–1780.

[28] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning. ACM, pages 160–167.

[29] Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.O., Padó, S., Pennacchiotti, M., Romano, L. and Szpakowicz, S., 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*.

 DOI : 10.37823/insight.v3i02.156