



Tersedia online di www.journal.unipdu.ac.id
Unipdu

Terakreditasi S2 – SK No. 34/E/KPT/2018

Halaman jurnal di www.journal.unipdu.ac.id/index.php/register



Parsing struktur semantik soal cerita matematika berbahasa indonesia menggunakan recursive neural network

Parsing the semantic structure of indonesian math word problems using the recursive neural network

Agung Prasetya ^a, Chastine Fatichah ^b, Umi Laili Yuhana ^c

^{a,b,c} Teknik Informatika, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

email: ^a prasetya.agung@gmail.com, ^b chastine@if.its.ac.id, ^c yuhana@if.its.ac.id

INFO ARTIKEL

Sejarah artikel:

Menerima 15 Januari 2019
Revisi 1 Juni 2019
Diterima 1 Juni 2019
Online 27 Juni 2019

Kata kunci:

parsing
pohon biner
Recursive Neural Network
soal cerita
struktur semantik

Keywords:

binary tree
math word problem
semantic structure
parsing
Recursive Neural Network

Style APA dalam menyitasi artikel ini:

Prasetya, A., Fatichah, C., & Yuhana, U. L. (2019). *Parsing struktur semantik soal cerita matematika berbahasa indonesia menggunakan recursive neural network*. Register: Jurnal Ilmiah Teknologi Sistem Informasi, 5(2), 106-115.

ABSTRAK

Soal cerita berperan penting untuk kemajuan pengembangan kecerdasan buatan. Hal ini karena penyelesaian soal cerita melibatkan pengembangan sebuah sistem yang mampu memahami bahasa alami. Pembentukan sistem penyelesaian soal memerlukan mekanisme untuk mendekomposisikan teks soal ke segmen-segmen teks untuk diterjemahkan ke jenis operasi hitung. Segmen-segmen tersebut ditentukan melalui proses *parsing* semantik struktur soal agar menghasilkan segmen-segmen yang maknanya menunjuk operasi hitung. Sejumlah metode usulan saat ini sesuai untuk diterapkan pada soal cerita berbahasa Inggris dan belum diterapkan pada soal cerita berbahasa Indonesia. Dampaknya adalah segmen-segmen yang dihasilkan belum tentu menghasilkan urutan pengerjaan operasi yang sesuai makna cerita. Penelitian ini mengusulkan penggunaan *Recursive Neural Network* (RNN) sebagai parser struktur semantik soal cerita berbahasa Indonesia. Pengujian parser struktur semantik soal dilakukan terhadap soal-soal yang berasal dari Buku Sekolah Elektronik (BSE) Sekolah Dasar (SD) dari Pusat Perbukuan Kementerian Pendidikan dan Kebudayaan. Hasil pengujian menunjukkan akurasi akhir sebesar 86,4%.

ABSTRACT

Math word problems play an important role for the development of artificial intelligent. This is because solving word problems involves the development of a system that can understand natural language. Designing a system for solving math word problems requires a mechanism for decomposing a text into segments of text to be translated into math operation. The segments are categorized through the process of parsing the semantic structure of the word problems to obtain segments whose meanings refer to math operation. A number of current proposed methods are suitable to be applied to English math word problems and have never been applied to Indonesian math word problems. The impact is that the segments produced are not necessarily in line with the sequences of operations appropriate with the meaning of the story. This study proposed the use of Recursive Neural Network (RNN) as a parser of semantic structure of Indonesian math word problems. The testing of the parser was carried out on the math word problems taken from the Elementary School's Electronic School Book (BSE) published by the Book Center of the Ministry of Education and Culture. The result of the testing showed that the final accuracy was 86.4%.

© 2019 Register: Jurnal Ilmiah Teknologi Sistem Informasi. Semua hak cipta dilindungi undang-undang.

1. Pendahuluan

Soal cerita berperan penting untuk kemajuan pengembangan kecerdasan buatan (Clark & Etzioni, 2016; Clark, 2015). Hal ini karena penyelesaian soal cerita melibatkan pengembangan sebuah sistem yang

Parsing struktur semantik soal cerita matematika berbahasa indonesia menggunakan ... <http://doi.org/10.26594/register.v5i2.1537>
© 2019 Register: Jurnal Ilmiah Teknologi Sistem Informasi. Semua hak cipta dilindungi undang-undang.

mampu memahami bahasa alami. Sistem yang dikembangkan tersebut harus dapat memberikan jawaban ketika diberikan sebuah narasi dan pertanyaan.

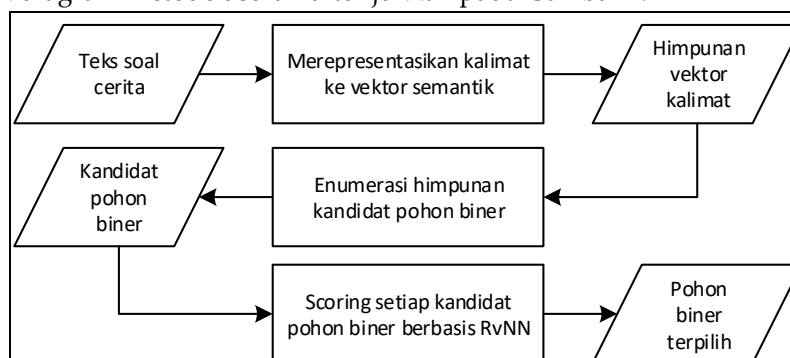
Pembentukan sistem penyelesaian soal memerlukan mekanisme untuk mengidentifikasi jenis operasi hitung. Sejumlah penelitian mengidentifikasi keseluruhan jenis operasi dengan menyeleksi kerangka atau *template* persamaan yang cocok dengan soal (Zhou, Dai, & Chen, 2015 ; Kushman, Artzi, Zettlemoyer, & Barzilay, 2014; Upadhyay, Chang, Chang, & Yih, 2016; Mitra & Baral, 2016; Wang, Liu, & Shi, 2017; Wang, et al., 2018). Mekanisme ini mensyaratkan ketersediaan sebuah repositori yang diasumsikan bahwa *template* pada repositori mencakup berbagai bentuk soal. Mekanisme berbasis *template* ini kurang efektif ketika dihadapkan pada sebuah soal yang persamaannya tidak tersedia pada repositori. Untuk memperbaiki hal ini, sejumlah penelitian mendekomposisikan teks soal terlebih dahulu ke sejumlah segmen teks (Koncel-Kedziorski, Hajishirzi, Sabharwal, Etzioni, & Ang, 2015; Roy, Vieira, & Roth, 2015). Segmen-segmen teks tersebut kemudian diklasifikasikan ke operasi hitung dengan menggunakan model klasifikasi teks. Selain lebih efektif untuk mengidentifikasi jenis operasi pada soal campuran yang kompleks, mekanisme ini lebih efisien karena tidak memerlukan proses pembentukan repositori.

Performa mekanisme identifikasi jenis operasi dengan mendekomposisikan soal sangat dipengaruhi oleh proses penentuan segmen. Penentuan segmen tidak dapat diselesaikan hanya dengan membatasi bagian-bagian tertentu teks menggunakan dua *operand*. Pemecahan teks soal ke segmen perlu memperhitungkan struktur semantik soal agar menghasilkan segmen-segmen yang maknanya menunjuk ke operasi hitung. Tetapi, metode-metode berbasis dekomposisi teks yang telah diusulkan di atas sesuai untuk diterapkan pada soal cerita berbahasa Inggris dan belum tentu sesuai untuk menyelesaikan soal cerita berbahasa Indonesia. Ketika menghadapi kasus soal campuran, hal ini berpotensi menghasilkan urutan pengerjaan operasi yang tidak sinkron dengan makna cerita.

Berdasarkan permasalahan di atas, sebuah metode sangat diperlukan untuk melakukan *parsing* struktur semantik soal berbahasa Indonesia. Proses *parsing* tersebut dimodelkan sebagai permasalahan dalam memprediksi struktur. Makalah ini mengusulkan penerapan *Recursive Neural Network* (RvNN) untuk menyelesaikan prediksi struktur semantik soal bahasa Indonesia. Model RvNN sebelumnya telah berhasil digunakan untuk menyelesaikan masalah *parsing* sintaksis kalimat dan *parsing* struktur *image* (Socher, Lin, Ng, & Manning, 2011; Socher, Pennington, Huang, Ng, & Manning, 2011).

2. Usulan Metode Parsing Struktur Semantik Soal Cerita Berbahasa Indonesia

Metode usulan untuk *parsing* struktur semantik soal cerita tersusun atas tiga proses utama, yaitu merepresentasikan kalimat ke vektor semantik, enumerasi himpunan kandidat pohon biner, dan *scoring* setiap kandidat pohon biner berbasis RvNN. Proses pertama mengambil masukan berupa teks soal cerita. Proses ini akan memecah soal ke dalam beberapa kalimat. Setiap kalimat akan direpresentasikan ke vektor berdimensi n menggunakan model *Recursive Autoencoder*. Proses yang kedua berupaya mendapatkan berbagai kemungkinan pohon biner yang dapat dibentuk dengan diberikan sejumlah kalimat. Proses yang terakhir menghitung skor dan memilih skor maksimum dari kandidat-kandidat pohon biner. Blok diagram metode usulan ditunjukkan pada Gambar 1.



Gambar 1. Blok diagram metode usulan *parsing* struktur semantik soal cerita

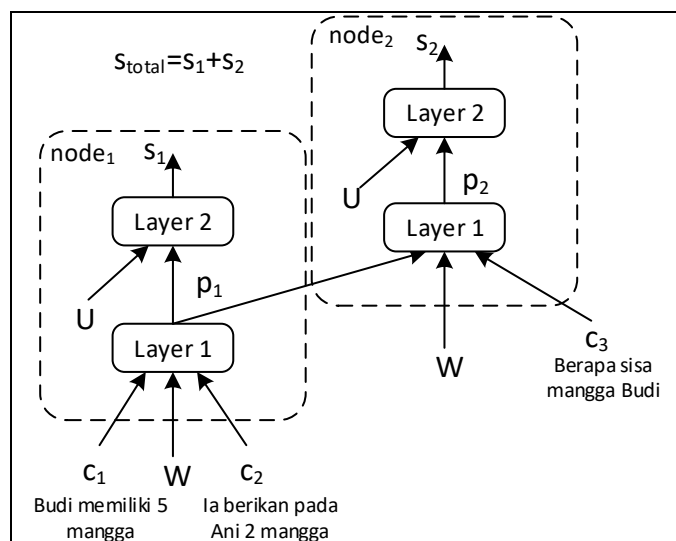
Parsing struktur soal berupaya untuk mendapatkan pohon biner yang merepresentasikan struktur makna soal cerita. Penelitian ini menerapkan model RvNN (Socher, Lin, Ng, & Manning, 2011; Socher, Huang, Pennington, Ng, & Manning, 2011; Socher, Pennington, Huang, Ng, & Manning, 2011)

untuk menghitung skor dari berbagai kemungkinan bentuk pohon biner dari sebuah soal. RvNN diintegrasikan pada sebuah fungsi prediksi f . Fungsi prediksi f akan memilih sebuah pohon biner yang maksimum. Dengan demikian, fungsi prediksi f harus dibentuk agar menghasilkan skor yang tinggi untuk pohon biner yang benar dan memberikan skor yang rendah untuk pohon biner-pohon biner yang salah. Diberikan sebuah teks soal x yang tersusun atas n -kalimat dan parameter RvNN θ , fungsi prediksi f dapat didefinisikan pada Persamaan 1,

$$f(x, \theta) = \arg \max_{t \in T(x)} RvNN(t, \theta) \quad (1)$$

yang mana $T(x)$ merupakan himpunan pohon biner yang dapat dibentuk dari soal x .

RvNN pada prinsipnya merupakan sebuah arsitektur *Neural Network* yang terdiri dari dua buah *layer* yang masing-masing menggunakan fungsi aktivasi *nonlinear* dan *linear*. Dua buah *layer* ini melekat pada setiap *node* nonterminal pada pohon biner. RvNN hanya dapat bekerja jika struktur pohon biner telah terbentuk. Gambar 2 menunjukkan contoh penggunaan RvNN untuk menghitung skor sebuah pohon biner dari soal "Budi memiliki 5 mangga. Ia memberikan pada Ani 2 mangga. Berapa sisa mangga Budi?".



Gambar 2. Contoh penerapan RvNN untuk menghitung skor sebuah pohon biner dari soal

Seperti yang ditunjukkan pada contoh di atas, skor total s_{total} sebuah pohon biner diperoleh dengan menjumlahkan skor pada keseluruhan *node*. Skor *node* s merupakan *output* dari fungsi *linear* pada *layer* 2. Fungsi *linear* pada *layer* 2 mengambil dua jenis masukan, yaitu parameter U dan vektor p . Vektor p yang berdimensi n merupakan *output* fungsi *nonlinear* pada *layer* 1. Fungsi *nonlinear* pada *layer* 1 mengambil tiga masukan salah satunya adalah parameter W . Dua masukan lain adalah dua vektor kalimat jika *layer* 1 berada pada *node leaf*. Jika *layer* 1 berada pada *node nonleaf* maka masukan dapat berupa vektor *output layer* 1 yang berasal dari *node child*. Pada RvNN, setiap *node* berbagi parameter *layer* 1 dan *layer* 2 yang sama. Proses perhitungan skor total dilakukan secara *bottom-up* dimulai dari *node leaf* hingga ke *root*.

Perhitungan skor total hanya dapat dilakukan jika kalimat-kalimat soal telah direpresentasikan sebagai vektor berdimensi n . Untuk mengubah kalimat ke vektor, penelitian ini menggunakan *Recursive Autoencoder* (Socher, et al., 2013). Vektor dari hasil proses penerapan *Recursive Autoencoder* merepresentasikan semantik kalimat. Mekanisme ini memanfaatkan vektor kata dan struktur sintaks kalimat untuk mendapatkan sebuah vektor yang merepresentasikan makna kalimat. Untuk merepresentasikan kata ke vektor, penelitian ini menggunakan model *Skip-Gram* (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Sedangkan struktur sintaks kalimat diperoleh dengan menggunakan algoritma *Shift-Reduce-Parser* (Zhu, Zhang, Chen, Zhang, & Zhu, 2013).

Fungsi prediksi pada Persamaan 1 memerlukan mekanisme untuk mendapatkan himpunan pohon biner dari sebuah soal. Pada penelitian ini, himpunan tersebut diperoleh dengan cara mengenumerasi semua kemungkinan bentuk pohon biner yang dapat dibentuk dari sebuah soal.

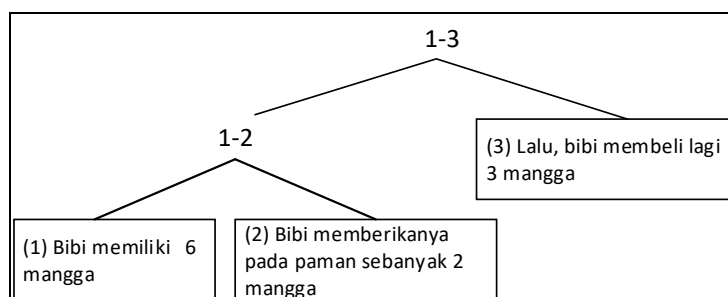
Walaupun cara ini kurang efisien, jenis soal yang digunakan pada penelitian ini relatif sederhana, tersusun atas 2–4 kalimat sehingga tidak mengalami kendala. Penelitian selanjutnya dapat mempertimbangkan mekanisme lain yang lebih efisien untuk pembentukan himpunan pohon biner.

Parameter θ pada Persamaan 1 merujuk pada parameter *layer 1* dan *layer 2*, U dan W . Parameter ini perlu dioptimalkan agar menghasilkan skor total yang tinggi pada pohon biner yang benar dan skor total yang rendah untuk pohon biner yang salah. Optimasi parameter θ dilakukan dengan *supervised learning*. Proses *learning* berupaya menentukan parameter θ dengan menggunakan sejumlah *dataset* yang terdiri dari m -pasangan soal x dan pohon biner soal $t, \{(x_i, t_i)\}_{i=1}^m$. Fungsi objektif untuk proses *learning* ditunjukkan pada Persamaan 2 (Socher, Lin, Ng, & Manning, 2011; Ratlif, Bagnell, & Zinkevich, 2006),

$$J(\theta) = \frac{\lambda}{2} \|\theta\|^2 + \frac{1}{m} \sum_{i=1}^m \max_{t \in T(x_i)} \{l(t_i, t) + RvNN(t, \theta)\} - RvNN(t_i, \theta) \quad (2)$$

yang mana λ merupakan bobot *regularizer* dan l merupakan fungsi *structured loss margin* yang mengukur perbedaan struktur dua buah pohon biner. Proses *learning* akan mengubah nilai θ agar RvNN dapat menghasilkan skor yang lebih tinggi untuk pohon biner yang benar (pohon biner yang disediakan pada *dataset*) t_i dibandingkan dengan skor pohon biner yang salah $t \in T(x_i) \setminus t_i$. Perbedaan nilai skor kedua pohon biner tersebut mengikuti nilai fungsi l .

Mengikuti Taskar, Klein, Collins, Koller, dan Manning (2004), fungsi l ditentukan berdasarkan jumlah perbedaan *index span* antara dua pohon. *Span* merupakan pasangan *index* yang menunjukkan posisi kalimat paling kiri dan kanan yang ada di bawah sebuah *node nonleaf*. Fungsi l menjumlahkan total *index span* yang berbeda di semua nonterminal *node*. *Index span* yang berbeda bernilai 1, jika sama maka bernilai 0. Gambar 2 mengilustrasikan pemberian *index span* di setiap *node* pada sebuah pohon biner.



Gambar 3. *Index span* pada sebuah pohon biner untuk soal

3. Hasil Uji Coba dan Pembahasan

3.1. Dataset

Penelitian ini menggunakan soal-soal cerita di Buku Sekolah Elektronik (BSE) yang disediakan oleh Pusat Perbukuan Kementerian Pendidikan dan Kebudayaan. BSE tersebut diunduh dari [website http://bse.kemdikbud.go.id/](http://bse.kemdikbud.go.id/). Penelitian ini menggunakan soal-soal yang mengandung maksimal dua operasi hitung dasar (penjumlahan, pengurangan, perkalian, dan pembagian). Selain itu, penelitian ini mengabaikan soal-soal yang memerlukan proses konversi satuan dan mengabaikan soal-soal yang memerlukan informasi tambahan untuk menyelesaikannya. Tabel 1 menunjukkan jumlah soal cerita yang telah dikumpulkan dari beberapa BSE kelas 1–4 sekolah dasar.

Tabel 1. Informasi detail mengenai *dataset*

Jenis Karakteristik Dataset	Nilai
Jumlah total soal	1212
Jumlah soal campuran	308
Jumlah soal noncampuran	904
Jumlah soal dengan jumlah kalimat < 3	167

Setiap soal selanjutnya disusun ke struktur pohon biner. Setiap *node* nonterminal memiliki tepat 2 cabang. Setiap *node* baik terminal maupun *leaf* memiliki salah satu label dari 5 jenis label. Label-label ini merepresentasikan jenis operasi hitung dasar yang antara lain penjumlahan, pengurangan,

perkalian, pembagian, dan nonoperator. Proses pembentukan struktur pohon biner soal dibantu dengan sebuah aplikasi untuk pembentukan *Rhetorical Structure*, yaitu *RST Tool* versi 3.0 (O'Donnell, 2000).

Ada beberapa pemrosesan teks kalimat sebelum proses pembentukan struktur pohon biner, antara lain: 1) mengganti setiap token angka, 2) tanggal atau waktu menjadi token khusus yaitu *num*, *date* atau *time*, 3) mengubah setiap huruf kapital ke huruf kecil, dan 4) menghapus setiap tanda baca kecuali tanda penghubung “-” untuk kata jamak.

Soal-soal yang menjadi data latih parser struktur semantik memiliki lebih dari 2 kalimat. Soal-soal yang hanya tersusun atas 1 atau 2 kalimat menyebabkan bias dalam proses *training*, karena hanya ada satu kemungkinan struktur yang dapat dibentuk. Informasi mengenai jumlah *dataset* tersaji pada Tabel 1.

Hasil pengumpulan *dataset* menunjukkan ada 7 jenis pola pohon biner yang dapat dikumpulkan. Pola-pola ini sudah meliputi soal cerita yang tersusun atas dua kalimat. Pola-pola ini dapat dianggap sebagai kelas-kelas RvNN. Jumlah soal untuk masing-masing pola ditunjukkan pada Tabel 2. Sedangkan contoh soal cerita yang tergolong pola 1 dan pola 3 ditunjukkan pada Gambar 4 dan Gambar 5.

Tabel 2. Jenis dan jumlah soal

Jenis Pola	Jumlah Pola
Jumlah Pola 1	142
Jumlah Pola 2	166
Jumlah Pola 3	161
Jumlah Pola 4	124
Jumlah Pola 5	145
Jumlah Pola 6	230
Jumlah Pola 7	244

```
<node mathoperation="penjumlahan" type="root">
  <node mathoperation="nooperation" type="non_terminal">
    <node mathoperation="nooperation" type="leaf" id="0">adi punya 5 topi</node>
    <node mathoperation="nooperation" type="leaf" id="1">diberi paman 2 topi</node>
  </node>
  <node mathoperation="nooperation" type="leaf" id="2">berapa topi adi sekarang</node>
</node>
```

Gambar 4. Contoh soal cerita yang tergolong pola 1

```
<node mathoperation="nooperation" type="root">
  <node mathoperation="nooperation" type="leaf" id="0">ani akan membuat kue lapis</node>
  <node mathoperation="perkalian" type="non_terminal">
    <node mathoperation="nooperation" type="non_terminal">
      <node mathoperation="nooperation" type="leaf" id="1">ani membeli 2 kantong telur</node>
      <node mathoperation="nooperation" type="leaf" id="2">setiap kantong berisi 5 butir telur</node>
    </node>
    <node mathoperation="nooperation" type="leaf" id="3">berapa jumlah telur yang ani beli</node>
  </node>
</node>
```

Gambar 5. Contoh soal cerita yang tergolong pola 3

3.2. Hasil Uji Coba Parsing Struktur Soal Cerita Berbahasa Indonesia

Penentuan parameter *Recursive Neural Network* (W, b, U) untuk parser dilakukan melalui *supervised learning*. Sebelum melakukan pelatihan, pada penelitian ini, setiap kalimat direpresentasikan ke sebuah vektor menggunakan model *Recursive Autoencoder* (Socher, et al., 2013). *Recursive Autoencoder* memerlukan mekanisme untuk mengubah kata ke vektor. Penelitian ini menggunakan model *Skip-Gram* untuk mengubah kata ke vektor. Pengujian yang dilakukan mempertimbangkan dimensi vektor kata sebesar 50, 100, dan 200.

Penelitian ini menerapkan *k-fold cross validation* dengan $k = 5$ untuk mengurangi bias dalam mengevaluasi penentuan parameter RvNN. Data yang digunakan untuk data pelatihan maupun pengujian ditentukan dengan membagi terlebih dahulu soal-soal pada setiap pola sebanyak jumlah *fold*. Pembagian ini dilakukan secara acak. Setelah soal-soal per pola terkelompokkan, selanjutnya untuk setiap *fold*, kelompok-kelompok per pola tersebut digabungkan. Beberapa parameter *learning* perlu dioptimalkan pada proses pelatihan parser antara lain bobot *regularizer* λ dan nilai penalti *structure loss* Persamaan 1 pada Persamaan 3. Ukuran optimal suatu nilai parameter didasarkan pada tingkat akurasi parser. Pohon biner yang diprediksi oleh parser RvNN bernilai benar jika strukturnya secara tepat sama dengan hasil struktur pohon biner yang disediakan oleh *dataset*.

Nilai-nilai dari *hyperparameter* ini dioptimalkan melalui *batch subgradient descent*. Pengaturan nilai *learning rate* di setiap iterasi dilakukan dengan menerapkan *rule Constant Step Length*. *Rule Constant Step Length* menentukan nilai *learning rate* untuk setiap iterasi i dengan $\alpha_i = \gamma / \|g_i\|_2$, di mana g_i merupakan *gradient parameter* untuk iterasi ke- i dan γ merupakan sebuah koefisien dengan $\gamma > 0$. Pada penelitian ini, nilai koefisien γ ditetapkan sebesar 0,8. Berdasarkan eksperimen awal yang telah dilakukan, *rule* ini memiliki tingkat *convergence* yang lebih cepat dibandingkan *rule* lain untuk kasus ini. Tabel 3 menunjukkan beberapa parameter yang akan dioptimalkan pada proses pelatihan.

Tabel 3. Parameter RvNN yang diuji coba

Nama Parameter	Nilai-nilai yang diuji coba
Nilai bobot L2-regularizer (λ)	1e-6, 1e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2
Nilai penalti <i>structure loss</i> (κ)	1,0; 0,9; 0,8; 0,7; 0,6
Dimensi vektor kata	50, 100 dan 200

Subgradient descent belum tentu menghasilkan nilai fungsi objektif yang menurun untuk iterasi berikutnya. Hal ini karena penggunaan operasi *max* pada fungsi objektif. Pada setiap iterasi, parameter RvNN yang terbaik akan disimpan sebagai parameter global yang optimal jika fungsi objektif yang dihasilkan pada suatu iterasi lebih optimal daripada iterasi-iterasi sebelumnya.

Uji coba pertama adalah untuk mengetahui pengaruh nilai bobot regularizer λ terhadap rata-rata akurasi. Pada skenario ini nilai penalti *structure loss* κ dan dimensi vektor kata ditetapkan sebesar 0,8 dan 50. Hasil uji coba untuk skenario pertama ditunjukkan pada Tabel 4.

Tabel 4. Hasil akurasi untuk beberapa nilai L2-regularizer dengan nilai penalti sebesar 0,8 dan vektor kata 50

Bobot L2-regularizer (λ)	Akurasi fold 0	Akurasi fold 1	Akurasi fold 2	Akurasi fold 3	Akurasi fold 4	Rata-rata Akurasi
1e-6	0,893	0,889	0,857	0,833	0,764	0,847
1e-5	0,887	0,878	0,857	0,839	0,779	0,848
1e-4	0,881	0,896	0,857	0,842	0,807	0,856
5e-4	0,884	0,898	0,851	0,836	0,793	0,852
1e-3	0,875	0,893	0,863	0,845	0,787	0,852
5e-3	0,893	0,890	0,859	0,835	0,764	0,848
1e-2	0,890	0,890	0,854	0,857	0,776	0,853

Tabel 5. Hasil akurasi untuk beberapa nilai penalti *structure loss* dengan bobot L2 sebesar 1e-4 dan vektor kata 50

Nilai Penalti <i>Structure loss</i> (κ)	Akurasi fold 0	Akurasi fold 1	Akurasi fold 2	Akurasi fold 3	Akurasi fold 4	Rata-rata Akurasi
0.9	0.893	0.893	0.857	0.833	0.784	0.852
0.8	0.881	0.896	0.857	0.842	0.807	0.856
0.7	0.901	0.890	0.848	0.836	0.784	0.852
0.6	0.878	0.886	0.859	0.848	0.801	0.854

Tabel 4 menunjukkan hasil rata-rata akurasi terbaik diberikan ketika nilai bobot L2-regularizer sebesar 1e-4, yaitu 0,856. Meskipun demikian, perbedaan antara ketujuh nilai tidak terlalu tajam. Perubahan nilai bobot L2 tidak terlalu memengaruhi nilai rata-rata akurasi. Uji coba berikutnya bertujuan mengetahui pengaruh nilai penalti *structure loss* κ . Pada uji coba ini, nilai bobot L2-regularizer

ditetapkan berdasarkan hasil terbaik pada uji coba sebelumnya, yaitu sebesar $1e-4$. Sedangkan vektor kata ditetapkan sebesar 50. Hasil rata-rata akurasi pada uji coba ini ditunjukkan pada Tabel 4.

Tabel 5 menunjukkan hasil akurasi terbaik yaitu 0,856 diberikan oleh nilai penalti *structure loss* sebesar 0,8. Sama halnya dengan uji coba sebelumnya, perubahan nilai rata-rata akurasi tidak terlalu tajam terhadap perubahan nilai penalti *structure loss*. Variasi nilai penalti tidak terlalu memengaruhi hasil rata-rata akurasi parser berbasis RvNN. Uji coba ketiga bertujuan untuk mengetahui pengaruh vektor kata terhadap rata-rata akurasi *parsing*. Pada skenario ini, bobot L2 dan nilai penalti *structure loss* ditetapkan berdasarkan nilai uji coba sebelumnya, yaitu sebesar $1e-4$ dan 0.8. Hasil uji coba pada skenario ketiga ditunjukkan pada Tabel 6.

Tabel 6. Hasil akurasi dari beberapa ukuran dimensi vektor kata

Dimensi vektor kata	fold 0	fold 1	fold 2	fold 3	fold 4	Rata-Rata akurasi
50	0,881	0,896	0,857	0,842	0,807	0,856
100	0,887	0,898	0,901	0,811	0,790	0,857
200	0,878	0,890	0,869	0,839	0,845	0,864

Tabel 6 menunjukkan hasil rata-rata akurasi terbaik diberikan oleh dimensi vektor kata sebesar 200. Tabel 6 juga menunjukkan bahwa hasil rata-rata akurasi semakin meningkat ketika dimensi vektor kata semakin besar. Meskipun demikian, peningkatan rata-rata akurasi tidak terlalu tajam.

3.3. Pembahasan Hasil Uji Coba

Tabel 7. Contoh hasil *parsing* struktur semantik soal yang benar

Soal cerita	Prediksi	Aktual
Avi mendapat 50 butir telur asin. Diberikan kepada tetangganya 30 butir. Ibu membeli lagi 20 butir telur asin. Berapa butir telur asin yang ada di rumah Avi sekarang?	[0-3] <i>null</i> [0-2] <i>null</i> [0-1] <i>null</i> [0] avi mendapat 50 butir telur asin [1] diberikan kepada tetangganya 30 butir [2] ibu membeli lagi 20 butir telur asin [3] berapa butir telur asin yang ada di rumah avi sekarang	[0-3] <i>null</i> [0-2] <i>null</i> [0-1] <i>null</i> [0] avi mendapat 50 butir telur asin [1] diberikan kepada tetangganya 30 butir [2] ibu membeli lagi 20 butir telur asin [3] berapa butir telur asin yang ada di rumah avi sekarang
Doni mempunyai 16 buah kacang. Doni membagi kacang itu pada kedelapan temannya. Mereka mendapatkan kacang sama banyak. Berapa kacang yang diterima masing-masing teman Doni?	[0-3] <i>null</i> [0-2] <i>null</i> [0] doni mempunyai 16 buah kacang [1-2] <i>null</i> [1] doni membagi kacang itu pada kedelapan temannya [2] mereka mendapatkan kacang sama banyak [3] berapa kacang yang diterima masing-masing teman doni	[0-3] <i>null</i> [0-2] <i>null</i> [0] doni mempunyai 16 buah kacang [1-2] <i>null</i> [1] doni membagi kacang itu pada kedelapan temannya [2] mereka mendapatkan kacang sama banyak [3] berapa kacang yang diterima masing-masing teman doni
Eni mempunyai 50 kancing. Kancing-kancing itu dimasukkan sama banyak dalam 5 kotak. Berapakah isi masing-masing kotak?	[0-2] <i>null</i> [0-1] <i>null</i> [0] eni mempunyai 50 kancing [1] kancing-kancing itu dimasukkan sama banyak dalam 5 kotak [2] berapakah isi masing-masing kotak	[0-2] <i>null</i> [0-1] <i>null</i> [0] eni mempunyai 50 kancing [1] kancing-kancing itu dimasukkan sama banyak dalam 5 kotak [2] berapakah isi masing-masing kotak

Berdasarkan hasil uji coba pertama, perubahan bobot *L2-regularizer* tidak terlalu memengaruhi hasil rata-rata akurasi parser berbasis RvNN. Perubahan naik maupun turun dari bobot L2 tidak menghasilkan perubahan *linear* dari rata-rata akurasi. Perbedaan yang tidak terlalu besar ini kemungkinan karena penggunaan fungsi *linear* pada *layer 2* RvNN. Fungsi *linear* ini menghasilkan bilangan riil dengan batas atas bawah berupa nilai positif atau negatif tak hingga. Karena *output* dari fungsi *linear* ini menjadi masukan untuk fungsi objektif, maka fungsi objektif juga akan menghasilkan bilangan riil dengan batas atas atau bawah berupa bilangan tak hingga. Di sisi lain, proses pelatihan mengupayakan nilai parameter di setiap *layer* RvNN (parameter *U* dan *W*) agar selalu menghasilkan

nilai fungsi objektif yang selalu menurun. Proses pelatihan akan menurunkan nilai fungsi objektif tersebut hingga nilai negatif tak hingga. Pada kondisi ini, bobot L2 hanya berfungsi untuk menentukan pergeseran posisi kurva fungsi objektif. Di mana pun posisi kurva fungsi objektif ini, proses pelatihan pada akhirnya akan selalu berupaya menentukan parameter RvNN agar fungsi objektif selalu menuju negatif tak hingga.

Hasil uji coba kedua menunjukkan bahwa perubahan nilai penalti *structure loss* tidak terlalu mempengaruhi hasil rata-rata akurasi. Hal ini karena fungsinya hanya sebagai skala bagi fungsi *structure loss*. Akan tetapi, nilai penalti berpengaruh terhadap waktu yang diperlukan untuk *convergence* ketika *learning*. Hasil catatan waktu yang diperlukan untuk *convergence* ketika *learning* untuk keempat nilai penalti *structure loss*, yaitu nilai penalti 0,6 sebesar 2,1 jam, nilai penalti 0,7 sebesar 2,21 jam, nilai penalti 0,8 sebesar 2,23 jam, dan nilai penalti 0,9 sebesar 2,58 jam. Hal ini kemungkinan besar disebabkan karena peran dari fungsi *structure loss*. Nilai dari fungsi *structure loss* bertindak sebagai margin minimal untuk membedakan antara struktur pohon biner yang benar (pohon biner yang disediakan pada *dataset*) dengan struktur pohon biner yang salah. Ketika nilai penalti semakin besar maka nilai *structure loss* semakin besar dan secara otomatis margin juga semakin besar. Untuk mencapai nilai margin yang semakin besar, proses *training* lebih lama karena memerlukan proses perulangan tambahan untuk mengupdate parameter RvNN.

Hasil uji coba ketiga menunjukkan bahwa ukuran vektor kata memengaruhi hasil rata-rata akurasi parser. Semakin tinggi ukuran dimensi vektor kata, tingkat rata-rata akurasi semakin meningkat. Ukuran dimensi vektor kata ini secara otomatis menjadi ukuran dimensi vektor kalimat maupun vektor gabungan kalimat. Peningkatan rata-rata akurasi kemungkinan karena vektor berdimensi besar memiliki tingkat detail yang lebih tinggi daripada vektor berdimensi rendah pada medan vektor. Beberapa contoh hasil *parsing* struktur semantik soal yang benar ditunjukkan pada Tabel 7. Sedangkan beberapa contoh hasil *parsing* struktur semantik soal yang salah ditunjukkan pada Tabel 8.

Tabel 8. Contoh hasil *parsing* struktur semantik soal yang salah

Soal cerita	Prediksi	Aktual
Pak Umar beternak ayam ras 90 ekor. Ayam tersebut ditempatkan dalam 10 buah kandang. Masing-masing kandang ditempati ayam dengan jumlah yang sama. Berapakah ayam yang ada dalam tiap kandang?	[0-3] <i>null</i> [0] pak umar beternak ayam ras 90 ekor [1-3] <i>null</i> [1-2] <i>null</i> [1] ayam tersebut ditempatkan dalam 10 buah kandang [2] masing-masing kandang ditempati ayam dengan jumlah yang sama [3] berapakah ayam yang ada dalam tiap kandang	[0-3] <i>null</i> [0-2] <i>null</i> [0] pak umar beternak ayam ras 90 ekor [1-2] <i>null</i> [1] ayam tersebut ditempatkan dalam 10 buah kandang [2] masing-masing kandang ditempati ayam dengan jumlah yang sama [3] berapakah ayam yang ada dalam tiap kandang
Dengan motor Andi telah menempuh jarak 50 km. Tiap jam Andi menempuh jarak 5 km. Berapa jam Andi sudah mengendarai motor tersebut?	[0-2] <i>null</i> [0-1] <i>null</i> [0] dengan motor andi telah menempuh jarak 50 km [1] tiap jam andi menempuh jarak 5 km [2] berapa jam andi sudah mengendarai motor tersebut	[0-2] <i>null</i> [0] dengan motor andi telah menempuh jarak 50 km [1-2] <i>null</i> [1] tiap jam andi menempuh jarak 5 km [2] berapa jam andi sudah mengendarai motor tersebut

4. Kesimpulan

Kesimpulan yang dapat ditarik dari hasil uji coba adalah model RvNN dapat digunakan sebagai parser struktur semantik soal berbahasa Indonesia. Hasil uji coba menunjukkan tingkat akurasi terbaik sebesar 0,864 atau 86,4 % dengan nilai penalti *structure loss* sebesar 0,8; bobot *l2-regularizer* sebesar $1e-4$ dan ukuran dimensi vektor kata sebesar 200. Hasil uji coba menunjukkan bahwa dimensi vektor kata merupakan faktor yang berpengaruh terhadap hasil rata-rata akurasi parser struktur semantik soal Bahasa Indonesia berbasis RvNN. Ada beberapa hal yang dapat dipertimbangkan untuk penelitian selanjutnya. Hal pertama terkait dengan mekanisme alternatif untuk mengenumerasi kandidat struktur semantik soal sebelum proses *scoring*. Hal lain yaitu perlunya mempertimbangkan penggabungan

fungsi objektif untuk parser dan untuk *classifier* operasi hitung melalui skema *joint learning*. Penggabungan ini berpotensi menghasilkan parameter RvNN yang sesuai dengan persoalan karena dilatih menggunakan dua fungsi objektif yang berbeda secara bersamaan.

5. Referensi

- Clark, P. (2015). Elementary School Science and Math Tests as a Driver for AI: Take the Aristo Challenge! *Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence* (pp. 4019-4021). Austin, Texas, USA: AAAI Press.
- Clark, P., & Etzioni, O. (2016). My Computer Is an Honor Student – but How Intelligent Is It? Standardized Tests as a Measure of AI. *AI Magazine*, 37(1), 5-12.
- Koncel-Kedziorski, R., Hajishirzi, H., Sabharwal, A., Etzioni, O., & Ang, S. D. (2015). Parsing Algebraic Word Problems into Equations. *Transactions of the Association for Computational Linguistics*, 3, 585-597.
- Kushman, N., Artzi, Y., Zettlemoyer, L., & Barzilay, R. (2014). Learning to Automatically Solve Algebra Word Problems. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 271-281). Baltimore, Maryland, USA: Association for Computational Linguistics.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). *Efficient Estimation of Word Representations in Vector Space*. Retrieved from Arxiv: <https://arxiv.org/pdf/1301.3781.pdf>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. Lake Tahoe, Nevada, USA: Neural Information Processing Systems (NIPS).
- Mitra, A., & Baral, C. (2016). Learning To Use Formulas To Solve Simple Arithmetic Problems. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 2144-2153). Berlin, Germany: Association for Computational Linguistics.
- O'Donnell, M. (2000). RSTTool 2.4 - A Markup Tool for Rhetorical Structure Theory. *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, (pp. 253-256).
- Ratlif, N. D., Bagnell, J. A., & Zinkevich, M. A. (2006). Maximum Margin Planning. *International Conference on Machine Learning*. Pittsburgh, USA: International Machine Learning Society - ICML.
- Roy, S., Vieira, T., & Roth, D. (2015). Reasoning about Quantities in Natural Language. *Transactions of the Association for Computational Linguistics*, 3, 1-13.
- Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., & Manning, C. D. (2011). Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. *Advances in Neural Information Processing Systems 24 (NIPS 2011)* (pp. 1-9). Granada, Spain: Neural Information Processing Systems (NIPS).
- Socher, R., Lin, C. C.-Y., Ng, A. Y., & Manning, C. D. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. *International Conference on Machine Learning (ICML)*. Bellevue, Washington, USA: International Machine Learning Society (IMLS).
- Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 151-161). Edinburgh, Scotland, UK: Association for Computational Linguistics.
- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1631-1642). Seattle, Washington, USA: Association for Computational Linguistics.
- Taskar, B., Klein, D., Collins, M., Koller, D., & Manning, C. (2004). Max-Margin Parsing. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.
- Upadhyay, S., Chang, M.-W., Chang, K.-W., & Yih, W.-t. (2016). Learning from Explicit and Implicit Supervision Jointly For Algebra Word Problems. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 297-306). Austin, Texas: Association for Computational Linguistics.
- Wang, L., Zhang, D., Gao, L., Song, J., Guo, L., & Shen, H. T. (2018). MathDQN: Solving Arithmetic Word Problems via Deep Reinforcement Learning. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)* (pp. 5545-5552). New Orleans, Louisiana, USA: AAAI Press.

- Wang, Y., Liu, X., & Shi, S. (2017). Deep Neural Solver for Math Word Problems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 845-854). Copenhagen, Denmark: Association for Computational Linguistics.
- Zhou, L., Dai, S., & Chen, L. (2015). Learn to Solve Algebra Word Problems Using Quadratic Programming. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 817–822). Lisbon, Portugal: Association for Computational Linguistics.
- Zhu, M., Zhang, Y., Chen, W., Zhang, M., & Zhu, J. (2013). Fast and Accurate Shift-Reduce Constituent Parsing. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 434-443). Sofia, Bulgaria: Association for Computational Linguistics.