

Pengenalan Intent pada Natural Language Understanding Berbahasa Indonesia dengan Menggunakan Metode Convolutional Neural Network

Daniel Adi, Leo Willyanto Santoso, Alvin Nathaniel Tjondrowiguno
Program Studi Informatika, Fakultas Teknologi Industri
Universitas Kristen Petra, Surabaya
Indonesia

danieladilinhung@gmail.com , leow@petra.ac.id, alvin.nathaniel@petra.ac.id

ABSTRAK

Untuk mengikuti perkembangan teknologi dan kebiasaan masyarakat, *intelligent bot* sudah menjadi bagian dari dunia bisnis yang membantu pelaku bisnis menjaga relasi dengan *customer*. Namun sayangnya *resource* untuk pembuatan *intelligent bot* berbahasa Indonesia masih sangat sedikit apabila dibandingkan dengan *High Resource Language* seperti English. Untuk itu perlu diadakan penelitian mengenai *Natural Language Understanding* bahasa Indonesia, yang merupakan salah satu bagian penting dari *intelligent bot*.

Metode yang digunakan adalah *Convolutional Neural Network* untuk melakukan *training* pada model. Model terdiri dari *embedding layer*, *convolutional layer*, *max pooling*, *flatten*, *dropout*, dan *softmax layer*. Dalam pembuatan model ada banyak variabel yang bisa diuji coba seperti *dropout*, jumlah *filter*, ukuran *filter*, dan sebagainya.

Penelitian ini menunjukkan bahwa jumlah dan kualitas data untuk setiap kategori bisa mempengaruhi bagaimana suatu model memahami fitur dari kategori tersebut sehingga berdampak pada *precision*. Kualitas dari *word2vec*, salah satu *resource* penting yang digunakan di dalam model bisa memberikan dampak yang signifikan terhadap *precision*. Besar kecilnya *dropout* mempengaruhi bagaimana model menangkap fitur penting dari data. Dari berbagai pengujian, *precision* terbaik yang dihasilkan adalah 93 %.

Kata Kunci: *Artificial neural network, Keras, convolutional neural network, Intelligent Bot.*

ABSTRACT

To keep up with technological developments and people behavior, intelligent bot has become part of the business world which help them maintain good relation with their customer. Unfortunately, resource for intelligent bot in Indonesian language is very scarce compared to High Resource Language like English. Therefore further research about Natural Language Understanding in Indonesian language is needed.

We use Convolutional Neural Network method to train our model. Model consist of embedding layer, convolutional layer, max pooling, flatten, dropout, and softmax layer. In the process of making model, there are many variable that can be tested such as dropout, number of filter, size of filter, etc.

This research show that the amount and quality of data for each category can affect how a model understand the feature of each

category which affect the overall precision. The quality of word2vec, one of the most important resource in the model can give significant impact on precision. The size of dropout can affect how the model understand the important feature of data. From various tests, we found that the best precision is 93 %.

Keywords: *Artificial neural network, Keras, convolutional neural network, Intelligent Bot.*

1. LATAR BELAKANG

Seiring dengan perkembangan zaman, dunia teknologi khususnya *intelligent bot* semakin memberikan dampak yang signifikan terhadap dunia bisnis. Berdasarkan *survey* yang dilakukan oracle, “*With regards to chatbots, which are in many ways the most recognisable form of AI, 80% of sales and marketing leaders say they already use these in their CX or plan to do so by 2020*” [6]. Hal ini membuktikan bahwa *intelligent bot* menjadi hal yang penting dan memberikan dampak signifikan bagi para pelaku bisnis. Salah satu teknologi penting yang tidak bisa lepas dari *intelligent bot* adalah *Natural Language Processing*.

Natural Language Understanding (NLU) merupakan bagian dari *Natural Language Processing* (NLP) yang berhubungan dengan bagaimana suatu program bisa memahami pesan (*intent*) yang ingin disampaikan manusia. Pada dasarnya, kesulitan yang dihadapi pada pengembangan *intelligent bot* Indonesia saat ini adalah minimnya kemampuan *intelligent bot* untuk memahami pesan (*intent*) disampaikan manusia [1]. Beberapa *Intelligent bot* di Indonesia masih menggunakan sistem SPOK sehingga kebiasaan pengguna yang cenderung tidak menggunakan SPOK membuat pesan sulit dipahami [3].

Sebagian besar pengembangan NLU saat ini hanya ditujukan ke *High Resource Language* (HRL) seperti English. Menurut Julia Hirschberg dan Christopher D. Manning : “*A major limitation of NLP today is the fact that most NLP resources and systems are available only for high-resource languages (HRLs), such as English, French, Spanish, German, and Chinese. In contrast, many low-resource languages (LRLs) — such as Bengali, Indonesian, Punjabi, Cebuano, and Swahili—spoken and written by millions of people have no such resources or systems available*” [4]. Hal ini menyebabkan pengembangan NLU berbahasa Indonesia tidak sebaik dengan NLU HRL.

Metode *state of the art* yang digunakan untuk mengenali *intent* user pada bahasa English adalah CNN (*Convolutional Neural Network*) [2]. CNN tidak bergantung pada susunan kata seperti SPOK melainkan melalui semantik kata – kata yang sudah di

vektorisasi. Metode ini sudah terbukti bisa memberikan *akurasi* pengenalan *intent* yang baik (92%) [5]. Namun sayangnya metode ini masih belum dianalisa untuk NLU dari bahasa Indonesia.

2. TINJAUAN STUDI

Pada percobaan implementasi NLU pada Arabic Dialogue System, CNN sebagai *State-of-the-Art* digunakan untuk mengenali *intent* dari *user* pada *home automation*. Hasilnya CNN bisa memberikan akurasi yang hampir menyerupai CNN pada bahasa English (92%) [2].

Convolutional Neural Network for Sentence Classification merupakan journal karya Yoon Kim yang membahas tentang fitur CNN untuk klasifikasi. Salah satu klasifikasi yang dilakukan adalah klasifikasi suatu kalimat ke dalam suatu kategori dari kumpulan kategori yang sudah ditentukan. Proses klasifikasi ini menghasilkan akurasi 92% [5].

Percobaan lain yang dilakukan adalah penggunaan CNN untuk klasifikasi yang dilakukan oleh Ye Zhang dan Bryon C. Wallace. Dengan sistem klasifikasi pertanyaan yang sama, mereka telah menghasilkan akurasi sebesar 91.54 % . [7]

3. DESAIN SISTEM

Proses meliputi *Preprocessing* , Membuat *Embedding matrix* dari Word2Vec, pembuatan *Model Neural Network* dengan CNN, sistem Untuk *Training* dan *Testing*, serta sistem untuk *Chat* berdasarkan model yang sudah dibuat.

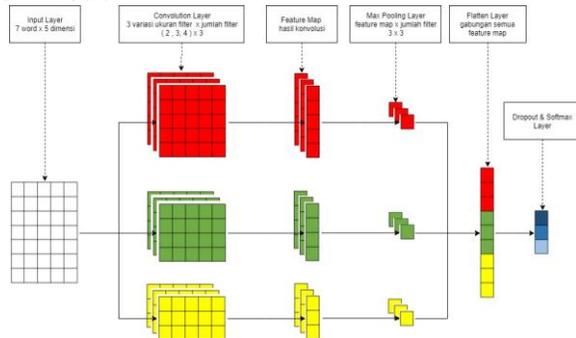
3.1 Dataset dan Preprocessing

Dataset merupakan hasil survei dari calon mahasiswa berupa 336 pertanyaan tentang pertanyaan yang biasanya diajukan calon mahasiswa. *Preprocessing* dilakukan dengan menghapus tanda baca serta mengubah semua kata menjadi *lowercase*. Hasil data dari *preprocessing* dibagi menjadi data *traing* dan data *test* dengan perbandingan 4:1.

3.2 Embedding Matrix

Pada proses ini juga terjadi tokenizing, sequencing dan padding terhadap datat *train*. *Embedding matrix* berisi kumpulan kata dimana setiap kata memiliki kumpulan vektor yang merepresentasikan makna semantik dari suatu kata. *Embedding matrix* yang akan digunakan untuk model berisi kumpulan kata dari data *train* beserta nilai vektor untuk setiap kata tersebut yang diambil dari word2vec. Kata yang tidak ditemukan di word2vec akan diisi dengan nilai vektor 0.

3.3 Model



Gambar 1. Model CNN

Model CNN dari gambar 1 terdiri dari:

- *Input layer* untuk menampung kata – kata dari kalimat data *train*. Di dalam *input layer* terjadi proses *embedding* dimana model menggunakan nilai – nilai vektor hasil proses *embedding matrix* untuk mengisi nilai vektor dari kata – kata yang sudah ditampilkan di *layer input*.

- *Convolutional layer* menggunakan 3 variasi ukuran *filter* (2,3,4) dimana setiap variasi ukuran *filter* terdiri dari sekumpulan *filter* dengan ukuran yang sama. Pada *layer* ini terjadi proses konvolusi dengan menggunakan *activation function* RELU antara *input layer* dan *filter*. Hasil dari proses ini adalah feature map.

- *Max pooling layer* berfungsi untuk mengurangi jumlah fitur dari feature map karena jumlah fitur pada feature map terlalu banyak untuk diproses.

- *Flatten layer* menggabungkan semua hasil *max pooling* yang hasilnya terpisah – pisah menjadi 1 matrix 1 dimensi.

- *Dropout & softmax layer* berfungsi untuk mengurangi *overfitting* pada model dan melakukan proses klasifikasi berdasarkan fitur yang sudah didapat.

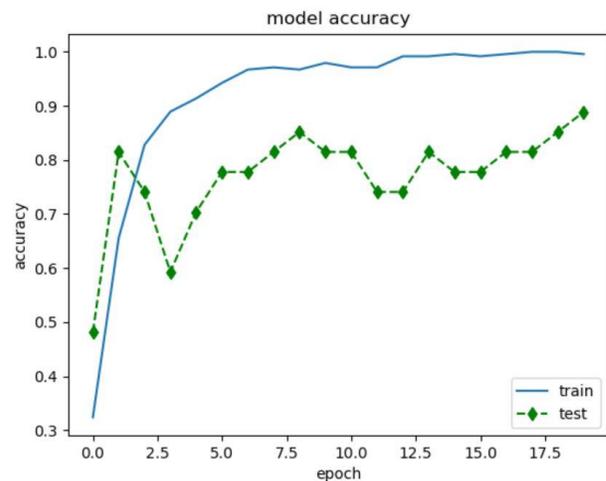
3.4 Training dan Testing

Training dilakukan dengan menggunakan ADAM *optimizer* untuk melakukan update nilai weight dari *filter* sehingga program bisa menghasilkan model dengan performa klasifikasi yang paling baik. Model hasil *training* diuji dengan data *test* yang sudah kita dapat dari proses *preprocessing*. Performa model diukur menggunakan *precision* yang merupakan perbandingan antara model yang diprediksi dengan benar dan jumlah total dadta

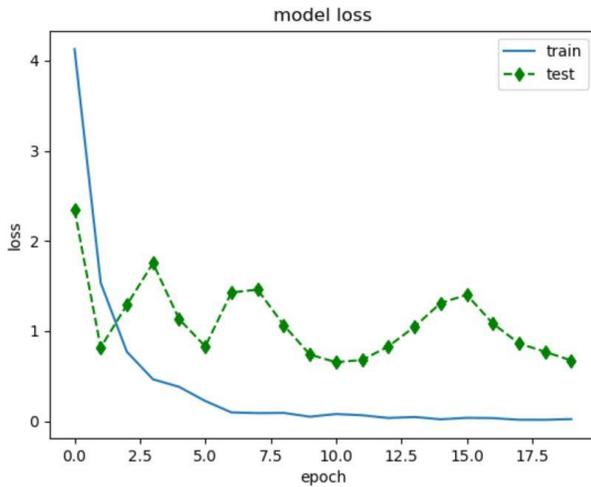
4. PENGUJIAN SISTEM

Proses pengujian dilakukan secara kronologis dimana setiap pengujian akan menggunakan konfigurasi terbaik dari pengujian sebelumnya. Untuk konfigurasi awal proses *training* dilakukan dengan konfigurasi 512 jumlah *filter*, 3 jenis *filter* berukuran 2, 3, dan 4, serta persentase *drop* 50% dan *epoch* yang berjumlah 20 *epoch*. Semua percobaan untuk setiap konfigurasi akan dilakukan sebanyak 10 kali.

4.1 Epoch



Gambar 2. Akurasi Epoch



Gambar 3. Loss Epoch

Pembuatan *neural network model* diawali dengan *weight* yang nilainya *random* sehingga menghasilkan tingkat akurasi yang berbeda-beda pula. Menurut gambar 2 dan 3, *epoch* yang digunakan adalah 20 melihat *loss* dan akurasi pada *epoch* 20 sudah cukup baik.

4.2 Pengujian Word2vec

Tabel 1. Pengujian Word2vec

Status word2vec	Precision tertinggi	Precision Terendah	Rata – Rata Precision
Updated	95.4%	89.3 %	91 %
Not Updated	93.9 %	83.3 %	88 %

- *Not Updated* : Setelah penelitian lebih lanjut diketahui bahwa 66 dari 438 jenis kata yang ada di data tidak ditemukan didalam word2vec. Kata – kata yang tidak ditemukan di word2vec akan diisi dengan nilai vektor 0 sehingga memungkinkan adanya penurunan *precision*.

- *Updated* : 46 dari 66 kata yang tidak ditemukan diganti dengan kata dasar atau kata lain yang memiliki makna semantik yang sama dan ada di word2vec.

Berdasarkan tabel 1, hasil menunjukan bahwa tidak ditemukannya kata – kata di dalam word2vec bisa memberikan dampak penurunan *precision*. Hal ini menunjukkan bahwa kualitas word2vec sangat mempengaruhi *precision* model terutama mengingat kata yang tidak ditemukan di word2vec akan diisi dengan 0.

4.3 Pengujian Jumlah Filter

Tabel 2. Pengujian Jumlah Filter

Status word2vec	Jumlah filter	Precision tertinggi	Precision Terendah	Rata – Rata Precision
Updated	512	92.4%	86.3 %	90 %
Updated	256	95.4 %	87.8 %	91 %
Updated	128	93.9%	86.3 %	90 %
Updated	64	93.9%	87.8 %	90 %
Updated	36	92.4 %	87.3 %	90 %

Berdasarkan tabel 2, jumlah *filter* tidak memberikan dampak yang cukup signifikan pada *precision* model. Untuk saat ini dapat kita

simpulkan bahwa *filter* 256 memiliki hasil *precision* yang paling baik.

4.4 Pengujian Dropout

Tabel 3. Pengujian Dropout

Jumlah filter	Dropout	Precision tertinggi	Precision Terendah	Rata – Rata Precision
256	0.1	93.9 %	90.9 %	93 %
256	0.2	93.9 %	84.8 %	91 %
256	0.3	93.9 %	87.8 %	91 %
256	0.4	93.9 %	90.9 %	92 %
256	0.5	95.4 %	87.8 %	91 %

Hasil pengujian berbagai *drop* pada tabel 3 diatas menyimpulkan bahwa percobaan dengan nilai *drop* yang lebih kecil menghasilkan *precision* yang lebih baik. Hal ini disebabkan karena jumlah *filter* sebesar 256 x 3 (jumlah jenis *filter*) tidak mampu menangkap karakteristik dari kategori (terlalu banyak yang di *drop*) apabila persentase *drop* 50 %.

4.5 Variasi Ukuran Filter

Tabel 4. Pengujian Jumlah Jenis Filter

Dropout	Ukuran filter	Precision tertinggi	Precision Terendah	Rata – Rata Precision
0.1	2,3,4	93.9 %	90.9 %	93 %
0.1	3,4,5	93.9 %	86.3 %	91 %
0.1	2,3,4,5	95.4 %	89.3 %	90 %
0.1	2,3	95.4 %	86.3 %	90 %
0.1	2	95.4 %	89.3 %	91 %

Hasil pengujian pada tabel 4 menunjukkan bahwa model dengan ukuran *filter* 2,3,4 menghasilkan *precision* terbaik sebesar 93 %.

4.6 Perbandingan Struktur Model filter

Tabel 5. Pengujian Perbandingan Struktur

Ukuran filter	Model	Precision tertinggi	Precision Terendah	Rata – Rata Precision
2, 3, 4	1 filter	93.9 %	90.9 %	93 %
2,2,3,3,4,4	1 pasang filter	92.4 %	74.2 %	89 %

Model yang telah dibuat memiliki struktur model dengan menggunakan 3 *filter* dengan 3 ukuran yang berbeda sedangkan struktur model yang direferensi dari Zhang & Wallace [7], menggunakan 3 pasang *filter* (total 6) dimana setiap pasang memiliki variasi ukuran *filter* yang berbeda dengan pasangan yang lain. Hasil pengujian pada tabel 5 menunjukan bahwa model yang menggunakan 1 *filter* untuk setiap ukuran memiliki peforma yang lebih baik daripada model dengan 1 pasang *filter* untuk setiap ukurannya.

4.7 Pengujian Kategori

Percobaan dilakukan dengan mencoba membagi 336 pertanyaan hasil survei menjadi 5 kategori yang lebih spesifik : Pelajaran (72 pertanyaan), Karir (73 pertanyaan), Other (126 pertanyaan), kesulitan (31 pertanyaan), dan *requirement* (34 pertanyaan). 5 kategori ini dibuat dengan catatan bahwa setiap kategori telah mencakup minimal kurang lebih 10% dari total data. Berikut adalah hasil *precision* untuk setiap kategori dengan perhitungan

jumlah pertanyaan yang diprediksi dengan benar / jumlah hasil prediksi untuk kategori tersebut.

Tabel 6. Precision per Kategori (5 kategori)

Kategori	Precision Kategori
Pelajaran	0.93
Karir	1.0
Other	0.85
Kesulitan	0.73
Requirement	0.5

Pada tabel 6, *precision* Kategori kesulitan dan *requirement* memiliki hasil yang cukup rendah apabila dibandingkan dengan kategori lain. Salah satu faktor penyebab *precision* yang rendah adalah sedikitnya jumlah data pada kategori tersebut apabila dibandingkan dengan kategori – kategori lainnya. Kategori kesulitan dan *requirement* memiliki jumlah persentase data kurang lebih 10% saja dari total data. Pengujian selanjutnya yang dilakukan adalah menggabungkan kategori *requirement* ke dalam other. Percobaan kedua adalah menggabungkan kategori *requirement* dan kesulitan ke other.

Tabel 7. Precision per Kategori (4 Kategori)

Kategori	Precision Kategori
Pelajaran	0.88
Karir	1.0
Other	0.90
Kesulitan	0.86

Tabel 8. Precision per Kategori (3 Kategori)

Kategori	Precision Kategori
Pelajaran	0.92
Karir	0.95
Other	0.93

Setelah melihat hasil pengujian pada tabel 7 dan 8, Hasil proses dengan 4 kategori memiliki hasil yang jauh lebih baik apabila dibandingkan dengan 5. Hal ini menunjukkan data dari kategori *requirement* masih belum bisa menangkap karakteristik dari kategori tersebut karena jumlah data yang kurang dan variasi kata yang relatif lebih bervariasi dari kategori – kategori lainnya sehingga mengakibatkan penurunan *precision* yang cukup banyak. Data dari kategori kesulitan memang berjumlah sedikit namun variasi kata dari kategori ini cenderung lebih sedikit sehingga penurunan *precision* tidak sebesar yang diakibatkan kategori *requirement*.

5. KESIMPULAN DAN SARAN

Kesimpulan dari pengujian ini adalah:

- *CNN* memberikan prospek yang bagus sebagai metode ketika digunakan untuk klasifikasi kalimat berbahasa Indonesia.

Prediksi yang dihasilkan memiliki *precision* yang mendekati *precision* bahasa English yaitu 93%.

- Konfigurasi terbaik untuk skripsi *CNN* pada *epoch* 20 ini adalah *filter* ukuran 2, 3, 4 yang masing berjumlah 256 diikuti dengan *dropout* sebesar 10 %, data dengan kualitas (variasi kata) terbaik dan *word2vec* sudah diupdate untuk mengenali kata sebaik mungkin.
- Kualitas *word2vec* terkait kemampuan *word2vec* mengenali kata dan jumlah serta kualitas (variasi kata) data *input* yang merepresentasikan suatu kategori sangat berperan penting dalam bagaimana suatu model mengenali *intent*.

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut untuk program ini adalah sebagai berikut:

- Mengaplikasikan program ini ke data dengan jumlah yang lebih banyak serta kualitas yang lebih baik serta kategori yang lebih beragam
- Penelitian mendalam tentang *word2vec* untuk menghasilkan *word2vec* dengan kualitas yang sebaik mungkin
- Mengintegrasikan metode pengenalan *intent* ini dengan *entity recognition* agar *intelligent bot* bisa menjawab pertanyaan yang lebih rumit.

6. REFERENCES

- [1] Barkin, J. 2016. *When Bots Fail At Conversation*. URL= <https://medium.com/@joshbarkin/when-bots-fail-at-conversation-d7419605f5cc>
- [2] Bashira, A. M., Hassan, A., Rosman, B., Duma, D., & Ahmed, M. (2018). Implementation of A Neural Natural Language Understanding. *ACLing 2018*, (p. 228). Dubai.
- [3] Eka, R. 2017. *Pengembangan Layanan Berbasis Chatbot di Indonesia Sebagai “The Next Big Thing”*. URL= <https://dailysocial.id/post/pengembangan-layanan-berbasis-chatbot-di-indonesia-sebagai-the-next-big-thing>
- [4] Hirschberg, J., & Manning, C. D. 2015. *Advances in natural language processing*. URL= <https://cs224d.stanford.edu/papers/advances.pdf>
- [5] Kim, Y. 2014. *Convolutional Neural Networks for Sentence Classification*. 4. arXiv:1408.5882
- [6] Oracle. 2016. *Can Virtual Experience Replace Reality ?* URL= https://www.oracle.com/webfolder/s/delivery_production/docs/FY16h1/doc35/CXResearchVirtualExperiences.pdf
- [7] Zhang, Y., & Wallace, B. C. 2016. A Sensitivity Analysis of (and Practitioners’ Guide to) *Convolutional Neural Network for Sentence Classification*. 6. arXiv:1510.03820