ISSN 2085-4552

Pencarian *Question-Answer* Menggunakan *Convolutional Neural Network* Pada Topik Agama Berbahasa Indonesia

Rizqa Raaiqa Bintana[1], Chastine Fatichah[2], Diana Purwitasari[3]
Departemen Informatika, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
rizqa.raaiqa.bintana@gmail.com
chastine@cs.its.ac.id
diana.purwitasari@gmail.com

Diterima 6 Mei 2018
Disetujui 8 Juni 2018

***Abstract*—Community-based question answering (CQA) is formed to help people who search information that they need through a community. One condition that may occurs in CQA is when people cannot obtain the information that they need, thus they will post a new question. This condition can cause CQA archive increased because of duplicated questions. Therefore, it becomes important problems to find semantically similar questions from CQA archive towards a new question. In this study, we use convolutional neural network methods for semantic modeling of sentence to obtain words that they represent the content of documents and new question. The result for the process of finding the same question semantically to a new question (query) from the question-answer documents archive using the convolutional neural network method, obtained the mean average precision value is 0,422. Whereas by using vector space model, as a comparison, obtained mean average precision value is 0,282.**

***Index Terms*—community-based question answering, convolutional neural network, question retrieval**

I. PENDAHULUAN

Komunitas tanya-jawab dibentuk untuk mempermudah seseorang dalam memperoleh informasi yang dibutuhkannya melalui suatu komunitas, contohnya Yahoo! Answers. Salah satu kondisi yang bisa terjadi dalam komunitas tanya-jawab adalah ketika pencari informasi tidak mampu menemukan informasi yang dibutuhkan, sehingga mereka akan menginputkan pertanyaan baru ke dalam sistem. Hal ini dapat mengakibatkan jumlah arsip dokumen pertanyaan dan jawaban menjadi ganda. Sehingga menjadi permasalahan yang penting untuk bisa menemukan pertanyaan yang sama secara semantik antara pertanyaan baru terhadap pertanyaan yang ada di arsip.

Classical retrieval models, seperti TF-IDF dan Okapi BM25, menggunakan representasi *bag-of-words* dan tidak mampu secara efektif menangkap informasi kontekstual dari sebuah kata. Model ini bekerja dengan mempertimbangkan informasi kemunculan kata dalam sebuah dokumen. Sebagian besar tugas temu kembali (*retrieval*) menggunakan metode-metode yang berdasarkan semantik dengan pencocokan leksikal untuk pengambilan informasi. Hal ini sebagian disebabkan bahwa konteks yang sama sering dinyatakan dengan menggunakan kosa kata dan gaya bahasa yang berbeda dalam dokumen dan *query*. Beberapa penelitian menggunakan pengetahuan dari *WordNet* untuk menemukan kata yang sama secara semantik dan membantu pengukuran nilai kemiripan semantik diantara dua kata [1]. *WordNet* merupakan *database* leksikal yang menyimpan sinonim suatu kata, dan digunakan secara luas dalam analisa teks, namun terbatas hanya untuk kata dalam bahasa inggris.

Berbagai metode diusulkan untuk pembelajaran representasi kata terdistribusi (*word embeddings*) dalam ruang vektor berdimensi rendah. Representasi kata terdistribusi membantu algoritma pembelajaran (*learning algorithm*) untuk mencapai kinerja yang lebih baik dengan cara mengelompokkan kata-kata yang mirip, dan telah diterapkan secara luas pada bidang bahasan pemrosesan bahasa alami (*natural language processing*) [2] [3]. Selain menggunakan representasi kata terdistribusi, beberapa metode lain untuk memodelkan kalimat (*neural sentence models*, model yang dikombinasikan terhadap struktur *neural network*), seperti *Neural Bag-of-Words* (NBOW), *recurrent neural network* (RNN), *recursive neural network* (RecNN), dan *convolutional neural network* (CNN) [4].

NBOW merupakan metode yang sederhana dan intuitif, namun mempunyai kekurangan dimana susunan kata menjadi hilang. Pemodelan kalimat berdasarkan RNN sensitif terhadap susunan kata, tetapi memiliki bias terhadap kata-kata terbaru yang dibutuhkan sebagai inputan. Hal ini memberikan RNN kinerja yang sangat baik dalam memodelkan bahasa, tetapi kurang optimal untuk memodelkan keseluruhan kalimat. RecNN mengadopsi struktur yang lebih umum untuk mengkodekan kalimat. Di setiap *node* dalam *tree*, konteks pada anak *node* kiri dan kanan

digabungkan oleh *classical layer*. Bobot dari lapisan dibagi di semua *node* dalam *tree*. Lapisan yang dihitung pada *node* atas memberikan sebuah representasi untuk kalimat. Namun, RecNN bergantung pada *external constituency parse tree* yang disediakan oleh *external parse tree*. CNN mempunyai kelebihan yaitu, dapat mempertahankan informasi susunan kata yang sangat penting untuk kalimat pendek, serta aktivasi nonlinier dalam CNN dapat mempelajari karakteristik yang lebih abstrak [4].

Pemodelan kalimat adalah cara untuk menganalisa dan merepresentasikan isi semantik yang ada dalam sebuah kalimat, yang melibatkan pemahaman bahasa alami. *Neural sentence models* digunakan untuk menghasilkan kata demi kata dari suatu kalimat [5] [6]. *Neural network* digunakan untuk mengekstrak struktur semantik yang tersimpan dalam sebuah kalimat ataupun dokumen. *Neural sentence models* bekerja dengan cara memetakan kata melalui inputan *query* dan representasi kata terdistribusi dari koleksi dokumen. Dari pemetaan kata tersebut akan diperoleh ekstrak kata yang dianggap sama secara semantik terhadap *query* melalui lapisan (*layer*) proyeksi.

Dalam penelitian ini, diterapkan model CNN untuk pemodelan semantik kalimat pertanyaan. Penelitian ini bertujuan untuk mengetahui performansi metode CNN dalam menemukan pertanyaan dari arsip komunitas tanya-jawab yang sesuai dan sama secara semantik dengan pertanyaan baru (*query*) yang diinputkan oleh penanya. Pada bagian berikutnya dalam tulisan ini berisi tinjauan literatur yang terkait. Dilanjutkan dengan bagian penjelasan metodologi penelitian, penjelasan hasil dan pembahasan. Dan bagian terakhir diberikan kesimpulan.

II. LITERATUR TERKAIT

A. *Questions Retrieval*

Dalam komunitas tanya-jawab, berbagai cara telah dipelajari untuk menyelesaikan permasalahan *lexical gap* dalam temu kembali pertanyaan (*questions retrieval*). Meskipun sebagian besar model *retrieval* yang sederhana mengasumsikan bahwa kemunculan kata benar-benar independen, namun informasi kontekstual sangat penting untuk mendeteksi maksud pencarian tertentu dari sebuah *query*. Pendekatan berbasis model terjemahan (*translation model-based*) diusulkan [7] yang mencoba untuk mengekstrak hubungan frase-ke-frase berdasarkan *click through data*. Hubungan tersebut diharapkan menjadi lebih efektif dalam menjembatani kesenjangan antara *query* dan dokumen. Pendekatan lainnya yang diterapkan dalam temu kembali pertanyaan adalah dengan pencocokan leksikal untuk pengambilan informasi. Pendekatan ini berdasarkan sifat semantik antar kata dalam *query* dan koleksi dokumen, sehingga konteks yang sama namun dinyatakan dengan kosa kata yang berbeda juga akan dapat di-*retrieve*.

B. *Word Embeddings*

Word embeddings (representasi kata terdistribusi) merupakan cara yang merepresentasikan kata-kata bahasa alami dengan cara mempertahankan kemiripan semantik dan sintaksis di antara kata-kata tersebut. Hal ini didapat melalui representasi kata-kata sebagai vektor berdimensi tinggi, yaitu hubungan spasial di antara *embeddings* merepresentasikan hubungan di antara kata-kata. Sebagai contoh, representasi dari kata "fisika" dan "kimia" akan dekat secara bersama, dan kata "mobil" akan dekat dengan kata "balap" dan "supir". Ada dua teknik yang dikembangkan untuk memperoleh *word embeddings* yaitu, word2vec dan GloVe. Teknik tersebut dilakukan dengan mengolah bentuk bebas teks sehingga menghasilkan vektor berkualitas tinggi yang merepresentasikan kata-kata. Teknik yang digunakan dalam penelitian ini adalah teknik word2vec.

Word2vec, diperkenalkan oleh Mikolov dkk (2013), menggunakan teknik yang disebut "*skip-gram with negative sampling*". Teknik ini tidak memprediksi kata berdasarkan pada konteks, tapi mencoba untuk memaksimalkan klasifikasi sebuah kata berdasarkan kata lain dalam kalimat yang sama. Lebih tepatnya, kita menggunakan setiap kata (*current word*) sebagai inputan untuk *log-linear classifier* dengan lapisan proyeksi yang kontinyu, dan memprediksi kata-kata dalam jarak tertentu sebelum dan setelah kata inputan tersebut (*current word*). Dari penelitian Mikolov dkk (2013) ditemukan bahwa peningkatan jarak memperbaiki kualitas vektor kata yang dihasilkan, tetapi juga meningkatkan kompleksitas komputasi. Karena kata-kata yang lebih jauh jaraknya biasanya kurang terkait dengan kata inputan daripada yang berjarak dekat dengan kata inputan. Berikut gambaran umum tentang cara kerja teknik word2vec:

a. Mengambil kata di dalam koleksi dokumen (*corpus*) latih, dan sejumlah kata-kata yang terletak dekat dengan konteks.

b. Merepresentasikan setiap kata-kata tersebut melalui sebuah vektor (sejumlah daftar kata).

Karena teknik word2vec dan GloVe menangkap hubungan semantik dan sintaksis, kedua teknik ini bisa digunakan untuk pencarian (sinonim, *query expansion*) serta rekomendasi. *Word embeddings* terlihat tidak memberikan diskriminatif antara konsep terkait namun konsep yang berbeda.

C. *Convolutional Neural Network*

Ada beberapa metode untuk memodelkan kalimat yang disebut *neural sentence model*. Peranan penting dari *neural sentence model* adalah untuk merepresentasikan *variable-length sentence* sebagai *fixed-length vector*. *Convolutional neural network* (CNN) merupakan salah satu *neural sentence model* yang digunakan untuk memodelkan kalimat [8]. Hal pertama yang dilakukan dalam tahap model CNN

adalah mentransformasi semua kata tunggal (*token*) dalam kalimat pertanyaan menjadi vektor melalui *lookup layer* dan menggunakan *word embedding* dalam kalimat secara berurutan. CNN merangkum makna sebuah kalimat melalui *convolutional layer* dan *pooling layer*, hingga mencapai sebuah representasi *fixed-length vector* pada lapisan (*layer*) akhir.

CNN mempunyai kelebihan, yaitu dapat mempertahankan informasi susunan kata dimana hal tersebut menjadi sangat penting untuk kalimat pendek. *Convolutional layer* menerapkan matriks filter satu dimensi yang melewati tiap baris fitur dalam matriks kalimat. Pembelitan (*convolving*) filter yang sama dengan *n*-gram di setiap posisi dalam kalimat memungkinkan fitur-fitur untuk diekstrak secara bebas dari posisi mereka dalam kalimat. *Convolutional layer* diikuti oleh *dynamic pooling layer* dan non-linearitas dari pemetaan fitur [8].

Arsitektur *convolutional* untuk kalimat pemodelan, seperti digambarkan pada Gambar 1, dibutuhkan sebagai inputannya berupa *word embedding* (yang dilatih terlebih dahulu dengan metode *unsupervised*) dalam kalimat yang selaras secara berurutan, dan meringkas makna kalimat melalui lapisan *convolutional* dan *pooling*, sampai mencapai representasi kata dalam *fixed length vector* pada lapisan terakhir.

Embeddings untuk seluruh kata dalam kalimat *s* membangun matriks inputan $\mathbf{s} \in R^{n_w X l_s}$, dimana l_s menyatakan panjang *s*. Sebuah *convolutional layer* diperoleh melalui *convolving* sebuah matriks dari bobot (filter) $\mathbf{m} \in R^{nxm}$ dengan matriks aktivasi pada layer berikutnya, dimana *m* adalah lebar filter. Lapisan (*layer*) pertama diperoleh dengan menggunakan *convolutional filter* untuk matriks kalimat **s** dalam layer inputan. Dimensi n_w dan lebar filter *m* adalah *hyper-parameters* pada *network*.

Jaringan menangani rentetan inputan dari berbagai variasi panjang kata. Lapisan dalam jaringan interleave *convolutional layers* dan *dynamic k-maxpooling layers* satu dimensi. Dynamic *k*-max pooling adalah generalisasi dari operator *max pooling*. Operator *max pooling* merupakan fungsi subsampling non-linear yang mengembalikan nilai-nilai maksimum [8].

III. METODE PENELITIAN

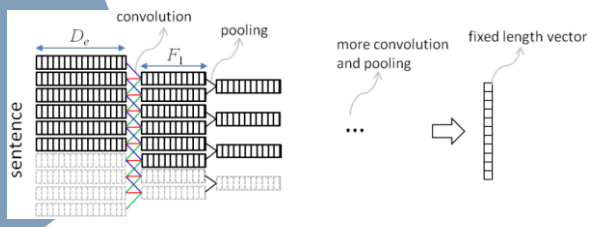
Untuk mencapai tujuan penelitian, maka detail rancangan keseluruhan proses yang dilakukan dalam penelitian ini seperti yang terlihat pada Gambar 2. Berdasarkan rancangan tersebut akan dibangun sistem yang mencakup keseluruhan proses untuk membantu proses *training* dan *testing* dalam penelitian ini.

A. *Dataset*

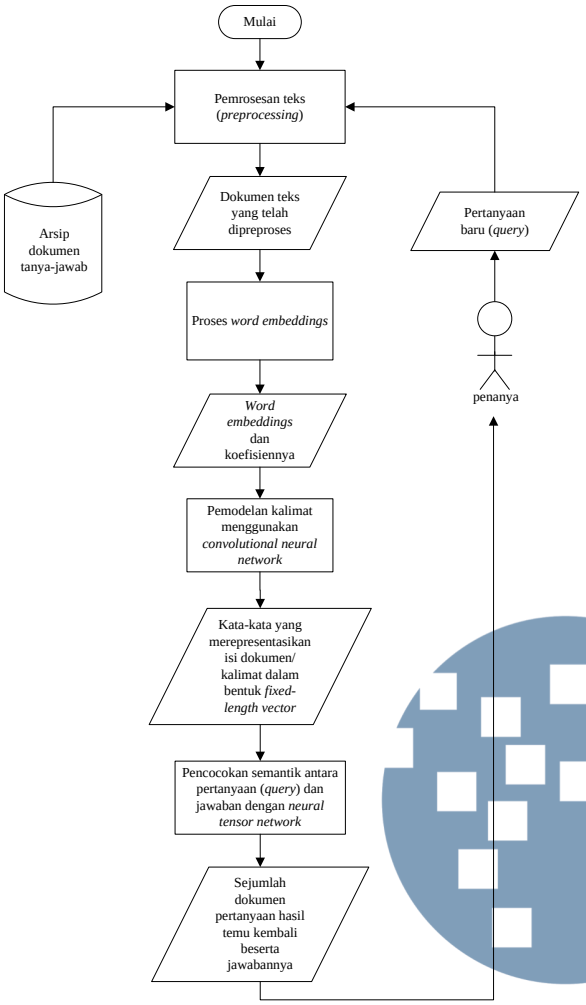
Tahapan penelitian dimulai dari pengumpulan data penelitian. Dalam penelitian ini, *dataset* yang digunakan adalah dokumen pertanyaan-jawaban yang diambil dari komunitas tanya-jawab online www.piss-ktb.com. Pertanyaan baru yang diinputkan penanya, yang selanjutnya akan disebut *query*, akan dibandingkan dengan pertanyaan yang ada di arsip (koleksi dokumen) komunitas tanya-jawab dengan tujuan untuk menemukan pertanyaan dari arsip komunitas tanya-jawab yang sama secara semantik dengan *query*.

B. Pemrosesan Teks

Seperti yang dapat dilihat pada Gambar 2, beberapa proses yang akan dilakukan dalam penelitian ini setelah pengumpulan data yaitu, melakukan preproses terhadap arsip dokumen tanya-jawab dan *query*, melakukan proses *word embeddings* terhadap keseluruhan kata dalam dokumen tanya-jawab yang telah dipreproses, proses pemodelan kalimat untuk dokumen pertanyaan dan jawaban di sisi *training* dan *query* di sisi *testing* yang dimodelkan menggunakan *convolutional neural network*, dan mengukur kecocokan antara pertanyaan dan jawaban yang ada di arsip dokumen untuk proses *training* serta antara *query* dan dokumen jawaban pada proses *testing* menggunakan model *neural tensor network*.



Gambar 1. Arsitektur *convolutional* secara keseluruhan dalam memodelkan kalimat [9]



Gambar 2. Rancangan keseluruhan proses dalam penelitian

Pemrosesan teks dalam penelitian ini dilakukan sebagai tahapan preproses terhadap arsip dokumen tanya-jawab dan *query*. Tujuan dilakukannya pemrosesan teks adalah untuk membersihkan data sebagai langkah awal untuk analisis data. Proses yang dilakukan dalam tahapan ini, yaitu:

1. Tahap penghapusan *tag markup* dan format khusus dari dalam dokumen pertanyaan-jawaban. Sebelum dilakukan tokenisasi, semua *tag markup* dan format khusus akan dihapus dari dalam dokumen. Karena koleksi dokumen pertanyaan-jawaban yang digunakan dalam penelitian ini adalah *file* dengan ekstensi .html, maka seluruh *tag* maupun *javascript* serta *style* akan dihapus.
2. Tahap tokenisasi, merupakan proses pemisahan rangkaian kata. Dalam tahap ini, seluruh kata di dalam dokumen atau kalimat dipisahkan menjadi potongan kata tunggal (*term*). Dalam proses ini juga dilakukan penghapusan karakter-karakter tertentu, yaitu tanda baca serta mengubah semua kata ke bentuk huruf kecil (*lowercase*).
3. Tahap penghapusan *stop-words* (*linguistic preprocessing*). Setelah tahap tokenisasi dilakukan, maka dilanjutkan dengan tahap penghapusan *stop-words* dari dalam dokumen. Dalam tahap ini, ada dua operasi utama yang dilakukan yaitu penghapusan *stop-words* (*stop-words removal*) dan *stemming* (pemotongan imbuhan). *Stop-words* adalah kata yang sering muncul dalam dokumen, namun kata tersebut tidak dapat mendeskripsikan topik atau sub-topik dari dokumen tersebut, sehingga tidak dapat membedakan dokumen satu dengan dokumen lainnya di dalam koleksi (*corpus*). Karena itu, kata tersebut dihapus dari dalam dokumen. Contoh data inputan penelitian dan output dalam tahapan *preprocessing* dapat dilihat pada Tabel 1.

Tabel 1. Dokumen sebelum dan setelah tahapan *preprocessing*

Dokumen Sebelum *Preprocessing*
3729. puasa : cara mengqadha puasa yang tidak diketahui jumlahnya pertanyaan >> agus suryo komputro assalamu alaikum wa rahmatullahi wa barakatuh saya mau tanya nih ... jika saya (laki-laki) dulunya lalai dalam beribadah khususnya berpuasa bulan ramadhan sering bolong bolong, saya berniat membayar puasa ramadhan saya yang bolong tapi sudah lupa berapa banyak yang bolong .... apa yang harus saya lakukan berdasar al-quran & hadits yang ada ? syukron jawaban >> ghufron bkl wa'alaikumussalaam warohmatullah wabarokaatuh wajib mengqadha puasa sampai yakin sudah dikerjakan semua. referensi: hawasyi asy-syarwani iii / 396 وَلَوْ عَلِمَ أَنَّهُ صَامَ بَعْضَ اللَّيَالِي وَبَعْضَ الْأَيَّامِ وَلَمْ يَعْلَمْ مِقْدَارَ الْأَيَّامِ الَّتِي صَامَهَا فَظَاهِرٌ أَنَّهُ يَأْخُذُ بِالْيَقِينِ فَمَا تَيَقَّنَهُ مِنْ صَوْمِ الْأَيَّامِ أَجْزَأَهُ وَقَضَى مَا زَادَ عَلَيْهِ سم اهـ حواشي الشرواني ج ٣ ص ٣٩٦ مكتبة دار إحياء التراث العربي "apabila ada seseorang mengetahui bahwa dirinya berpuasa sebagian jatuh pada malam hari (karena tinggal di daerah yang tidak diketahui batas siang dan malamnya), dan sebagian jatuh pada siang hari, sedangkan dia tidak mengetahui jumlah puasa yang dikerjakan pada siang harinya, maka menurut qoul yang jelas orang tersebut wajib mengambil hitungan yang diyakininya, maka hitungan puasa siang hari yang diyakininya itu cukup baginya (untuk dijadikan jumlah puasa siang harinya) dan wajib mengqadha sisanya puasa yang dilakukan pada malam harinya" . wallahu a'lam
Dokumen Setelah *Preprocessing*
puasa mengqadha puasa jumlah laki lalai ibadah puasa bulan ramadhan bolong bolong niat bayar puasa ramadhan bolong lupa bolong laku dasar al-quran hadits wajib mengqadha puasa yakin hawasyi asy-syarwani puasa jatuh malam tinggal daerah batas siang malam jatuh siang jumlah puasa siang orang wajib ambil hitung hitung puasa siang jumlah puasa siang wajib mengqadha sisa puasa malam

C. Proses Pemodelan Kalimat Menggunakan *Convolutional Neural Network*

Selanjutnya akan dilakukan pemodelan semantik terhadap masing-masing kalimat pertanyaan dan jawaban di dalam dokumen. Dalam proses pemodelan semantik, digunakan model CNN. Metode CNN digunakan hanya untuk proses pemodelan kalimat,

bukan untuk proses klasifikasi seperti pada umumnya. *Hyperparameters* yang dibutuhkan dalam menerapkan model CNN, yaitu inputan kata representasi (*word embeddings*), jumlah *convolution filters*, *pooling strategies* (*max-pooling*), dan fungsi aktivasi. Dalam proses pemodelan semantik dengan CNN, terkait *natural language processing*, maka inputan yang digunakan berupa koefisien (nilai) dari masing-masing *word embeddings* terhadap kosa kata yang ada dalam kalimat pertanyaan serta jawaban dimana direpresentasikan sebagai matriks dua dimensi. *Word embeddings* diperoleh melalui cara seperti yang telah dijelaskan pada bab II. Data inputan untuk proses dalam memperoleh *word embeddings* adalah berupa dokumen teks yang telah dipreproses dan output dari proses ini berupa beberapa *word embeddings* yang merepresentasikan suatu kata beserta koefisien *word embeddings*-nya (yang menunjukkan nilai kemiripan atau kedekatan makna antara kata dan *word embeddings*-nya yang diperoleh dari proses *word embeddings*), sehingga satu kata memungkinkan akan memiliki beberapa kata lain yang mungkin memiliki kedekatan makna. Dalam penelitian ini, dari hasil akhir proses *word embeddings* untuk setiap kosa kata, diambil sebanyak 15 kata yang merepresentasikan sebuah kosa kata. Contoh output dari proses *word embeddings* seperti terlihat pada Tabel 2.

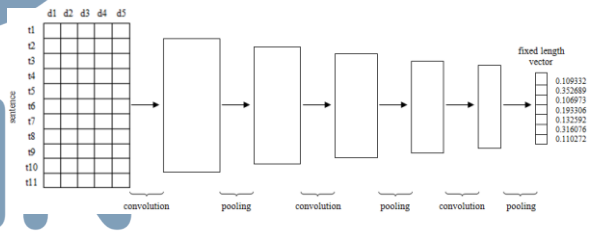
Hal selanjutnya yang dilakukan dalam tahap model CNN adalah mentransformasi semua kata tunggal (*token*) dalam kalimat (dalam hal ini direpresentasikan dalam bentuk koefisien dari masing-masing *word embeddings* yang dimiliki setiap kata tunggal dalam kalimat) menjadi vektor oleh *lookup layer*, kemudian mengubahnya (*encode*) menjadi *fixed-length vector* melalui *convolutional layer* dan *pooling layer* dengan kedalaman layer 3. Dalam penelitian ini, *filters slide* melewati *full rows* sebuah matriks sehingga lebar *filters slide* akan sama dengan lebar matriks inputan (jumlah *word embeddings* yang akan digunakan untuk per kata). Sedangkan untuk tinggi (*region size*) *filters slide* atau *sliding windows* melewati 3 kata. *Convolutional layer* menerapkan matriks filter satu dimensi yang melewati tiap baris fitur dalam matriks kalimat. Pembelitan (*convolving*) filter yang sama dengan *n*-gram di setiap posisi dalam kalimat memungkinkan fitur-fitur untuk diekstrak secara bebas dari posisi mereka dalam kalimat. Gambar 3 menunjukkan gambaran dari proses pemodelan kalimat menggunakan CNN dengan inputan berupa koefisien dari masing-masing *word embeddings* (*d*) terhadap kata tunggal (*t*) dalam kalimat yang akan dimodelkan hingga diperoleh output dari proses ini dalam bentuk *fixed-length vector*.

D. Pencocokan Semantik Antara Kalimat Pertanyaan-Jawaban Menggunakan *Neural Tensor Network*

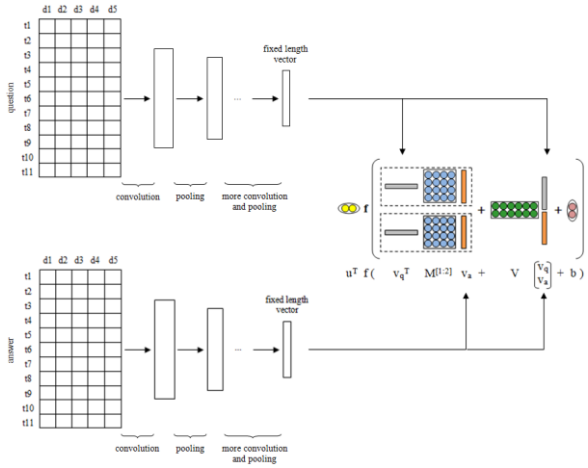
Fixed-length vector dari masing-masing kalimat dokumen pertanyaan dan jawaban yang diperoleh dari proses pemodelan kalimat dengan menggunakan CNN akan diukur kecocokannya. Dalam penelitian ini, pencocokan tersebut dimodelkan dengan *non-linear tensor layer*, dimana sebelumnya sudah pernah digunakan untuk pemodelan interaksi relasional data secara eksplisit [10]. Sebuah pertanyaan *q* dan jawabannya *a*, akan digunakan dua CNN untuk memodelkan keduanya menjadi *fixed vectors q* dan *fixed vectors a*. Berikutnya *Neural Tensor Network*, sebuah *tensor layer* yang diterapkan pada akhir dari CNN untuk memodelkan relasi antara pertanyaan dan jawabannya. Proses tersebut digambarkan seperti pada Gambar 4. *Tensor layer* menghitung kecocokan pasangan pertanyaan-jawaban melalui *score function* seperti pada (1).

Tabel 2. Beberapa kosa kata dan *word embeddings*-nya

Kosa Kata	***Word Embeddings***
puasa	mati; capai; tinggal; bawa; bani; ijtihad; kalang; warga; bulan; alami; salah; saleh; pimpin; anggap; uang
mengqadha	harun; saleh; ahli; dosa; majlis; hasan; ijma; bani; puasa; sedekah; nyata; cari; takwil; anak; makkah
ibadah	uang; nadzar; jasa; capai; witir; santri; negara; qadha; mudah; izin; pesantren; nikah; akibat; ajak; kencing
ramadhan	qobliyah; amal; hadir; qunut; jumat; subuh; tahiyat; maghrib; tidal; majlis; harap; shalatnya; neraka; pahala; fardhu
niat	arah; mutlak; jatuh; taukid; tarik; qobul; batin; talak; hajar; panitia; jil; ijtihad; zain; syar; yakin
bayar	lunas; negara; pajak; berat; kena; angsur; riba; ganti; pecah; izin; qadha; palsu; makkah; akibat; wajib
wajib	rugi; qadha; santri; nadzar; jasa; zakat; negara; fidyah; nishab; akibat; masyhur; daerah; pindah; pajak; berat



Gambar 3. Deskripsi pemodelan kalimat menggunakan CNN



Gambar 4. Pencocokan kalimat antara pertanyaan dan jawaban [4]

$$s(q,a) = u^T f\left(v_q^T M^{[1:r]} v_a + V\begin{bmatrix} v_q \\ v_a \end{bmatrix} + b\right) \quad (1)$$

dimana f = *tanh* adalah *standard nonlinearity*, $M^{[1:r]\in R^{n_S x\, n_S x\, r}}$ adalah sebuah *tensor*, r adalah jumlah *tensor slice*, dan parameter yang lainnya adalah bentuk standar dari *neural network*, $V \in R^{r\, x\, 2n_S}$, $b \in R^r$ dan $u \in R^r$.

Parameter dalam penelitian ini adalah L, W_{CNN}^q, W_{CNN}^a, u, $M^{[1:r]}$, V, dan b. Dimana L adalah *word embeddings*, W_{CNN}^q dan W_{CNN}^a adalah parameter dari CNN untuk pertanyaan dan jawaban, dan parameter lainnya dari *tensor layer*. *Objective function*-nya adalah (2).

$$L = \sum_{(q,a)\in C}\sum_{(q,a')\in C'}[\gamma - s(q,a) + s(q,a')] + \lambda\|\theta\|_2^2 \quad (2)$$

dimana $\gamma > 0$ (dalam penelitian ini nilainya 1) adalah *maximum margin*, dan (q, a') adalah pasangan pertanyaan-jawaban yang salah (random). C adalah kumpulan data latih dari pasangan pertanyaan-jawaban dalam dokumen dan C' menunjukkan kumpulan dari seluruh pasangan pertanyaan-jawaban yang salah. Dan untuk optimasinya, digunakan L-BFGS. Selanjutnya, *fixed-length vector* dari kalimat *query* dan dokumen jawaban akan diukur kecocokannya pada sisi *testing*. Output dari tahapan ini, yaitu berupa nilai kecocokan antara *query* dan jawaban dari arsip dokumen penelitian.

IV. HASIL DAN PEMBAHASAN

Untuk uji coba, jumlah pasangan dokumen yang digunakan terdiri dari 200 dokumen tanya-jawab dan akan diujikan untuk 5 pertanyaan baru (*query*) yang diinputkan penanya. Masing-masing dari *query* tersebut akan dibandingkan terhadap 200 dokumen tanya-jawab dengan tujuan untuk mendapatkan pertanyaan yang relevan terhadap *query* dengan menggunakan metode CNN. Output dari penelitian berupa nilai kecocokan antara *query* dan dokumen jawaban yang diperoleh menggunakan *neural tensor network* dengan menerapkan (1) dan (2) sebagaimana yang telah dijelaskan pada bab III. Output tersebut akan dikembalikan kepada penanya dalam bentuk tampilan dokumen pertanyaan-jawaban dari arsip berdasarkan urutan nilai kecocokan tersebut.

Pertanyaan-jawaban yang dikembalikan dinilai secara objektif apakah relevan atau tidak terhadap *query*. Selanjutnya dilakukan evaluasi dan validasi terhadap dokumen tanya-jawab hasil temu kembali dengan cara mengukur kualitas hasil temu kembali pertanyaan. Parameter yang digunakan untuk mengukur kualitas hasil temu kembali tersebut adalah *mean average precision* (MAP) dimana secara luas digunakan dalam *question retrieval*. *Mean average precision* untuk satu set *query* adalah rata-rata dari nilai presisi rata-rata (*average precision*) untuk setiap *query* yang dihitung menggunakan (3).

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \quad (3)$$

dimana Q adalah jumlah *query* dan *AveP* diperoleh dari (4) untuk masing-masing *query*. *Average precision* (*AveP*) adalah rata-rata nilai presisi yang diperoleh untuk kumpulan k dokumen teratas yang ada setelah setiap dokumen yang relevan di-*retrieve*, dan nilai ini kemudian dirata-ratakan berdasarkan kebutuhan informasi.

$$AveP = \frac{\sum_{k=1}^{n}(P(k)\times rel(k))}{number\ of\ relevant\ documents} \quad (4)$$

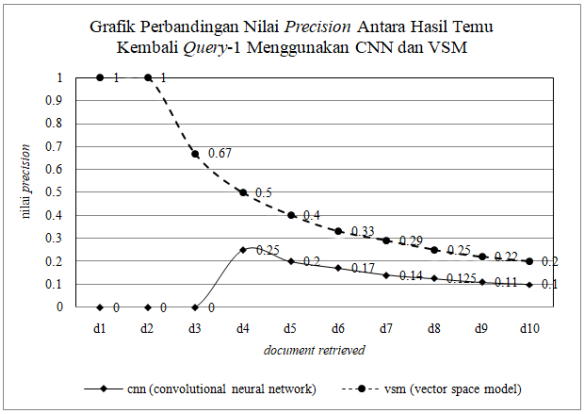
n adalah jumlah dokumen yang di-*retrive*, $P(k)$ adalah nilai presisi dokumen di peringkat k yang dihitung menggunakan (5), dan $rel(k)$ merupakan fungsi indikator yang bernilai 1 jika dokumen di peringkat k adalah dokumen yang relevan, jika tidak maka bernilai 0. Jika dokumen yang relevan tidak di-*retrieve* sama sekali, maka nilai pada (4) dianggap 0 [11].

$$P(k) = \frac{Jumlah\ dokumen\ retrieve\ dan\ relevant\ hingga\ K\ teratas}{K} \quad (5)$$

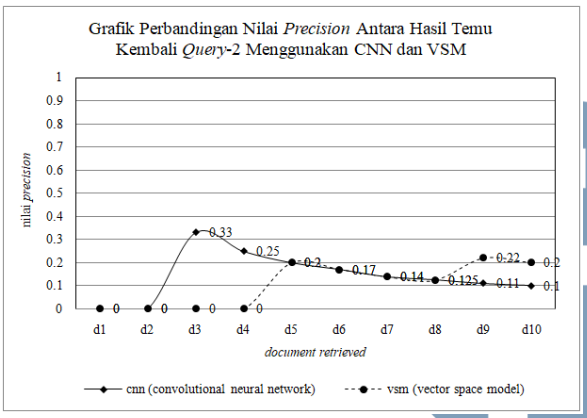
Dalam penelitian ini, nilai *precision* yang dihitung merupakan nilai *precision* pada masing-masing 10 dokumen teratas dari hasil temu kembali (*top-10 retrieved documents*), atau disebut juga sebagai *precision*@10. Kelima *query*, selain di uji cobakan terhadap metode CNN juga diuji cobakan terhadap metode VSM (*vector space model*) dalam tahapan proses mencari dan menemukan dokumen pertanyaan yang sama secara semantik dari dalam koleksi dokumen tanya-jawab terhadap *query*.

Gambar 5 memperlihatkan grafik perbandingan nilai *precision* masing-masing dokumen pertanyaan-jawaban hasil temu kembali (*retrieve*) terhadap *query*-1 yang berada di urutan 10 dokumen teratas hasil pencarian dengan menggunakan metode CNN dan VSM. Nilai *precision* tersebut diperoleh menggunakan (5). 10 dokumen pertanyaan-jawaban yang ditemukan dengan menggunakan metode CNN berbeda dengan yang ditemukan dengan menggunakan metode VSM.

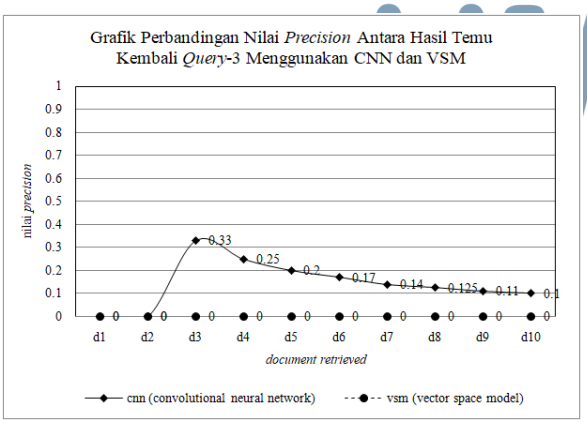
Untuk nilai *precision* dari hasil temu kembali dokumen pertanyaan terhadap masing-masing *query*-2, *query*-3, *query*-4, dan *query*-5 yang berada di urutan 10 dokumen teratas hasil pencarian menggunakan metode CNN dan VSM, dapat dilihat grafik perbandingannya pada Gambar 6, Gambar 7, Gambar 8, dan Gambar 9. Sepuluh dokumen pertanyaan-jawaban yang ditemukan oleh kedua metode tersebut berbeda-beda. Kelima *query* inputan tersebut masing-masing mengembalikan beberapa dokumen pertanyaan-jawaban dari dalam koleksi dokumen tanya-jawab sebagai hasil temu kembali dari proses pencarian.



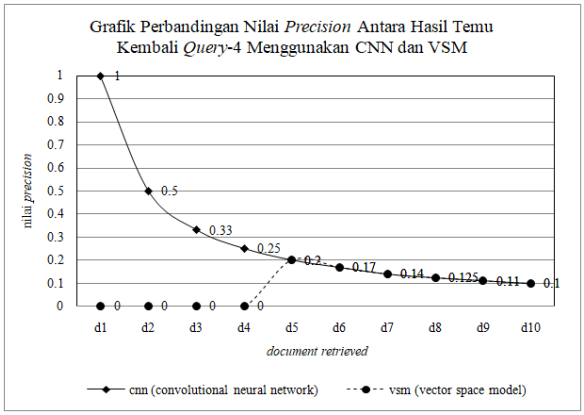
Gambar 5. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query*-1 menggunakan metode CNN dan VSM



Gambar 6. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query*-2 menggunakan metode CNN dan VSM

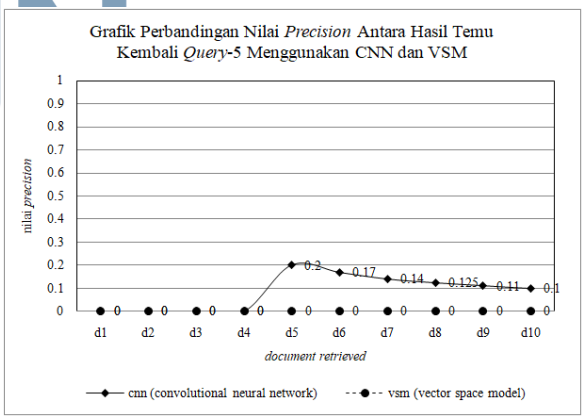


Gambar 7. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query*-3 menggunakan metode CNN dan VSM



Gambar 8. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query*-4 menggunakan metode CNN dan VSM

Dokumen pertanyaan-jawaban yang dikembalikan tersebut, oleh sistem dianggap memiliki kerelevanan dengan *query*. Namun hanya diambil 10 dokumen pertanyaan dalam peringkat teratas yang dikembalikan oleh sistem untuk dijadikan acuan penilaian evaluasi terhadap hasil temu kembali pertanyaan oleh sistem. Dari hasil tersebut, dapat dihitung nilai *average precision*-nya untuk masing-masing *query* inputan menggunakan (4). Sehingga diperoleh nilai *average precision* terhadap hasil temu kembali dokumen pertanyaan menggunakan metode CNN untuk *query* inputan pertama hingga kelima berturut-turut yaitu, 0,25, 0,33, 0,33, 1, dan 0,2. Sedangkan nilai *average precision* terhadap hasil temu kembali dokumen pertanyaan menggunakan metode VSM untuk *query* inputan pertama hingga kelima berturut-turut yaitu, 1, 0,21, 0, 0,2, dan 0.



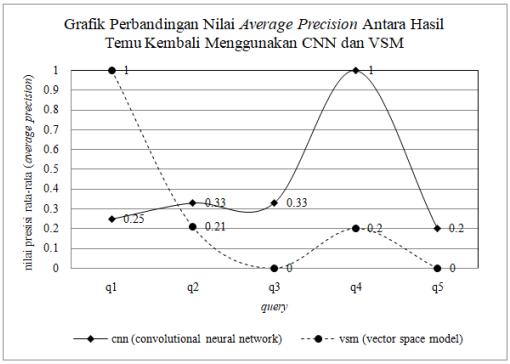
Gambar 9. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query*-5 menggunakan metode CNN dan VSM

Tabel 3. *Average precision* (*AveP*) untuk hasil temu kembali pertanyaan menggunakan CNN dan VSM

***Query* Inputan Ke-**	**Kalimat *Query***	***AveP* (CNN)**	***AveP* (VSM)**
1	assalamualaikum... apakah wajib mencabut sesuatu yang palsu dari tubuh	0,25	1

Query Inputan Ke-	Kalimat *Query*	*AveP* (CNN)	*AveP* (VSM)
	jenazah sebelum dikubur?		
2	perintah membaca ta'awudz dalam al-quran. assalamualaikum... saya pernah dengar seorang khatib ketika akan mengucapkan ayat al-quran begini "qoolallaahhu ta'ala fil qur anil adzim.. a u'dzubillahhi minasysyaithonirrojim" setelah itu baru membaca ayat al-quran. pertanyaan saya : 1. apa khatib itu termasuk berbohong? karena di dalam al-quran tidak ada bacaan ta'awudz. 2. solusinya yang tepat gimana?	0,33	0,21
3	assalamu'alaikum. kepada para kyai dan member. mohon ditakwilkan mengenai mimpi meninggal dunia. sekian dan terimakasih sebelumnya	0,33	0
4	assalamualaikum... bagaimana hukum muamalahnya bensin oplosan / campuran ? apakah termasuk tindak kriminal ?	1	0,2
5	assalaamu'alaikum. afwan, mohon pencerahannya... hal-hal apa sajakah yang di senangi oleh allah swt ? sehingga allah swt memberi ridho dan rahmat-nya kepada orang tersebut ? syukron katsir	0,2	0

Nilai MAP untuk sekumpulan uji coba adalah rata-rata dari nilai *average precision* untuk setiap *query*. MAP dipengaruhi oleh bobot dari setiap *query* yang dilaporkan dalam bentuk penilaian *average precision* tiap *query*, baik itu diperoleh banyak dokumen pertanyaan yang relevan dengan *query* maupun yang sangat sedikit yang relevan dengan *query*. Nilai MAP untuk keseluruhan uji coba terhadap lima *query* yang berbeda tersebut dengan metode CNN dihitung menggunakan (3), sehingga diperoleh nilai MAP-nya yaitu 0,422. Sedangkan nilai MAP untuk keseluruhan uji coba dengan metode VSM yaitu 0,282.



Gambar 10. Grafik perbandingan nilai *average precision* antara hasil temu kembali menggunakan metode CNN dan VSM

V. SIMPULAN

Kemampuan metode *convolutional neural network* dalam menemukan pertanyaan yang sama secara semantik dengan *query* dari dalam arsip dokumen tanya-jawab bernilai 0,422 berdasarkan hasil hitungan *mean average precision*-nya (MAP). Sedangkan pencarian dengan menggunakan *vector space model*, MAP-nya bernilai 0,282. Untuk memperoleh nilai MAP yang lebih besar lagi (mendekati 1), dapat dilakukan penelitian lebih lanjut dengan kemungkinan menggabungkan penggunaan metode pengelompokkan dokumen untuk proses pencarian yang lebih efisien atau penggunaan metode pencarian yang lainnya.

DAFTAR PUSTAKA

[1] K. Wang, Z. Ming, and Tat-Seng Chua, "A Syntactic Tree Matching Approach to Finding Similar Question in Community-based QA Services," in *Proceedings of SIGIR*, 2009, pp. 187-194.

[2] J. Turian, L. Ratinov, and Y. Bengio, "Word Representations: A Simple and General Method for Semi-supervised Learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Sweden, 2010, pp. 384-394.

[3] R. Collobert et al., "Natural Language Processing (Almost) from Scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493-2537, 2011.

[4] Xipeng Qiu and Xuanjing Huang, "Convolutional Neural Tensor Network Architecture for Community-based Question Answering," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 1305-1311.

[5] T. Mikolov and G. Zweig, "Context Dependent Recurrent Neural Network Language Model," in *Spoken Language Technology (SLT)*, 2012, pp. 234-239.

[6] N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, 2013, pp. 1700-1709.

[7] J. Gao, X. He, and Jian-Yun Nie, "Clickthrough-Based Translation Models for Web Search: from Word Models to Phrase Models," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Canada, 2010, pp. 1139-1148.

[8] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of ACL*, 2014.

[9] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen, "Convolutional Neural Network Architectures for Matching Natural Language Sentences," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Canada, 2014, pp. 2042-2050.

[10] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Nevada, 2013, pp. 926-934.

[11] C. D. Manning, P. Raghavan, and H. Schutze, *An Introduction to Information Retrieval*. England: Cambridge University Press, 2009.