Record and Library Journal
https://e-journal.unair.ac.id/index.php/RLJ

Volume 6, No. 1, 2020
e-ISSN: 2442-5168

LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code

Performansi LSTM Network dan OCR untuk Klasifikasi Kode Decimal Dewey Classification

Yesy Diah Rosita dan Yanuarini Nur Sukmaningtyas

Universitas Islam Majapahit

Paper Type:

Research Paper

Abstract

Background of the study: Giving book code by a librarian in accordance with the Decimal Dewey Classification system aims to facilitate the search for books on the shelf precisely and quickly.

Purpose: The first step in giving code to determine the class of books is the principal division which has 10 classes.

Method: This study proposed Optical Character Recognition to read the title text on the book cover, preprocessing the text, and classifying it by Long Short-Term Memory Neural Network.

Findings: In general, a librarian labeled a book by reading the book title on the book cover and doing book class matching with the book guide of DDC. Automatically, the task requires time increasingly. We tried to classify the text without OCR and utilize OCR which functions to convert the text in images into text that is editable. BY the experimental result, the level of classification accuracy without utilizing OCR is higher than using OCR.

Conclusion: The magnitude of the accuracy is 88.57% and 74.28% respectively. However, the participation of OCR in this classification is quite efficient enough to assist a beginner librarian to overcome this problem because the accuracy difference is less than 15%.

Keywords: *librarian, library, text*

Submitted 30 June 2019
Accepted 11 April 2020

* Correspondence:
Yesy Diah Rosita

E-mail:
yesidiahrosita@gmail.com

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

Abstrak

Latar Belakang Masalah: Pemberian kode buku oleh pustakawan sesuai dengan sistem Decimal Dewey Classification (DDC) bertujuan untuk memudahkan pencarian buku di rak secara tepat dan cepat.

Tujuan: Langkah pertama dalam memberikan kode untuk menentukan kelas buku adalah divisi utama yang memiliki 10 kelas.

Metode Penelitian: Secara otomatis, tugas ini membutuhkan waktu yang semakin lama. Penelitian ini mengusulkan Optical Character Recognition (OCR) untuk membaca teks judul pada sampul buku, preprocessing teks, dan mengklasifikasikannya dengan Long Short-Term Memory (LSTM) Network.

Temuan: Secara umum, pustakawan memberi label buku dengan membaca judul buku di halaman depan (sampul buku) dan melakukan kelas buku yang sesuai dengan pedoman DDC. Kami mencoba untuk mengklasifikasikan teks tanpa OCR dan menggunakan OCR yang berfungsi untuk mengubah teks dalam gambar menjadi teks yang dapat diedit. Berdasarkan hasil percobaan, tingkat akurasi klasifikasi tanpa menggunakan OCR lebih tinggi daripada menggunakan OCR.

Kesimpulan Besarnya akurasi adalah masing-masing 88,57% dan 74,28%. Namun, performansi OCR dalam klasifikasi ini cukup efisien untuk membantu pustakawan pemula dalam mengatasi masalah ini karena perbedaan akurasi kurang dari 15%.

Kata Kunci: perpustakaan, pustakawan, teks

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

Pendahuluan

Pada kegiatan pemberian notasi buku, pustakawan mencocokkan *keyword* atau subjek ke bagan klasifikasi buku. Pedoman yang sering digunakan untuk mengklasifikasikan buku berdasarkan subjeknya menggunakan sistem DDC (*Decimal Dewey Classification*). *Decimal Dewey Classification* merupakan sistem klasifikasi perpustakaan yang pertama kali diterbitkan di Amerika Serikat oleh Melvil Dewey (Dewey, 1876). Selain itu, DDC sudah digunakan oleh 200.000 perpustakaan dengan setidaknya 135 negara (Joudrey, Taylor, & Miller, 2015; Service, 2009). Namun, jumlah klasifikasi buku yang terdapat pada DDC relatif banyak, sehingga dibutuhkan waktu dalam pencarian notasi klasifikasi buku. Dalam hal ini, klasifikasi buku disebut sebagai kelas. DDC terbagi menjadi 3 bagian klasifikasi yakni klasifikasi utama, klasifikasi divisi, dan klasifikasi seksi. Jumlah notasi yang terdapat pada masing-masing klasifikasi secara berturut-turut adalah 10 kelas, 100 kelas, dan 1000 kelas.

Pada umumnya, seorang pustakawan mampu mengetahui jenis klasifikasi buku dengan membaca judul buku yang tertera pada halaman depan buku atau lebih dikenal dengan *book cover* karena sudah dapat mewakili isi buku. Untuk mengetahui notasi / kode klasifikasi utama buku, dibutuhkan waktu untuk mencocokkan subjek buku dengan pedoman DDC. Padahal jumlah buku yang diterima oleh perpustakaan sangat banyak, sehingga diperlukan kecepatan dan ketepatan dalam pemberian notasi pada buku. Sebagai langkah awal yang perlu dilakukan adalah pemberian notasi klasifikasi utama. Hal ini diperlukan untuk lebih mempermudah dalam pemberian notasi klasifikasi berikutnya.

Klasifikasi buku dapat dilihat berdasarkan pola teks pada judul buku yakni subjek buku, seperti halnya klasifikasi dokumen (Alfian Sukma, Badrus Zaman, 2015; Bijalwan, Kumar, Kumari, & Pascual, 2014). Terdapat beberapa metode yang dapat digunakan untuk mengetahui pola teks serta mengklasifikasikannya. Namun masing-masing metode yang digunakan pada penelitian-penelitian sebelumnya memiliki tingkat ketepatan yang berbeda. Adapun *state of art* penelitian ini tersaji pada tabel 1.

Tabel 1. *State of Art*

No	Model	Topik	Metode yang digunakan	Pengukuran	Hasil
1.	Junkai Yi, et. al, 2017	Klasifikasi informasi penting pada internet	*Deep-Learning Vocabulary Network* (DLVN), *Vector Space Model* (VSM), *Term Frequency-Inverse*	Jarak *Euclidience*	14.48, 13.75, 13.87
2.	Sandeep Kaur, et.al., 2016	Klasifikasi berita online	*Neural Network Classifier*	*Accuracy*	99.39%
3.	Doaa Mohey El-Din, 2016	Klasifikasi ulasan artikel ilmiah online	*Bag-of-Words Model, Term Frequency-Inverse*	*Accuracy*	81.7%
4.	Chunting Zhou, et. al., 2015	Klasifikasi Teks	*C-Long Short-Term Memory Network*	*Accuracy*	94,60%
5.	Vishwanath Bijalwan, et. al., 2014	Klasifikasi artikel berita	*Naive Bayes, Term Graph, dan K Nearest Neighbour (KNN)*	*Accuracy*	62.66%, 98.73%,
6.	Faisal Mohammad, et. al., 2014	Menerjemahkan tulisan tangan berbentuk gambar ke dalam bentuk teks yang bersifat editable	Model *Optical Character Recognition (OCR)*	Tingkat Pengenalan	75%

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal*, *6*(1). 45-56.

Sistem yang akan dibangun untuk menyelesaikan permasalahan tersebut terdiri dari 3 tahap, yaitu: tahap pembacaan judul teks buku (*book cover*), tahap ektraksi teks, dan tahap klasifikasi teks. Konsep yang akan digunakan untuk membaca teks pada *book cover* adalah *Optical Character Recognition* (OCR). Konsep ini merupakan cabang disiplin ilmu dari *computer science* yang mempunyai fungsi pembacaan teks yang berasal dari hasil *scan* gambar yakni menerjemahkan bentuk tulisan pada gambar ke dalam bentuk teks yang dapat di *edit* (Isheawy & Hasan, 2015; Mohammad, Anarase, Shingote, & Ghanwat, 2014; Vijayarani & Sakila, 2017). Hasil dari pembacaan teks pada gambar akan diekstrak terlebih dahulu untuk mendapatkan karateristik teks seperti kata dasar dan penghapusan kata yang tidak diperlukan (contoh: yang, ke, pada, dan, terdapat, dan sebagainya). Kemudian hasil ektraksi teks disimpan sistem dalam sebuah dataset (tempat penyimpanan data) sebagai data *training*. Data ini digunakan sistem dalam pelatihan (*training*) untuk pengklasifikasian kelas buku hingga membentuk sebuah *classifier* (pengklasifikasi). Metode yang digunakan untuk membentuk *classifier* yakni *Neural Network* (NN) karena merupakan teknik klasifikasi teks dengan akurasi lebih dari 90% (Kaur & Khiva, 2016). Setelah didapatkan *classifier* dengan performansi yang baik maka *classifier* tersebut dapat digunakan untuk mengklasifikasikan judul pada *book cover*. Adapun tujuan utama dari penelitian ini yaitu pustakawan tingkat pemula mampu mengetahui klasifikasi utama buku berdasarkan judul buku dengan lebih mudah dan tepat dengan memanfaatkan kamera untuk mengambil gambar *book cover*. Hal ini sangat diperlukan sebagai langkah awal untuk mengetahui bagian klasifikasi berikutnya karena tergantung dari klasifikasi utama.

Berdasarkan *state of art* pada tabel 1 maka model dan metode yang digunakan pada penelitian ini diantaranya: *Optical Character Recognition* (OCR), *Bag-of-Word, Term Frequency - Inverse Document Frequency* (TF-IDF) serta *Neural Network* (NN).

Optical Character Recognition (OCR)

Optical Character Recognition (OCR) merupakan proses penerjemah teks yang mengubah gambar ke dalam bentuk teks yang berbentuk *editable* (dapat di-*edit*) (Isheawy & Hasan, 2015; Mohammad et al., 2014; Vijayarani & Sakila, 2017). Hasil penerjemahan ini mengakibatkan ukuran *file* teks yang berbentuk *editable* jauh lebih kecil daripada teks yang berbentuk gambar. Dengan kata lain, hasil penerjemahan teks yang berbentuk gambar menjadi karakter ASCII. Adapun mekanisme penggunaan model OCR sebagai berikut:

a) *Binarization*, proses konversi gambar RGB (*red green blue color*) ke biner (*black and white color*).
b) *Noise Filtering*, proses untuk menghilangkan bagian *piixel* yang dianggap tidak penting.
c) *Segmentation*, pemisahan area pada setiap karakter yang terdeteksi.
d) *Normalization*, proses penskalaan (*scaling*) dan penebalan (*thinning*) karakter agar didapatkan ukuran yang sama.
e) *Feature Extraction*, proses pengambilan fitur-fitur setiap karakter.
f) Pencocokan karakter, penghitungan jarak kedua vektor karakter (acuan dan input) dengan menggunakan perhitungan jarak *Euclidience Distance*

Bag-of-Words (BoW)

Bag-of-words merupakan teknik yang banyak digunakan untuk *sentiment analysis* namun mempunyai 2 kelemahan yakni penggunaan evaluasi secara manual untuk leksikon dalam penentuan evaluasi kata-kata dan menganalisis tingkat sentimental kata dengan akurasi rendah karena mengabaikan efek tata bahasa dari kata-kata dan mengabaikan semantik kata-kata (Mohey, 2016). Selain itu, BoW dikenal juga sebagai *term-frequency counter* (Mathworks, 2018b). Model *bag-of-words* mencatat berapa kali kata-kata itu muncul di setiap kumpulan dokumen. Adapun tahapannya anatara lain:

a) *Tokenization*, proses pembagian teks ke dalam bagian sendiri-sendiri. Misal terdapat sebuah teks "majapahit terletak di mojokerto" maka menjadi "majapahit," "terletak," "di," dan

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

"mojokerto." Proses ini juga disisipi fungsi *lowercase* yang bertujuan menjadikan semua kata berbentuk huruf kecil.

b) *Remove Stop Word*, proses penghilangan kata-kata yang tidak penting yang telah ditentukan. Misal, daftar kata yang tidak penting seperti "yang," "di," "ke," "pada," "saat," dsb. Kemudian dicocokkan dengan hasil *tokenization* keberadaan *stop word*. Jika ada, maka kata tersebut dihilangkan.

c) *Vectorization*, proses konversi teks ke bentuk vektor dengan mencocokkan *unique words* (kata unik). Misal terdapat 10 *unique* words antara lain "universitas," "islam," "majapahit," "kerajaan," "mojokerto," "gajah," "mada," "kampus," "kuliah," "museum." Kemudian jika terdapat sebuah teks berisi "universitas islam majapahit mojokerto panorama hijau" menjadi [1 1 1 1 0 0]. Nilai 1 merupakan jumlah kata *unique words* yang terdapat pada teks tersebut dan sebaliknya.

Term Frequency – Inverse Document Frequency (TF-IDF)

Metode ini digunakan untuk mengevaluasi pentingnya kata pada sebuah teks atau dokumen dengan pemberian bobot (Man Lan, Chew Lim Tan, Jian Su, & Yue Lu, 2009).

a) *Term Frequency* (TF), merupakan jumlah kemunculan kata pada dokumen.

$$TF = \begin{cases} 1 + \log_{10} f(t,d); f(t,d) > 0 \\ 0; f(t,d) = 0 \end{cases} \quad (1)$$

Keterangan:
$f(t,d)$: jumlah kata/term pada sebuah dokumen.

b) TF *normalization*, merupakan perbandingan frekuensi kata/term dengan nilai maksimum dari keseluruhan frekuensi kata pada sebuah dokumen.

$$TF\_Normalization = 0.5 + 0.5x\left[\frac{f(t,d)}{\max\{f(t',d) : f(t',d), d \in d\}}\right] \quad (2)$$

c) *Inverse Document Frequency* (IDF), merupakan proses yang mampu menunjukkan bahwa seberapa besar keterkaitan kata terhadap keseluruhan dokumen.

$$IDF_j = \log(D / df_j) \quad (3)$$

Keterangan:
D : jumlah keseluruhan dokumen
df_j : jumlah dokumen yang mengandung kata/term yang dimaksud.
Kemudian dilanjutkn dengan pembobotan dengan menggunakan persamaan 4.

$$w_{ij} = tf_{ij}.\log(D / df_j) + 1 \quad (4)$$

Keterangan:
tf_{ij} : *term frequency* pada dokumen
D : jumlah dokumen keseluruhan
df_j : jumlah dokumen yang mengandung *term*/kata
Hasil dari pembobotan ini akan digunakan untuk mengetahui kecenderungan kelas dokumen.

Long Short-Term Memory (LSTM) *Network*

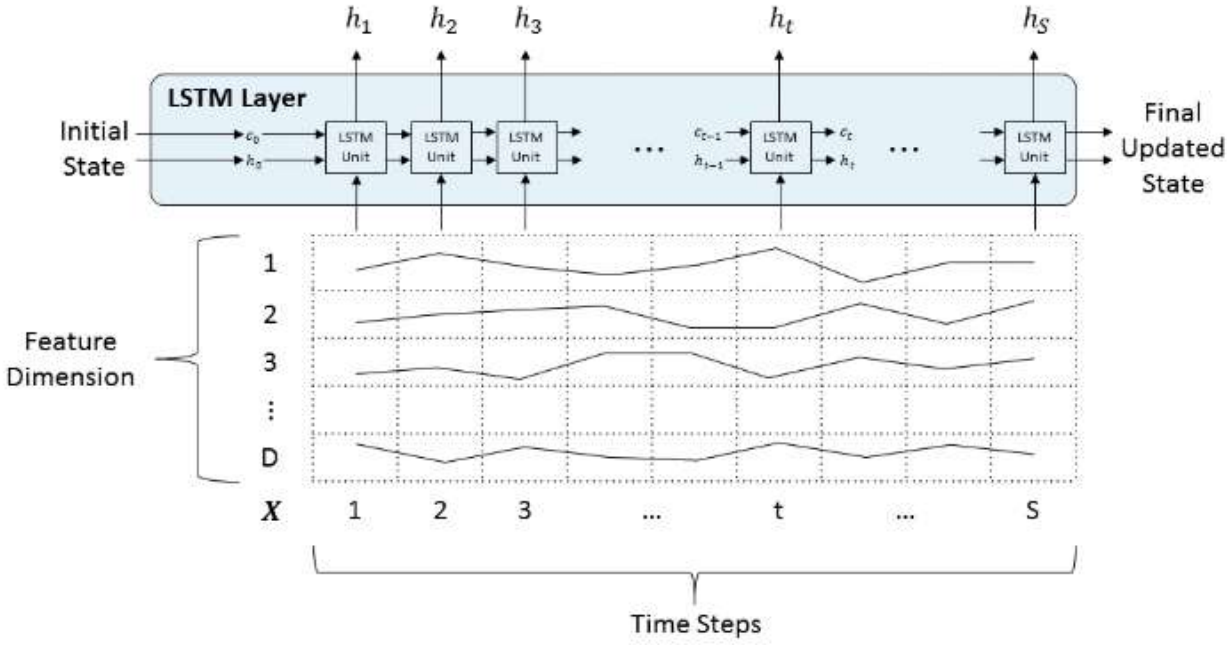
Sebuah merupakan kata-kata terurut yang saling memiliki ketergantungan satu dengan lainnya. LSTM *Network* adalah jenis *Recurrent Neural Network* (RNN) yang mampu mempelajari ketergantungan tersebut (Service, 2009; Zazo, Lozano-Diez, Gonzalez-Dominguez, Toledano, & Gonzalez-Rodriguez, 2016; Zhou, Sun, Liu, & Lau, 2015). Mekanisme LSTM *network* sama seperti *Neural Network* (NN) lainnya. Namun, untuk pengklasifikasian data teks, diperlukan pengonversian data teks ke dalam bentuk numerik dengan fungsi *word embeddings*. Fungsi ini dapat mengetahui secara detail semantik dari kata-kata, sehingga apabila terdapat kata-kata yag mirip, maka akan

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

menghasilkan vektor yang sama. Misal, jika digambarkan hubungan antara kata-kata dengan aritmatika vektor seperti teks yang berisi "*king is to queen as man is to woman*" menjadi persamaan "*king*" – "*man*" + "*woman*" = "*queen*".

Pertama, unit LSTM menentukan status awal jaringan dan langkah pertama dari X_1 *squence*, lalu menghitung *output* pertama h_1 dan memperbarui *cell state* c_1. Pada saat langkah *t*, unit menempati *current state* jaringan (c_{t-1}, h_{t-1}) dan langkah berikutnya X_1 *squence*, dan menghitung output h_1 serta memperbarui *cell state* c_t.

Layer state terdiri dari *output state* (*hidden state*) dan *cell state*. *Output state* pada *step t* terdiri *output* dari layer LSTM. *Cell state* terdiri informasi pembelajaran dari *step* sebelumnya. Pada setiap *step*, layer menambahkan atau mengurangi informasi dari *cell state* dimana layer mengontrol perubahan ini menggunakan *gates*.



Gambar 1. Arsitektur LSTM Networks (Sumber: http://mathworks.com)

Tabel 2. Komponen Layer LSTM Network

No.	Komponent	Fungsi
1.	Input gate (*i*)	Mengontrol level perubahan *cell state*
2.	Forget gate (*f*)	Mengontrol level ulang *cell state* (lupa)
3.	Layer input (*g*)	Menambah informasi pada *cell state*
4.	Output gate (*o*)	Mengontrol level *cell state* ditambahkan ke *output state*

Metode Penelitian

Sebelum dilakukan penelitian, terdapat beberapa hal yang harus dipersiapkan terlebih dahulu, seperti *tool* yang digunakan dalam penelitian ini. *Tool* yang digunakan untuk mendukung penelitian ini adalah *software* Matlab R2018a. *Software* ini sangat populer digunakan para peneliti tingkat internasional karena fitur-fitur yang dimilikinya hampir lengkap untuk dipergunakan para peneliti. Sedangkan data yang digunakan adalah informasi mengenai buku perpustakaan.

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal*, *6*(1). 45-56.

Dataset

Data diambil secara acak mulai dari perpustakaan kampus terdekat (Mojokerto-Jombang, Jawa Timur, Indonesia) dengan mengamati judul buku dan kode DDC pada label buku yang tertera dan sudah ditetapkan oleh pustakawan setempat. Selain itu, data diperoleh dari *dataset* yang terdapat di internet seperti Github dataset (Brian Kenji Iwana, Rizvi, Ahmed, Dengel, & Uchida, 2017), OCLC (*Online Computer Library Center*), dan perpustakaan *online* Oxford University. Alasan utama pengambilan data pada perpustakaan adalah klasifikasi buku telah diverifikasi dan ditetapkan oleh pustakawan sesuai pedoman DDC. Data yang diperoleh berupa judul buku, kode DDC, dan gambar cover buku untuk digunakan saat uji coba. Data dalam hal ini dibagi menjadi 2 bagian yakni data *training* dan data *testing*. Jumlah data sebanyak 15.000 data yang terdiri dari 14.930 data *training* dan sisanya dipergunakan untuk data *testing*. Parameter yang digunakan sebagai kelas atau *output* dari penelitian ini adalah kode DDC klasifikasi utama. Banyaknya jumlah data yang digunakan ini menyebabkan waktu yang dibutuhkan untuk pengumpulan data berlangsung selama kurang lebih 4 bulan. Peneliti melakukan pemindahan data per buku. Berbeda dengan data yang diperoleh dari Github *dataset* yang diperoleh dari Amazon.com (B. K. Iwana, Rizvi, Ahmed, Dengel, & Uchida, 2016). Peneliti hanya men-*download* data dari sumber tersebut yang telah disediakan meskipun kelas dari data tersebut bukan merupakan kode DDC namun dapat dikonversikan dengan kategori kelas yang ada. Proses konversi ini dapat dilakukan karena pada kategori kelas pada *dataset* tersebut merupakan subjek buku yang termasuk klasifikasi utama. Tingkat performansi sebuah *classifier* (pengklasifikasi) teks yang baik tidak hanya dipengaruhi banyaknya data yang digunakan sebagai data *training*, namun perlu diperhatikan kualitas teks yang digunakan apakah mengandung unsur ambiguitas atau bukan.

Tabel 3. Kode Divisi Utama *Decimal Dewey Classification* (DDC) *System*

No.	Kode	Subyek
1.	000	Computer science, information and general works
2.	100	Philosophy and psychology
3.	200	Religion
4.	300	Social sciences
5.	400	Language
6.	500	Pure Science
7.	600	Technology
8.	700	Arts and recreation
9.	800	Literature
10.	900	History and geography

Terdapat perbedaan antara data yang dipergunakan untuk proses *training* dan *testing* yakni gambar cover buku. Pada data *training* tidak mempergunakan gambar *cover* buku dan sebaliknya. Hal ini dimaksudkan untuk proses ujicoba ketika pemanfaatan OCR dalam pembacaan teks pada gambar sebelum dilakukan klasifikasi teks.

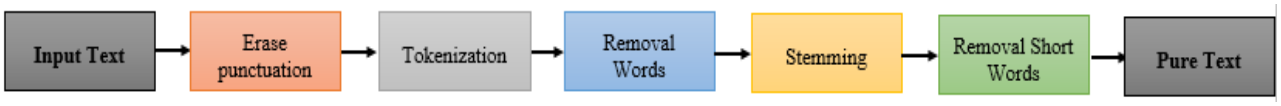
Desain Sistem

Pada penelitian ini, perlu dilakukan pembentukan sebuah *classifier* teks kelas buku

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

berdasarkan judul buku tersebut. Langkah yang pertama adalah teks pada *data training* dilakukan *pre-processing* terlebih dahulu yakni proses menormalisasikan teks sehingga didapatkan kata-kata yang murni tanpa ada unsur yang tidak diperlukan seperti imbuhan, tanda baca, nomer, dan kata-kata yang termasuk dalam *stop word list*. Tahapan *pre-processing* terdiri dari *lower case convertion*, *erase punctuation*, tokenization, *removal stop word list*, *stemming*, dan *removal short words*.

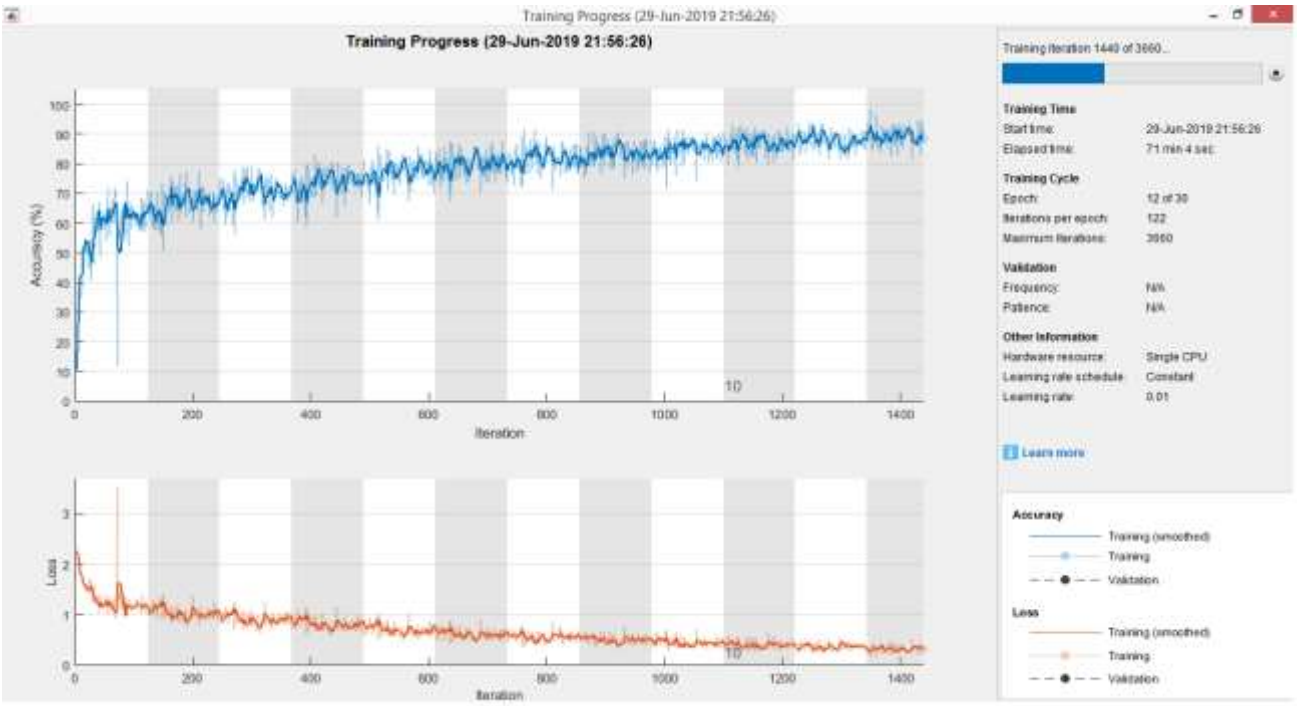
a) *Lower case convertion* adalah proses mengubah huruf besar pada teks menjadi huruf kecil. Hal ini bertujuan untuk menyeragamkan huruf pada teks. Contoh, terdapat sebuah teks berisi "*Decision Support System for Business*.", maka hasil yang didapatkan adalah "*decision support system for business*".
b) Tahap kedua adalah *erase punctuation* yakni menghilangkan tanda baca atau segala bentuk simbol yang notabene tidak memiliki makna yang penting pada teks, seperti tanda baca titik (.), tanda tanya (?), tanda seru (!), dan sebagainya. Contoh, hasil *lower case convertion* sebelumnya dapat dihasilkan teks "*decision support for system business.*" Simbol titik (.) dianggap simbol yang tidak memiliki makna penting atau suatu pertimbangan untuk menentukan kelas teks.
c) Tahap selanjutnya adalah *tokenization*. Proses ini memecah konten tekstual menjadi kata-kata, istilah simbol, atau beberapa elemen bermakna lainnya yang selanjutnya disebut sebagai token (Vijayarani & Janani, 2016). Namun, token dipisahkan oleh karakter spasi. Contoh terdapat sebuah teks berisi "*Decision Support System for Business*", ketika dilakukan *tokenization* maka dihasilkan 5 token yakni "*decision*", "*support*", "*system*", "*for*", dan "*business*".
d) Tahap keempat adalah *removal words*. Istilah *words* dalam hal ini merupakan kata-kata yang tidak memiliki makna penting atau suatu pertimbangan untuk menentukan kelas teks seperti halnya *punctuation* (tanda baca) sebelumnya. Contoh kata yang dimaksud adalah "*for*", "*of*", "*a*", "the," dan semacamnya. Berdasarkan hasil *tokenization*, apabila dilakukan *removal words* maka menghasilkan 4 token yakni "*decision*", "*support*", "*system*", dan "*business*".
e) Tahap kelima adalah *stemming* yang berfungsi menghilangkan imbuhan baik awalan maupun akhiran. Tujuan dari tahap ini adalah menghasilkan kata dasar atau mendekati kata dasar. Contoh, terdapat kata "*business*", apabila dilakakukan *stemming* maka menghasilkan "*busi*" karena kata "*ness*" merupakan imbuhan sehingga jumlah huruf pada kata tersebut menjadi berkurang.
f) Tahap terakhir adalah *removal short words*. Proses ini bertujuan menghilangkan semua kata yang terdiri dari jumlah kata yang relatif sedikit karena dianggap sebagai kata hubung seperti "*of*", "*the*", "*for*", dan semacamnya.



Gambar 2. Tahapan *Preprocessing* Teks

Hasil *preprocessing* teks pada *data training* dilakukan proses *training* untuk menghasilkan sebuah *classifier*. Metode yang digunakan untuk membentuk classifier ini adalah *Long Short-Term Memory* (LSTM) *Neural Network*. LSTM menunjukkan kinerja yang sangat tinggi dalam berbagai aplikasi meskipun memiliki kompleksitas dan komputasi yang relatif tinggi (Salehinejad, Sankar, Barfett, Colak, & Valaee, 2017). Dalam penggunaan jaringan LSTM untuk data teks, terlebih dahulu mengubah data teks menjadi urutan numerik dengan menggunakan *word embeddings* yakni memetakan kata dalam kosakata ke vektor numerik (Mathworks, 2018a). *Embeddings* ini dapat menangkap detail semantik dari kata-kata, sehingga kata-kata yang mirip memiliki vektor yang sama. Mereka juga memodelkan hubungan antara kata-kata melalui aritmatika vektor. Sebagai contoh, hubungan "raja adalah ratu seperti laki-laki dengan perempuan" dijelaskan oleh persamaan raja - laki-laki + perempuan = ratu.

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal*, *6*(1). 45-56.



Gambar 3. Grafik Pembentukan *Classifier* dengan LSTM Network

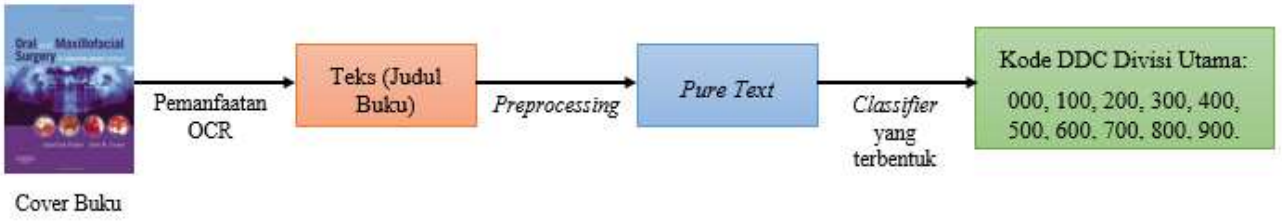
Terdapat 3 tahapan dalam proses *training* menggunakan LSTM Network, antara lain:

a) Impor data dan dilanjutkan *preprocessing* data. Data terdiri dari 2 kolom yang memiliki *header Content* dan *Label*. *Header content* berisi judul buku, sedangkan *label* berisi kelas judul buku, yakni kode DDC divisi utama.
b) Ubah kata-kata menjadi vektor numerik dengan melatih (*train*) *word embeddings*.
c) Pembuatan, sekaligus melatih jaringan LSTM menggunakan urutan vektor kata. Pada dasarnya, LSTM merupakan salah satu metode pengembangan dari *neural network* yang melibatkan juga *layer input*. Ukuran *input* sama dengan dimensi dari *embedding* kata sedangkan ukuran *output* ditentukan menjadi 180, serta jumlah kelasnya sesuai dengan jumlah kode DDC divisi utama. Dalam penggunaan *layer* LSTM untuk masalah klasifikasi *sequence-to-label*, diatur *mode output* menjadi "*last*" dan inisialiasi *learn rate* (tingkat pembelajaran *classifier*) sebesar 0.01. Selanjutnya, menambahkan *layer* yang sepenuhnya terhubung dengan ukuran yang sama dengan jumlah kelas, *layer softmax*, dan *layer* klasifikasi.

Pada saat proses *training* ditunjukkan tingkat grafik *accuracy* dan *loss*. Grafik tersebut akan menggambarkan seberapa besar performansi dari *classifier* yang dibentuk, semakin *accuracy* mendekati nilai 100 dan *loss* semakin mendekati nilai 0 maka performansi *classifier* yang dibentuk semakin baik. Selain itu, grafik tersebut akan menunjukkan jumlah iterasi, waktu yang dibutuhkan *training* serta semua yang berhubungan dengan kedua hal tersebut.

Ketika sebuah *classifier* telah didapatkan, maka *classifier* tersebut dapat dipergunakan untuk mengklasifikasikan data *testing*. Pada penelitian ini, proses pengklasifikasian teks tidak dilakukan secara langsung. Namun, pemanfaatan *Optical Character Recognition* (OCR) untuk membaca teks judul yang terdapat pada cover buku. Proses ini melalui *scanning* (pemindahan) cover buku yang berupa gambar dikonversikan dalam bentuk teks, sehingga teks tersebut dapat dilanjutkan ke proses.

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.



Gambar 4. Tahapan Pengklasifikasian Jenis Buku berdasarkan Judul Buku

Hasil dan Diskusi

Uji coba dilakukan terhadap data *testing* dengan jumlah 70 data. Masing-masing kelas terdiri dari 10 macam judul buku. Uji coba yang dilakukan ini terdiri dari dari 2 perlakuan, yaitu klasifikasi teks judul secara langsung dan klasifikasi teks judul buku yang diperoleh dari penggunaan OCR. Hal ini diperlukan untuk mengetahui performansi LSTM dan OCR itu sendiri apakah lebih baik menggunakan OCR atau sebaliknya. Selain itu, dipertimbangkan pula dari segi efisiensinya dalam penentuan kelas buku dengan menggunakan cover buku yang berisi judul buku.

Berdasarkan hasil ujicoba yang dilakukan didapatkan tingkat akurasi yang berbeda. Klasifikasi teks secara langsung tanpa memanfaatkan OCR mencapai 88.57%. Namun, ketika penggunaan OCR dalam pembacaan judul teks pada cover buku, *classifier* yang dibentuk dapat mengklasifikasikannya sebesar 74.28%. Hal ini, disebabkan karena pemanfaatan OCR terdapat 6 tahapan hingga didapatkan teks, sehingga sangat dimungkinkan terjadinya pengikisan jumlah kata dan kata-kata yang tak memiliki arti. Selain itu, masih terdapat beberapa hal yang perlu diperhatikan. Hasil *scanning* cover buku yang dihasilkan memiliki tingkat kecerahan yang rendah sehingga mempengaruhi kejelasan tulisan. Di lain hal, apabila *cover* buku di-*scan* secara keseluruhan maka penangkapan judul teks disertai dengan nama penerbit, penulis, dan kota/tempat sehingga menambah kata-kata yang tidak penting ketika dilakukan pengklasifikasian.

Dilihat dari sisi performansi LSTM *Network*, penggunaan data *training*, adanya teks-teks yang bersifat ambigu terhadap kode DDC divisi utama sebagai kelas buku, juga mempengaruhi tingkat akurasi dari *classifier* tersebut. Sebagai contoh, terdapat suatu teks mengandung unsur tulisan "*computer*" yang termasuk kategori *Computer science, information and general works* dengan kode DDC divisi utama 000 juga termasuk kategori *Technology* dengan kode DDC divisi utama 600. Hal ini mempengaruhi tingkat akurasi uji coba, karena tergantung kualitas dan jumlah data yang digunakan pada *data training*, selain arsitektur jaringan LSTM Network yang dibentuk. Meskipun begitu, dari segi efisiensi dalam pengklasifikasian buku yang dapat diaplikasikan oleh seorang pustakawan pemula cukup membantu dalam pemberian notasi buku sebagai langkah awal untuk mengetahui klasifikasi berikutnya.

Kesimpulan

Banyaknya tahapan pada *preprocessing* memiliki kelebihan dan kelemahan. Kelebihannya adalah teks yang didapatkan terbebas dari kata-kata/simbol yang tidak memiliki makna penting, sehingga didapatkan teks dengan kata dasar yang mendekati sempurna. Sebaliknya, semakin banyak tahapan *preprocessing* yang diperlukan maka secara otomatis durasi proses komputasi juga semakin tinggi.

Pembentukan sebuah *classifier* teks dengan LSTM Network tergantung dari data *training* yang digunakan. Data tersebut memiliki jumlah data yang cukup dan terhindar dari data yang bersifat ambigu. Selain hal tersebut, pembentukan jaringan LSTM mempunyai peran penting juga untuk menghasilkan performansi sebuah *classifier*. Terdapat inisialiasi variabel yang diperlukan seperti ukuran *input* dan *output*, *learn rate*, *output mode*, layer *softmax*, dan *layer* klasifikasi.

Penggunaan OCR ditujukan untuk membantu seorang pustakawan dalam penginputan judul

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal, 6*(1). 45-56.

teks secara efisien dengan cukup mengarahkan *cover* buku terhadap *web camera* atau *scanner*. Namun perlu dicatat, proses *scanning* tidak perlu dilakukan secara keseluruhan pada *cover* buku karena yang dibutuhkan hanya judul buku saja. Hal ini dimaksudkan untuk mendapatkan tingkat akurasi yang baik. Penggunaan OCR sebelum dilakukan pengklasifikasian dengan LSTM Network, memiliki tingkat akurasi lebih rendah dibandingkan tanpa menggunakan OCR, dengan selisih akurasi sebesar 14.29%. Hal ini, disebabkan dengan beberapa kemungkinan seperti, tulisan pada *book cover* dianggap kurang jelas sehingga dianggap sebagai *noise* (kotoran) dan spesifikasi kamera yang digunakan untuk *capturing/scanning* sebuah *cover* buku yang dapat mempengaruhi kualitas gambar untuk memperoleh teks buku. Hasil eksperimen yang telah dilakukan menunjukkan bahwa penggunaan LSTM dapat bekerja dengan baik untuk klasifikasi teks buku, namun dapat juga dipengaruhi oleh hasil penggunaan dari OCR itu sendiri.

Ucapan Terima Kasih

Kami ucapkan terima kasih kepada Kementerian Riset, Teknologi, Pendidikan Tinggi (Kemenristekdikti) Republik Indonesia atas pendanaan sepenuhnya pada penelitian ini.

Referensi

Alfian Sukma, Badrus Zaman, E. P. (2015). Klasifikafi Dokumen Temu Kembali Informasi dengan K-Nearest Neghbour. *Record and Library Journal of Airlangga University*, *1*, 129–138.

Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, *7*(1), 61–70. https://doi.org/10.14257/ijdta.2014.7.1.06

Dewey, M. (1876). *Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library (Project Gutenberg eBook)*.

Isheawy, N. A. M., & Hasan, H. (2015). Optical Character Recognition (OCR) System. *IOSR Journal of Computer Engineering Ver. II*, *17*(2), 2278–2661. https://doi.org/10.9790/0661-17222226

Iwana, B. K., Rizvi, S. T. R., Ahmed, S., Dengel, A., & Uchida, S. (2016). *Judging a Book by its Cover*. *arXiv prep*.

Iwana, Brian Kenji, Rizvi, S. T. R., Ahmed, S., Dengel, A., & Uchida, S. (2017). Judging a Book By its Cover. *Computer Vision and Pattern Recognition*.

Joudrey, D. N., Taylor, A. G., & Miller, D. P. (2015). *Introduction to Cataloging and Classification, 11th Edition*.

Kaur, S., & Khiva, N. K. (2016). *Internal News Classification Using Deep Learning*. *1*(1), 31–35.

Man Lan, Chew Lim Tan, Jian Su, & Yue Lu. (2009). Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(4), 721–735. https://doi.org/10.1109/TPAMI.2008.110

Mathworks. (2018a). LSTM Layer. In *Matlab Documentation*.

Mathworks. (2018b). Texture Analysis.

Mohammad, F., Anarase, J., Shingote, M., & Ghanwat, P. (2014). Optical Character Recognition Implementation Using Pattern Matching. *Iternational Journal of Computer Science and Information Technologies*, *5*(2), 2088–2090.

Mohey, D. (2016). Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, *7*(1), 244–252. https://doi.org/10.14569/ijacsa.2016.070134

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). *Recent Advances in Recurrent Neural Networks*. 1–21. Retrieved from http://arxiv.org/abs/1801.01078

Service, D. (2009) *Offers Library Users Familiarity and Consistency of a Timehonored Classification System Used in 200,000 Libraries Worldwide*.

Vijayarani, S., & Janani, R. (2016). Text Mining: open Source Tokenization Tools – An Analysis. *Advanced Computational Intelligence: An International Journal (ACII)*, *3*(1), 37–47. https://doi.org/10.5121/acii.2016.3104

Vijayarani, S., & Sakila, A. (2017). Online Optical Character Recognition (OCR) Tools - Performance Analysis. *International Journal of Advanced Research in Computer and Communication Engineering*, *6*(1), 55–58. https://doi.org/10.17148/IJARCCE

Zazo, R., Lozano-Diez, A., Gonzalez-Dominguez, J., Toledano, D. T., & Gonzalez-Rodriguez, J.

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal*, *6*(1). 45-56.

(2016). Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks. *PLoS ONE*, *11*(1). https://doi.org/10.1371/journal.pone.0146917

Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). *A C-LSTM Network for Text Classification*. Retrieved from http://arxiv.org/abs/1511.08630

To cite this document:
Rosita, Y. D. & Sukmaningtyas, Y. N. (2020). LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code. *Record and Library Journal*, *6*(1). 45-56.