USULAN PENELITIAN S1

AUTHORSHIP ATTRIBUTION UNTUK TEKS BERBAHASA INDONESIA DENGAN METODE MULTI-TASK LEARNING BERBASIS DEEP LEARNING

RICKY SETIAWAN
17/412652/PA/17971

PROGRAM STUDI ILMU KOMPUTER
DEPERTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2021

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Authorship attribution merupakan suatu permasalahan klasifikasi yang bertujuan untuk menentukan penulis dari suatu teks berdasarkan kumpulan data yang terdiri dari penulis dan tulisannya. Penelitian terkait dengan topik ini masih terus dipelajari karena terkait dengan aplikasi forensik yang penting seperti mengidentifikasi penulis dari pesan anonim dari forum ekstrimis, verifikasi penulis dari pesan yang bersifat mengancam dari suatu surat elektronik ataupun media sosial, dll. (Abbasi dan Chen, 2005), maupun penelitian tentang kemanusiaan dan sejarah, misalnya memprediksi penulis dari novel yang diterbitkan secara anonim atau dengan nama samaran, verifikasi keaslian tulisan dari suatu penulis, dll. (Juola, 2013).

Beberapa penelitian terkait *authorship attribution* fokus pada prediksi himpunan tertutup di mana diasumsikan bahwa penulis dari teks yang akan diprediksi merupakan anggota dari himpunan kandidat penulis. Dari sudut pandang pembelajaran mesin, hal ini dapat dilihat sebagai permasalahan klasifikasi *multi-class single-label* seperti pada penelitian Mohsen, dkk. (2016), Miura, dkk. (2017), Jiang, dkk. (2018), dan Tang, dkk. (2019).

Salah satu permasalahan utama dalam pendekatan *authorship attribution* adalah mendefinisikan fitur untuk mengukur gaya penulisan atau dalam penelitian biasa disebut sebagai *stylometry* (Holmes, 1994). Oleh karena itu, terdapat berbagai jenis pengukuran termasuk panjang kalimat, panjang kata, frekuensi kata, frekuensi karakter, dan kekayaan kosa kata telah diusulkan. Rudman (1998) mengestimasi terdapat kurang lebih 1.000 pengukuran berbeda yang telah diusulkan.

Penelitian terkait *authorship attribution* yang telah dilakukan sebelumnya oleh (Burger, dkk. , 2011 dan Cavalcante, dkk. , 2014) didasarkan pada asumsi *bag-of-word*. Penelitian tersebut mengabaikan semantik kata-kata padahal faktor-faktor tersebut penting dalam mengidentifikasi karakteristik penulis, sehingga terdapat informasi yang hilang menurut Jiang, dkk. (2018).

Metode *deep learning* seperti Convolutional Neural Network (CNN) oleh (Sierra, dkk. , 2017), dan Recurrent Neural Network (RNN) oleh (Jiang, dkk. , 2018) telah diusulkan untuk memecahkan permasalahan terkait *authorship attribution*.

Multi-task learning memanfaatkan korelasi diantara keterkaitan *task* untuk meningkatkan akurasi klasifikasi dengan belajar *task* secara paralel (Ruder, 2017). Terdapat beberapa jaringan syaraf berbasis *Natural Language Processing* (NLP) oleh Liu, dkk. (2016); Jiang, dkk. (2018) memanfaatkan *multi-task learning* dan berhasil meningkatkan akurasi klasifikasinya. Dasar dari arsitektur *multi-task* dari model tersebut yaitu menggunakan layer-layer awal yang sama untuk menentukan fitur umum. Setelah *shared layer*, layer sisanya dipecah menjadi beberapa *task* spesifik.

Pada penelitian ini *multi-task learning* berbasiskan *deep learning* akan diterapkan untuk memecahkan permasalahan *authorship attribution* terhadap teks bahasa Indonesia, sehingga hasil prediksi lebih akurat.

1.2 Rumusan Masalah

Dengan merujuk pada latar belakang di atas, rumusan masalah untuk penelitian ini adalah bagaimana pengaruh penggunaan *multi-task learning* berbasiskan *deep learning* pada permasalahan *author attribution* pada situs berita Indonesia.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini, yaitu:

1. Dataset yang digunakan dalam bahasa Indonesia.
2. Metode multi-task learning yang digunakan adalah hard parameter sharing dengan deep learning.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan suatu metode *multi-task learning* berbasis *deep learning* untuk mengidentifikasi nama dan jenis kelamin dari penulis suatu teks berbahasa Indonesia.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan dapat digunakan untuk mengidentifikasi plagiarisme, penulis anonim, forensik digital, dan dapat digunakan menjadi referensi untuk penelitian-penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Miura, dkk. (2017) menggunakan Neural Attention Network untuk mengidentifikasi gender dan bahasa. Dataset yang digunakan ada dua, yang pertama yaitu PAN@CLEF 2017 yang terdiri dari 11.400 user Twitter dalam empat bahasa yang digunakan untuk author profiling training corpus, dan dataset yang kedua yaitu Streaming Tweets yang didapatkan dari Twitter Streaming APIs, dataset twitter ini digunakan untuk pre-train word embedding matrix dari model. Model yang digunakan dalam penelitian ini menggunakan word dan character embedding yang diproses dengan recurrent neural network (RNN) layer, convolutional neural network (CNN) layer, attention mechanism layer, max-pooling layer, dan fully-connected (FC) layer. Pemrosesan kata menggunakan RNN dan Attention layer, kemudian untuk pemrosesan karakter menggunakan CNN dan max-pooling layer. Selanjutnya, kedua representasi tweet digabungkan dengan menggunakan CNN. Dalam penelitian ini, character embedding menggunakan *fasttext*. Akurasi rata-rata untuk identifikasi gender yaitu 81.27%, sedangkan untuk identifikasi bahasa yaitu 89.82%.

Liu, dkk. (2016) dalam penelitiannya menggunakan Recurrent Neural Network dengan Multi-Task Learning untuk melakukan klasifikasi dalam teks. Dataset yang digunakan terdiri dari dataset, yaitu SST-1 movie review dataset dengan lima kelas (negatif, agak negatif, netral, agak positif, positif) dalam Stanford Sentiment Treebank, SST-2 movie review dengan kelas binary, SUBJ subjektivitas dataset untuk mengklasifikasi setiap instance termasuk subjektif atau objektif, dan IMDB dataset yang terdiri dari 100.000 movie review dalam kelas binary. Model yang digunakan dalam penelitian ini untuk task yang berbeda menggunakan LSTM layer dan embedding layer yang sama, representasi task-specific menggunakan output layer dengan fungsi softmax. Dalam penelitian ini, word embedding dilatih dengan menggunakan word2vec. Rata-rata peningkatan akurasi dalam 4 dataset yaitu sebesar 0.8%, dengan fine-tuning lebih lanjut, akurasi meningkat hingga 2.0%.

Dalam penelitian Jiang, dkk. (2018) menggunakan Multi-Task Learning dengan Hierarchical Feature untuk mengatasi permasalahan author profiling (usia, gender, dan klasifikasi pekerjaan). Penelitian ini menggunakan Blog dataset yang terdiri dari 678.161 blog post oleh 19.320 blogger. Setiap blog memuat gender, usia, dan pekerjaan. Penelitian ini menggunakan tiga representasi fitur, yaitu character-level,

word-level, dan topic-level. Untuk mendapatkan representasi character-level, penelitian ini menggunakan model CNN dalam tiap kata, untuk representasi word-level didapatkan dengan menggunakan LSTM, dan yang terakhir representasi fitur topic-level menggunakan LDA untuk memodelkan korpus. Kemudian, ketiga representasi fitur tersebut digabungkan dengan menggunakan simple Hadamard product. Hasil gabungan representasi fitur tersebut dilanjutkan ke softmax classifier tiap-tiap task untuk memprediksi profil penulis dari suatu dokumen. Hasil akhir dari penelitian ini mendapatkan peningkatan rata-rata akurasi sebesar 2% setelah menggunakan multi-task learning.

Mohsen, dkk. (2016) melakukan penelitian dengan menggunakan deep learning untuk mengidentifikasi penulis dari suatu teks. Dataset yang digunakan dalam penelitian ini merupakan subset dari *Reuters Corpus Volume* 1 (RCV1). Dalam dataset ini terdiri dari 50 penulis di mana masing-masing telah menulis 100 dokumen. Penelitian ini menggunakan fitur berupa karakter n-gram dan *frequent words* dan menggunakan normalisasi min-max. Kemudian fitur yang telah dipilih diseleksi dengan menggunakan *Chi Square*. Ekstraksi fitur dalam penelitian ini menggunakan Stacked Denoising AutoEncoder (SDAE), dan pada tahap klasifikasi menggunakan Support Vector Machine (SVM) dengan 10-*fold cross validation*. Metode yang diusulkan pada penelitian ini, mendapatkan akurasi hingga 95.12%.

Penelitian Tang, dkk. (2019) melakukan identifikasi penulis pada situs media sosial Weibo dengan menggunakan Wasserstein Generative Adversial Network. Dataset yang digunakan terdiri dari 100.000 tulisan oleh 100 pengguna aktif Weibo yang dipilih secara manual. Untuk setiap tulisan, maksimal terdiri dari 140 karakter dan minimal 38 karakter. Pada penelitian ini setiap tulisan dilakukan *embedding* dengan menggunakan *pre-trained* Global Vectors for Word Representation (GloVe) setelah ditokenisasi. Setelah proses embedding, data ditransformasikan ke LSTM dan didapatkan sebuah output vektor 64 dimensi. Akhirnya, ditransformasikan kembali ke 2 buah fungsi aktivasi ReLU yang berukuran 64 dan 32, dan didapatkan keluaran. Wasserstein GAN digunakan untuk menghindari ketidakseimbangan data dan meningkatkan performa klasifikasi sebesar 14% secara rata-rata. Akurasi yang didapatkan dalam penelitian ini yaitu 85%.

Dari kelima penelitian di atas, belum ada penelitian yang menggunakan *Multi-task learning* berbasis *Deep Learning* untuk menyelesaikan permasalahan *authorship attribution*. Perbandingan lebih lanjut kelima penelitian di atas disajikan dalam tabel berikut.

Tabel 2.1. Tinjauan Pustaka

Penelitian	Permasalahan	Dataset	Metode	Hasil
Miura, dkk. (2017)	Identifikasi gender dan bahasa pengguna dari situs media sosial twitter.	PAN@CLEF 2017 dan Twitter Streaming APIs.	*Character embedding* menggunakan CNN dan max-pooling, *word embedding* menggunakan RNN dan attention layer. Kedua representasi digabungkan dengan CNN.	Akurasi rata-rata untuk identifikasi gender yaitu 81.27%, sedangkan untuk identifikasi bahasa yaitu 89.82%.
Liu, dkk. (2016)	Klasifikasi teks.	SST1, SST2, SUBJ, dan IMDB.	Multi-task learning dengan LSTM dan embedding layer. Task spesifik output layer menggunakan softmax.	Rata-rata peningkatan akurasi dalam 4 dataset yaitu sebesar 0.8%, dengan fine-tuning lebih lanjut, akurasi meningkat hingga 2.0%.
Jiang, dkk. (2018)	*Author profiling* (Usia, gender, dan klasifikasi pekerjaan) dari suatu blog.	Blog post dari blogger.	Menggunakan tiga representasi fitur, yaitu character level dengan CNN, word level dengan LSTM, dan topic level dengan LDA. Ketiga representasi fitur digabungkan dengan simple Hadamard product. Kemudian, dilanjutkan ke softmax classifier untuk tiap task.	Hasil akhir dari penelitian ini mendapatkan peningkatan rata-rata akurasi sebesar 2% setelah menggunakan multi-task learning.
Mohsen, dkk. (2016)	Identifikasi penulis dari suatu teks.	RCV1.	Menggunakan SVM dengan fitur yang telah diekstrasi dengan menggunakan SDAE dari fitur berupa karakter n-gram dan *frequent words*.	Metode yang diusulkan pada penelitian ini, mendapatkan akurasi hingga 95.12%.

Tabel 2.1. Tinjauan Pustaka (lanjutan)

Penelitian	Permasalahan	Dataset	Metode	Hasil
Tang, dkk. (2019)	Identifikasi penulis dari situs media sosial Weibo.	Weibo.	Menggunakan *word embedding* dengan GloVe, kemudian ditransformasikan ke LSTM menjadi vektor 64 dimensi dan dilanjutkan ke 2 buah fungsi ReLu berukuran 64 dan 32.	Akurasi yang didapatkan dalam penelitian ini yaitu 85%.

BAB III
LANDASAN TEORI

3.1 *Word Embedding*

Word embedding adalah salah satu metode representasi kata di mana setiap kata direpresentasikan dalam sebuah vektor. Word embedding merupakan salah satu metode yang populer dalam *natural language processing* (NLP) yang bertujuan untuk mempelajari representasi vektor *low-dimensional* kata dari suatu dokumen Li, dkk. (2017).

Berbeda dengan representasi tradisional one-hot yang merepresentasikan suatu kata dengan sebuah vektor yang besar dan setiap kata independen dengan kata lainnya, sedangkan dalam *word embedding* dipresentasikan dalam vektor berukuran N dan setiap kata memiliki dependensi dengan kata lainnya.

Pada word embedding, nilai pada vektor tidak diberikan secara manual. Nilai pada vektor merupakan suatu parameter yang dilatih dengan metode jaringan syaraf dan biasanya sering dilatih dengan menggunakan *deep learning*.

3.1.1 Word2Vec

Word2Vec merupakan metode statistik untuk mempelajari *word embedding* dari teks korpus secara efisien.

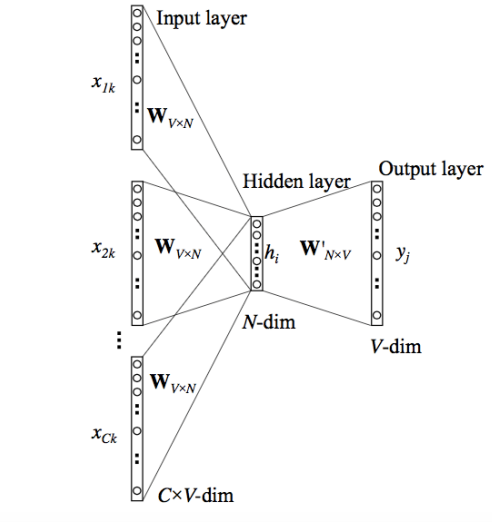
Dikembangkan oleh Mikolov, dkk. (2013) dengan tujuan untuk membuat *embedding* dengan pelatihan berbasiskan jaringan syaraf lebih efisien.

Dua model pembelajaran berbeda diperkenalkan yang dapat digunakan sebagai bagian dari pendekatan *word2vec* untuk mempelajari *word embedding* yaitu:

1. *Continuous Bag-of-Words* atau CBOW model.
2. *Continuous Skip-Gram* Model.

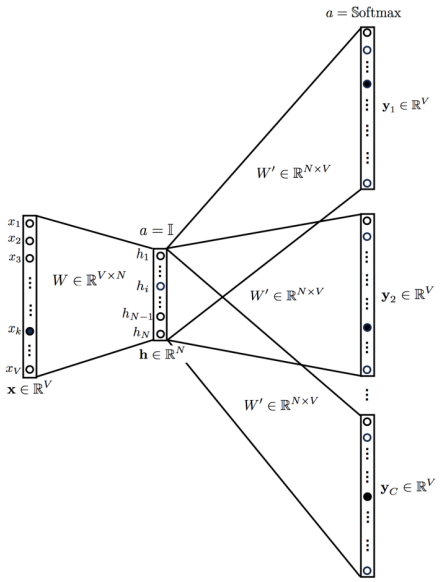
Model CBOW menerima konteks dari tiap kata sebagai masukan dan digunakan untuk memprediksi kata berdasarkan konteks yang diberikan. Arsitektur dari CBOW terdiri dari input layer yang merupakan representasi one-hot vektor berukuran V. *Hidden layer* terdiri dari N neuron dan output layer merupakan vektor berukuran V dengan elemennya merupakan nilai softmax. Neuron pada *hidden layer* merupak-

an salinan dari *weighted sum* masukan ke layer selanjutnya. Tidak terdapat fungsi aktivasi seperti sigmoid, tanh, maupun ReLU.



Gambar 3.1. Arsitektur CBOW

Model *continuous skip-gram* digunakan untuk memprediksi kata-kata sekitar dengan diberikan suatu kata mirip seperti kebalikan dari CBOW. Keluaran dari model berupa C distribusi probabilitas dari V kemungkinan.



Gambar 3.2. Arsitektur *Skip-gram*

Kedua model di atas menggunakan back-propagation untuk belajar. Menurut

Mikolov, dkk. (2013) *skip-gram* bekerja baik dengan data yang kecil dan dapat merepresentasikan kata-kata yang langka dengan baik.

Disisi lain, CBOW lebih cepat dan memiliki representasi yang lebih baik untuk kata yang lebih sering muncul.

3.1.2 GloVe

Global Vectors for Words Representation atau biasa disebut sebagai GloVe, merupakan algoritma pengembangan dari metode word2vec untuk mempelajari vektor kata secara efisien, dikembangkan oleh Pennington, dkk. (2014).

GloVe membangun model dengan menggunakan *word-to-word co-occurrence*. Dengan kata lain, jika 2 kata muncul bersama beberapa kali berarti kedua kata tersebut mempunyai kesamaan semantik.

3.1.3 FastText

FastText merupakan pengembangan dari *library word2vec* dengan informasi suku kata. Model ini membantu *embedding* memahami prefix dan suffix. *FastText* dapat memperoleh hasil yang sangat baik dalam representasi kata dan klasifikasi kalimat, secara khusus pada kasus kata tidak pernah dijumpai dengan menggunakan informasi karakter Bojanowski, dkk. (2017).

FastText memiliki keunggulan dibanding *word2vec*. Salah satunya yaitu kemampuan untuk menangani kata yang tidak pernah dijumpai sebelumnya. Misalnya kata "algoritmik" tetap akan diperoleh vektornya pada *fastText*. Metode *word2vec* akan menghasilkan *error* ketika menerima kata yang tidak pernah ada di dalam kamus.

3.2 *Deep Learning*

Deep Learning merupakan salah satu metode dalam *machine learning* berbasiskan jaringan syaraf tiruan dengan *feature learning*. Proses pembelajaran dapat dengan metode *supervised, semi-supervised*, atau *unsupervised* menurut Bengio, dkk. (2013); Schmidhuber (2015).

Kata "deep" biasanya merujuk pada jumlah *hidden layer* dalam jaringan syaraf. Jaringan syaraf tradisional biasanya hanya mempunyai 2-3 *hidden layer*, sedangkan *deep network* dapat mempunyai hingga ratusan *hidden layer*.

Pada *task supervised learning* model *deep learning* dilatih dengan menggunakan dataset yang sangat banyak dan arsitektur jaringan syaraf mempelajari fitur secara langsung dari data tanpa memerlukan ekstraksi fitur secara manual.

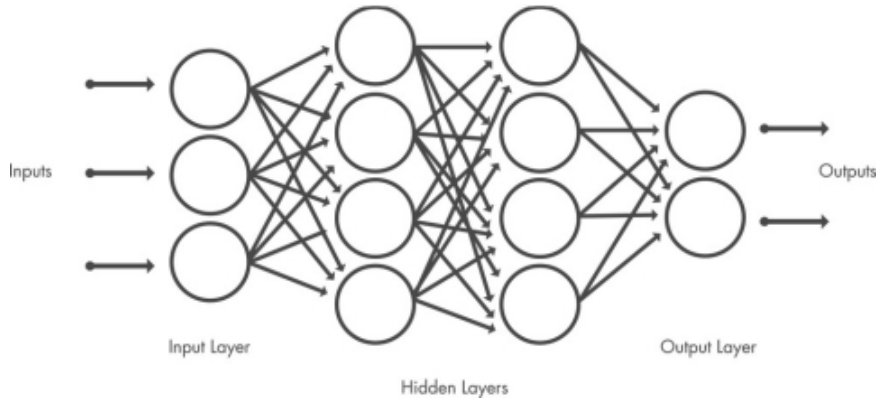
Beberapa jenis *deep learning* yang populer yaitu *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN).

3.2.1 *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan salah satu algoritma *deep learning*, biasanya diaplikasikan pada data citra visual. CNN dapat juga diaplikasikan dalam bidang NLP seperti dalam penelitian Kim (2014).

CNN adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *deep neural network* karena mempunyai jumlah *hidden layer* yang cukup banyak.

Mirip dengan jaringan syaraf lainnya, CNN terdiri dari *input layer*, *output layer*, dan banyak *hidden layer* di antaranya.



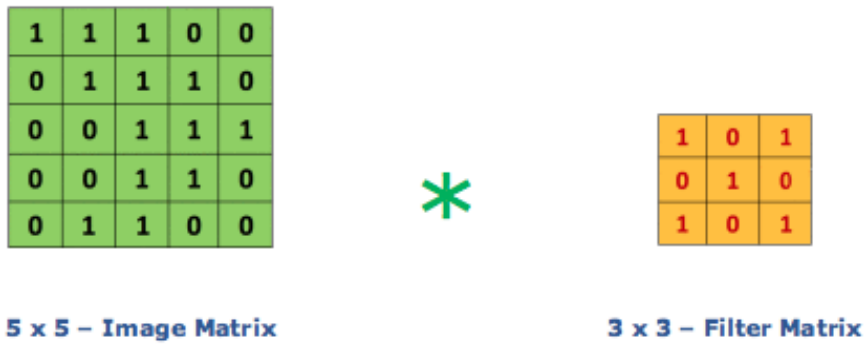
Gambar 3.3. Arsitektur CNN Sederhana

Layer-layer tersebut melakukan operasi yang mengubah parameter data dengan tujuan untuk mempelajari fitur spesifik dari data. Tiga dari layer yang paling umum pada CNN yaitu:

1. *Convolution Layer*

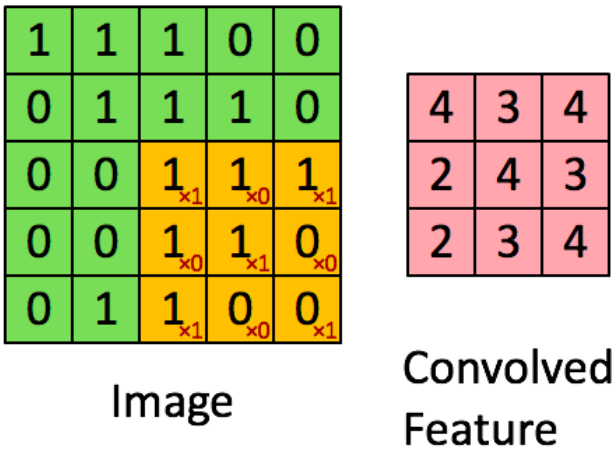
   Convolution merupakan layer pertama yang mengektraksi fitur dari masukan. Konvolusi adalah suatu istilah matematis yang menerima dua buah masukan yaitu sebuah matrix citra dan sebuah filter atau *kernel*.

Misalkan pada sebuah citra berukuran 5x5 yang nilai pikselnya adalah 0 atau 1, dan matriks filter 3x3 sebagai berikut:



Gambar 3.4. Matriks citra dikalikan dengan matriks filter

Kemudian, hasil konvolusi dari matriks citra berukuran 5x5 dikalikan dengan matriks filter berukuran 3x3 disebut dengan *feature map* seperti pada Gambar 3.5

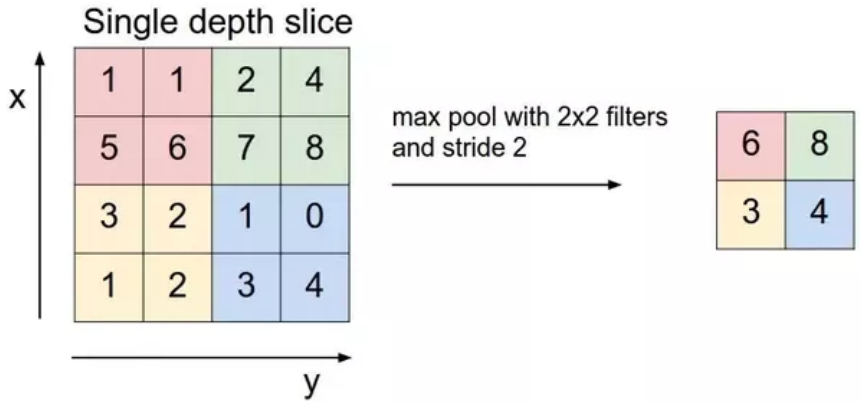


Gambar 3.5. 3x3 *Output* Matriks

2. *Pooling Layer*

Pooling layer akan mengurangi jumlah parameter ketika ukuran citra terlalu besar. Spasial *pooling* juga disebut sebagai *subsampling* atau *downsampling* mengurangi dimensionalitas dari tiap *map* tetapi tetap mempertahankan informasi penting. Dalam sebagian besar CNN, jenis *pooling layer* yang digunakan adalah *max pooling*. *Max pooling* membagi *feature map* menjadi beberapa *grid* kecil, lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun

matriks citra yang telah direduksi seperti yang telah ditunjukkan pada Gambar 3.6



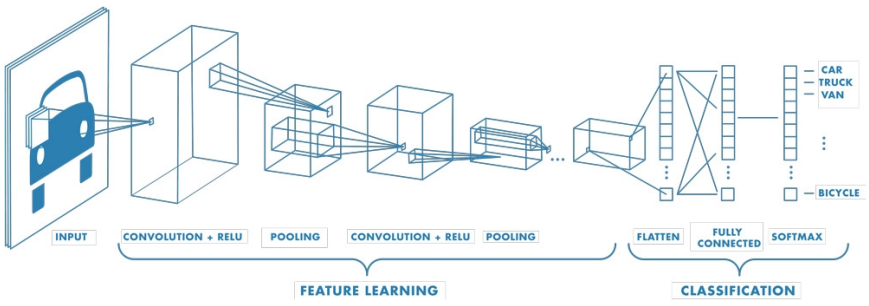
Gambar 3.6. ***Max Pooling***

3. Fungsi Aktivasi

   Fungsi aktivasi adalah fungsi *non linear* yang memungkinkan sebuah jaringan syaraf tiruan dapat men-tranformasi data masukan menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan *hyperlane* sederhana yang memungkinkan dilakukan klasifikasi. Dalam CNN fungsi aktivasi yang banyak digunakan yaitu *rectified linear unit* (ReLU). ReLU membuat proses pelatihan menjadi lebih cepat dan efisien dengan memetakan nilai negatif menjadi nol dan tidak mengubah nilai positif seperti pada persamaan 3.1.

$$f(x) = max(0, x) \tag{3.1}$$

Operasi tersebut diulangi sebanyak puluhan hingga ratusan layer, dengan tiap layer mempelajari untuk mengenali fitur yang berbeda.



Gambar 3.7. Contoh Arsitektur CNN

3.2.2 *Recurrent Neural Network*

Recurrent Neural Network (RNN) merupakan salah satu bentuk arsitektur jaringan syaraf tiruan di mana graf komputasi mempunyai *cycle* berarah. RNN dirancang khusus untuk memproses data yang bersambung/berurutan (*sequential data*). RNN biasanya digunakan untuk menyelesaikan tugas yang terkait dengan *time series*, misalnya prediksi saham.

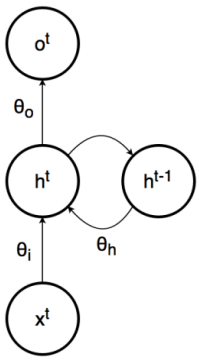
RNN mempunyai memori yang membuat model dapat menyimpan informasi sebelumnya. Hal ini yang memungkinkan RNN mengenali pola data dengan baik. Cara yang dilakukan RNN untuk dapat menyimpan informasi dari masa lalu adalah dengan melakukan *looping* di dalam arsitekturnya, yang secara otomatis membuat informasi dari masa lalu tetap tersimpan.

Ide di balik arsitektur RNN adalah bagaimana mengeksploitasi struktur data yang sekuensial. Nama RNN berasal dari fakta bahwa RNN beroperasi secara berulang. Hal ini berarti bahwa operasi yang sama dilakukan untuk setiap elemen dari suatu urutan, dengan *outputnya* bergantung pada *input* saat ini dan operasi sebelumnya. RNN berfokus pada sifat data di mana keadaan waktu saat sebelumnya atau saat ini (t) mempengaruhi keadaan waktu berikutnya (t+1).

Secara sederhana, persamaan 3.2 menjelaskan bagaimana RNN berevolusi setiap waktu:

$$\begin{aligned} o^t &= f(h^t; \theta) \\ h^t &= g(h^{t-1}, x^t; \theta), \end{aligned} \tag{3.2}$$

Di mana o^t merupakan keluaran RNN pada waktu ke-t, x^t merupakan masukan RNN pada waktu ke-t, dan h^t merupakan keadaan dari *hidden layer* pada waktu ke-t. Gambar 3.8 mengilustrasikan relasi di antara ketiga variabel dalam RNN.



Gambar 3.8. Model RNN

Persamaan pertama pada 3.2 menjelaskan bahwa, diberikan parameter θ (yang terdiri dari bobot dan bias dari jaringan), *output* pada waktu ke-t hanya bergantung pada keadaan *hidden layer* pada waktu ke-t. Persamaan kedua pada 3.2 menjelaskan bahwa, diberikan parameter θ yang sama, dan *hidden layer* pada waktu ke-t bergantung pada *hidden layer* pada waktu ke-$t-1$ dan masukan pada waktu ke t. Persamaan kedua menunjukkan bahwa RNN dapat mengingat informasi masa lalu dengan menggunakan h^{t-1} sebagai parameter perhitungan h^t saat ini.

Salah satu permasalahan RNN yang umum dijumpai karena fungsi transposisi adalah pada saat pelatihan komponen gradien vektor dapat mengembang atau mengecil secara eksponen dalam urutan yang panjang Hochreiter, dkk. (2001). Permasalahan *exploding* atau *vanishing gradient* membuat model RNN sulit untuk mempelajari korelasi jarak jauh dari suatu urutan.

Jaringan *Long short-term memory* (LSTM) diusulkan oleh Hochreiter dan Schmidhuber (1997) untuk mengatasi permasalahan *exploding/vanishing gradient*, sehingga RNN dapat digunakan pada data urutan yang sangat panjang.

3.3 *Multi-Task Learning*

Multi-Task Learning (MTL) merupakan salah satu sub bidang dari *machine learning* di mana beberapa pembelajaran *task* diselesaikan dalam waktu yang sama, dengan memanfaatkan kesamaan dan perbedaan diantara *task*. MTL dapat meningkatkan efisiensi dalam belajar dan akurasi prediksi untuk *task* spesifik ketika dibandingkan dengan melatih model secara terpisah Caruana. (1997).

Multi-task Learning telah sukses digunakan dalam berbagai aplikasi *machine learning* salah satunya dalam bidang *natural language processing* Liu, dkk. (2016); Jiang, dkk. (2018).

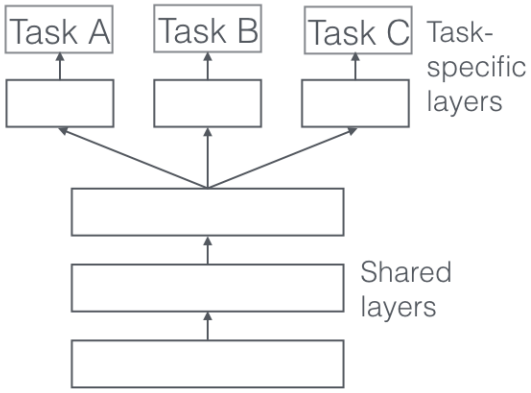
Pengaturan dalam MTL mirip dengan *transfer learning* Pan dan Yang (2010) tetapi juga mempunyai perbedaan yang signifikan. Dalam multi-task learning, tidak ada perbedaan diantara *task* yang berbeda dan tujuannya adalah untuk meningkatkan performa dari semua *task*. Namun, dalam *transfer learning* dengan tujuan untuk meningkatkan performa dari *task* target dengan bantuan dari *task* sumber, *task* target lebih berperan penting dibandingkan dengan *task* sumber. Oleh sebab itu, MTL memperlakukan semua *task* dengan sama sedangkan dalam *transfer learning task* target menarik lebih banyak perhatian dibandingkan semua *task*.

Terdapat dua cara yang paling umum untuk menggunakan *multi-task learn-*

ing dalam *deep neural network*. Dalam konteks *Deep Learning*, *multi-task learning* biasanya menggunakan antara *hard* atau *soft parameter sharing* dari *hidden layer*.

3.3.1 *Hard Parameter Sharing*

Hard parameter sharing merupakan pendekatan yang paling umum digunakan pada MTL Ruder (2017). Secara umum diaplikasikan dengan cara menggunakan *hidden layer* yang sama diantara semua *task*, dengan tetap menggunakan beberapa *task* spesifik *output layer*.

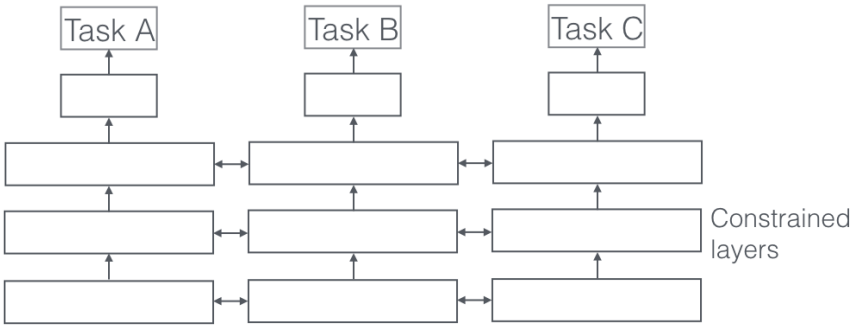


Gambar 3.9. Arsitektur *hard parameter sharing*

Hard parameter sharing sangat mengurangi risiko *overfitting*. Baxter (1997) menunjukkan bahwa risiko *overfitting* parameter bersama merupakan order N, di mana N merupakan jumlah *task*. Semakin banyak *task* yang dipelajari secara bersamaan, membuat model harus mencari representasi yang mencakup semua *task* dan semakin kecil pula kemungkinan *overfitting* pada *task* asli.

3.3.2 *Soft Parameter Sharing*

Dalam *soft parameter sharing* di lain sisi, untuk setiap *task* mempunyai model dengan parameter masing-masing. Jarak antara parameter pada model kemudian diregularisasi dengan tujuan untuk membuat parameter mirip. Duong, dkk. (2015) misalnya menggunakan ℓ_2 norm untuk regularisasi.



Gambar 3.10. Arsitektur *soft parameter sharing*

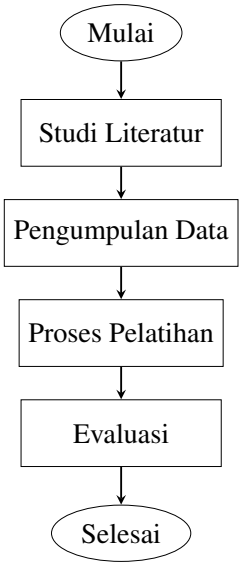
BAB IV
METODOLOGI PENELITIAN

4.1 Deskripsi Umum

Pada penelitian ini akan dikembangkan metode *multi-task learning* untuk memecahkan permasalahan *authorship attribution* pada kasus teks berbahasa Indonesia. Data yang akan digunakan bersumber dari situs berita Kompas dan Kumparan dan media sosial Twitter. *Word embedding* yang digunakan dalam penelitian ini adalah GloVe. Metode *multi-task learning* akan digunakan berbasiskan LSTM dengan tujuan untuk mengidentifikasi nama dan jenis kelamin penulis dari suatu teks berbahasa Indonesia.

4.2 Tahapan Penelitian

Penelitian dilakukan dalam beberapa tahapan. Tahapan-tahapan tersebut digambarkan dalam diagram alir pada Gambar 4.1 berikut.



Gambar 4.1. Diagram alir tahapan penelitian

4.3 Studi Literatur

Pada tahap pertama dalam penelitian ini dilakukan studi literatur. Studi literatur dilakukan dengan pencarian literatur yang berasal dari jurnal, *paper*, e-book, karya ilmiah, dan sebagainya yang terkait dengan topik penelitian.

4.4 Pengumpulan Data

Terdapat dua jenis data yang akan digunakan pada penelitian ini, yaitu tulisan pada situs berita dan tulisan pada media sosial Twitter. Dataset ini didapatkan dengan melakukan *scraping* menggunakan bahasa pemrograman *python*. Selanjutnya dijelaskan secara lebih rinci mengenai masing-masing dataset beserta karakteristiknya.

4.4.1 Situs Berita

Dataset situs berita yang digunakan berasal dari Kompas dan Kumparan. Terdapat 24.000 berita berbahasa Indonesia yang dibuat oleh 12 penulis. Dataset ini memiliki karakteristik menggunakan kata-kata baku dan untuk setiap berita mempunyai jumlah karakter yang cukup banyak mulai dari ratusan hingga puluhan ribu karakter.

4.4.2 Media Sosial

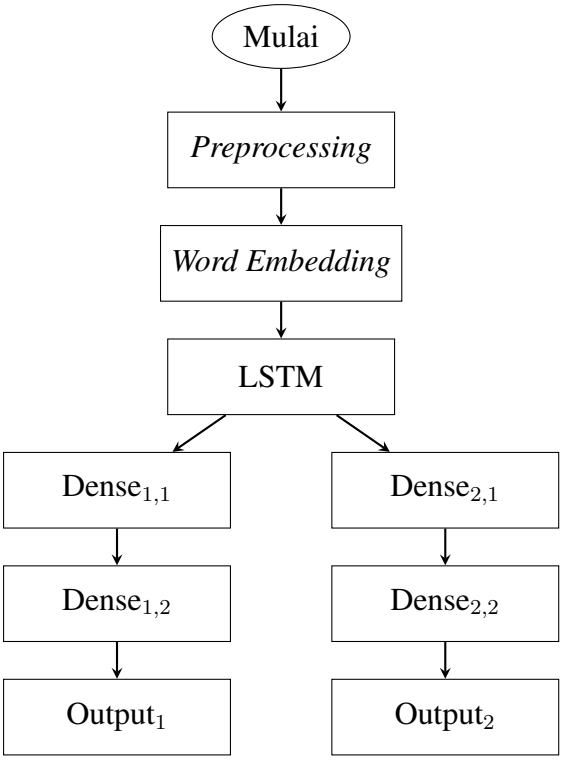
Dataset media sosial ini didapatkan dari Twitter. Dataset ini terdiri dari 20.000 *tweets* yang diambil dari 10 tokoh di Indonesia yang paling tidak mempunyai 10.000 *tweets*. Karakteristik pada dataset ini yaitu biasanya menggunakan kata-kata yang tidak baku dan maksimal terdiri dari 280 karakter.

4.5 Proses Pelatihan

Pada penelitian ini, proses pelatihan dilakukan dengan menggunakan variasi dari RNN yaitu LSTM. Sebelum diproses dalam LSTM, data teks diproses terlebih dahulu pada tahap preprocessing dan diubah representasinya menjadi vektor dengan menggunakan salah satu metode *word embedding* yaitu GloVe.

Arsitektur yang digunakan pada penelitian ini, menggunakan pendekatan *hard parameter sharing* pada *multi-task learning* yaitu dengan menggunakan layer LSTM dan *embedding layer* yang sama untuk *task* yang berbeda. Kemudian, pada layer

setelah LSTM bercabang menjadi dua *stack dense* layer. Terakhir, terdapat sebuah output layer. Arsitektur jaringan dapat dilihat pada Gambar 4.2 berikut.



Gambar 4.2. Arsitektur jaringan

4.5.1 *Preprocessing*

Preprocessing dilakukan dalam penelitian ini agar data bebas dari *noise* dan data dapat digunakan pada proses selanjutnya. *Preprocessing* yang digunakan dalam penelitian ini yaitu *data cleaning*, *lower casing*, dan tokenisasi.

Data cleaning yang dilakukan pada penelitian ini yaitu dengan menghapus angka dan tanda baca, sehingga teks hanya terdiri dari huruf alfabet. Kemudian, *lower casing* yaitu mengubah semua huruf menjadi huruf kecil. Terakhir, tokenisasi yaitu proses pemisahan teks menjadi potongan-potongan yang lebih kecil disebut sebagai token. Kata-kata, frasa, simbol, dan tanda baca dapat dianggap sebagai token.

Contoh *preprocessing* salah satu kutipan teks berita sebagai berikut:

- Teks: Seorang WNI yang berprofesi sebagai jurnalis, Veby Mega, tertembak peluru karet saat aksi demonstrasi di Hong Kong , Minggu (29/9).

- *Lower casing*: seorang wni yang berprofesi sebagai jurnalis, veby mega, tertembak peluru karet saat aksi demonstrasi di hong kong , minggu (29/9).
- *Data cleaning*: seorang wni yang berprofesi sebagai jurnalis veby mega tertembak peluru karet saat aksi demonstrasi di hong kong minggu
- Tokenisasi: ["seorang", "wni", "yang", "berprofesi", "sebagai", "jurnalis", "veby", "mega", "tertembak", "peluru", "karet", "saat", "aksi", "demonstrasi", "di", "hong", "kong", "minggu"]

4.5.2 *Word Embedding*

Terdapat tiga jenis *word embedding* yang dapat digunakan dalam merepresentasikan suatu kata menjadi vektor agar dapat diproses ke *deep learning*.

Dalam penelitian Tang, dkk. (2019), *word embedding* GloVe dengan dimensi 100 telah sukses digunakan untuk permasalahan *authorship identification*. Berdasarkan informasi tersebut dan karena GloVe dapat menangkap konteks informasi kata (seperti persamaan kata), penelitian ini menggunakan metode GloVe untuk *word embedding*.

4.5.3 LSTM

Model LSTM digunakan untuk tahapan ekstraksi fitur. Menurut Sneha dan Emilio (2018) LSTM dapat mengekstraksi fitur pada data sekuensial sehingga didapatkan konteks informasi secara efektif. Dengan memperhatikan kelebihan dalam mempelajari hubungan dalam data sekuensial, LSTM terbukti sangat efektif dalam berbagai permasalahan *Natural Language Processing*.

Pada penelitian Tang, dkk. (2019) digunakan *single layer* LSTM dengan *hidden layer* sebanyak 64 dimensi untuk meng-*encode* kalimat, algoritma Adam digunakan untuk mengoptimasi model dengan *learning rate* sebesar 1×10^{-4}, *batch size* sebesar 128, kemudian *layer dropout* dan regularisasi L2 juga digunakan. Berdasarkan informasi tersebut, penelitian ini mengadopsi parameter yang sama, namun dengan *batch size* sebesar 64 karena dataset yang digunakan lebih kecil dibandingkan dengan penelitian sebelumnya.

4.5.4 *Dense*

Layer ini terdiri dari 2 set *dense layer* yang masing-masing menggunakan fungsi aktivasi ReLU dengan ukuran 64 dan 32.

4.5.5 *Output*

Layer ini merupakan *task* spesifik output layer. Fungsi klasifikasi yang digunakan pada layer ini yaitu *softmax*. Output_1 digunakan untuk *task* identifikasi nama penulis sedangkan Output_2 digunakan untuk *task* identifikasi gender.

4.6 Evaluasi

Hasil dari proses pelatihan adalah model dengan parameter yang sudah dilatih menggunakan dataset yang dimiliki. Evaluasi nilai akurasi untuk tiap-tiap *task* dihitung dengan persamaan

$$\text{akurasi} = \frac{\text{banyak sampel yang diprediksi dengan benar}}{\text{banyak sampel data}} \tag{4.1}$$

Kemudian, untuk total akurasi gabungan dari kedua *task* dihitung dengan persamaan

$$\text{akurasi total} = \frac{\text{akurasi } task_1 + \text{akurasi } task_2}{2} \tag{4.2}$$

Di mana $task_1$ merupakan *task* untuk memprediksi nama *author*, dan $task_2$ merupakan *task* untuk memprediksi jenis kelamin dari *author*.

BAB V

JADWAL PENELITIAN

Jadwal penelitian disusun sebagai acuan dalam melaksanakan penelitian agar sesuai dengan target dan waktu yang telah ditentukan. Tahapan yang dilakukan terdiri dari studi literatur, pengumpulan data, implementasi, evaluasi, dan penulisan laporan. Tahapan-tahapan tersebut dilakukan dari bulan Januari hingga Mei 2021. Jadwal penelitian dapat dilihat pada tabel 5.1 berikut ini.

Tabel 5.1. Jadwal Penelitian

Kegiatan	2021																			
	Januari				Februari				Maret				April				Mei			
Minggu ke-	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literatur																				
Pengumpulan Data																				
Implementasi																				
Evaluasi																				
Penulisan Laporan																				

DAFTAR PUSTAKA

Abbasi, A. dan Chen, H., 2005, Applying authorship analysis to extremist-group web forum messages, *IEEE Inteligent Systems*, 20(5), 67-75.

Baxter, J., 1997, A Bayesian/Information Theoretic Model of Learning to Learn via Multiple Task Sampling, *Machine Learning*, 28, 7-39.

Begio, Y., Courville, A., dan Vincent, P., 2013, Representation Learning: A Review and New Perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828.

Bojanowski, P., Grave, E., Joulin, A., dan Mikolov, T., 2017, Enriching Word Vectors with Subword Information, `https://arxiv.org/pdf/1607.04606.pdf`, diakses 30 November 2020.

Burger, J., D., Henderson, J., Kim, G., dan Zarrela, G., 2011, Discriminating Gender on Twitter, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1301-1309.

Caruana, R., 1997, Multitask Learning, *Machine Learning*, 28, 41-75.

Cavalcante, T., Rocha, A., dan Carvalho, A., 2014, Large-Scale Micro-Blog Authorship Attribution: Beyond Simple Feature Engineering, *Iberoamerican Congress on Pattern Recognition*, 399-407.

Duong, L., Cohn, T., Bird, S., Cook, P., 2015, Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), 845-850.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., 2001, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, *IEEE Press*.

Hochreiter, S., Schmidhuber, J., 1997, LONG SHORT-TERM MEMORY, *Neural Computation*, 9(8), 1735-1780.

Holmes, D., I., 1994, *Authorship attribution*, Computers and the Humanities, 28, 87–106.

Jiang, Z., Yu, S., Qu, Q., Yang, M., Luo, J., dan Liu, J., 2018, Multi-task Learning for Author Profiling with Hierarchical Features, *International World Wide Web Conferences Steering Committee*, 55–56.

Juola, P., 2013, How a computer program helped reveal J. K. Rowling as author of A Cuckoo's Calling. *Scientific American*.

Kim, Y., 2014, Convolutional Neural Networks for Sentence Classification, https://arxiv.org/abs/1408.5882, diakses 30 November 2020.

Li, Y., Wang, S., Yang, T., dan Tang, J., 2017, Learning Word Representations for Sentiment Analysis, *Springer*, 26.

Liu, P., Qiu, P., dan Huang, X., 2016, Recurrent Neural Network for Text Classification with Multi-Task Learning, https://arxiv.org/abs/1605.05101, diakses 19 November 2020.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013, Distributed Representations of Words and Phrases and their Compositionality, https://arxiv.org/pdf/1310.4546.pdf, diakses 30 November 2020.

Miura, T., Taniguchi, T., Taniguchi, M., dan Ohkuma, T., 2017, Author Profiling with Word+Character Neural Attention Network Notebook for PAN at CLEF 2017, http://ceur-ws.org/Vol-1866/paper_90.pdf, diakses 20 November 2020.

Mohsen, A. M., El-Makky, N. M., dan Ghanem, N., 2016, Author Identification using Deep Learning, *2016 15th IEEE International Conference on Machine Learning and Applications*, 898-903.

Pan, S. J., Yang, Q., 2010, A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.

Pennington, J., Socher, R., Manning, C. D., 2014, GloVe: Global Vectors for Word Representation, https://nlp.stanford.edu/pubs/glove.pdf, diakses 30 November 2020.

Ruder, S., 2017, An Overview of Multi-Task Learning in Deep Neural Networks, https://arxiv.org/pdf/1706.05098.pdf, diakses 30 November 2020.

Rudman, J., The state of authorship attribution studies: Some problems and solutions. Computers and the Humanities, 31, 351-365.

Schmidhuber, J., 2015, Deep Learning in Neural Networks: An Overview, *Neural Network*, 61, 85-117.

Sierra, S., Montes-y-Gómez, M., Solorio, T., González, F. A., 2017, Convolutional Neural Networks for Author Profiling Notebook for PAN at CLEF 2017, http://ceur-ws.org/Vol-1866/paper_93.pdf, diakses 30 November 2020.

Sneha, K., Emilio, F., 2018, Deep neural networks for bot detection, *Information Sciences*, 467, 312-322.

Tang, W., Wu, C., Chen, X., Sun, Y., dan Li, C., 2019, Weibo Authorship Identification based on Wasserstein Generative Adversial Networks, *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, 1-5.

What Is Deep Learning?, https://www.mathworks.com/discovery/deep-learning.html, diakses pada 30 November 2020.