SENTIMENT EMBEDDINGS DOC2VEC PADA KLASIFIKASI KELUHAN POLUSI UDARA

Teguh Satriyo [1], Devi Tri Yuliani [1], Khirun Nisa [1]

Jurusan Sarjana Informatika
Fakultas Teknik dan Ilmu Komputer
Universitas Muhammadiyah Pekajangan Pekalongan
Jl. Raya Pahlawan No. Gejlig – Kajen Kab. Pekalongan Telp./Fax: (0285) 385313
e-mail: fatkhudin@gmail.com[1], ovieluo88@gmail.com[2]

Abstract

The community's need for healthy conditions and free from air pollution is the basis for the problems of the system under study. The system examines the sentiments of media users regarding the type of media that produces pollution smoke. The data used is Twitter social media data in which the GeoTag feature is already available to get precise locations where users send social media content. In general, documents with similar sentiments will be close together in the embeddings feature space so that they can be used to assess the performance of the sentiment analysis model. The system aims to benchmark the Sentiment Embeddings Sentiment Embeddings analysis works and Word2Vec analysis to get recommendations and early detection of pollution locations compared to other Deep Learning algorithms (Bert, LSTM and TextBloob).

Keywords: Pollution, twitter, sentiment embedding, word2Vec Design

Abstraksi

Kebutuhan masyarakat akan kondisi yang sehat dan bebas polusi udara menjadi dasar permasalahan dari sistem yang kaji. Sistem mengkaji sentimen pengguna media soal terkait jenis media penghasil asap polusi. Data yang digunakan adalah data media sosial twitter dimana didalamnya sudah tersedia fitur GeoTag untuk mendapatkan lokasi presisi dimana pengguna mengirimkan konten media sosial. Secara umum, dokumen dengan sentimen serupa, akan saling berdekatan di ruang fitur embeddings sehingga dapat digunakan untuk menilai kinerja model analisis sentimen. Sistem bertujuan untuk melakukan tolok ukur dari karya dan model analisis sentimen Sentiment Embeddings Analisis Word2Vec untuk mendapatkan rekomendasi dan deteksi dini lokasi polusi dibandingkan dengan algoritma Deep Learning yang lain (Bert, LSTM dan TextBloob).

Kata Kunci : Polusi, twitter, sentiment embedding, word2Vec.

1. Pendahuluan

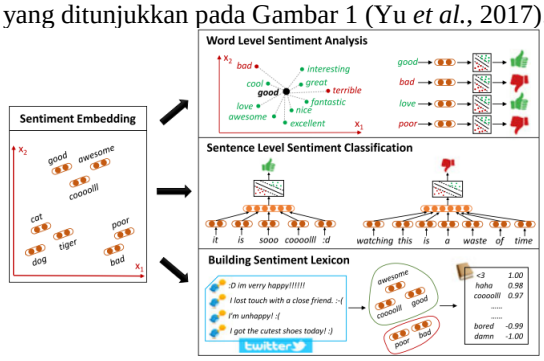
Era globalisasi sekarang ini, masyarakat sering bepergian dari satu tempat ke tempat lain, kebutuhan akan kondisi yang nyaman dari asap polusi menjadi permasalahan yang harus dipecahkan. Kondisi suatu wilayah terkait bencana sering berubah secara dinamis sehingga diperlukan sistem pelaporan yang sistematis, kontekstual dan *realtime* sehingga didapatkan kondisi wilayah yang cepat dan tepat. Media sosial dapat menjadi solusi dari sistem pelaporan yang dimaksud (Lei *et al.*, 2018)

Media sosial seperti Twitter, menyediakan pendekatan cepat untuk mengumpulkan informasi bersumber dari kerumunan dan cara menjangkau banyak anggota populasi sehingga dapat dimanfaatkan untuk mendukung upaya peringatan dan respons untuk lingkungan yang kurang sehat. Media sosial tersebut mampu memberikan lokasi metadata seperti kooordinat *latitude* dan *longitude* dimana pengguna mengirimkan konten media sosial (Stock, 2018). Salah satu konten media sosial yang sering dikirimkan adalah kejadian yang terjadi disekitar pengguna seperti polusi udara yang disebabkan asap baik kenalpot maupun cerobong asap. Populasi konten media sosial pada satu titik lokasi memberikan akurasi kejadian seperti epidemi penyakit menular dan polusi udara (Fan *et al.*, 2020) (Ghermandi and Sinclair, 2019). Media sosial dapat digunakan untuk mendeteksi secara dini perencanaan, peringatan, dan tanggap bencana bencana alam seperti tsunami, banjir dan tanah longsor dengan meninjau tantangan yang sedang berlangsung seperti mempertimbangkan masalah teknis, sosial dan kebijakan serta tantangan bagi sains dan tantangan praktis dalam menerapkan sistem (Landwehr *et al.*, 2016). Media sosial juga dapat diaplikasikan pada sistem untuk memprediksi dampak polusi pada lokasi tertentu (Zhang *et al.*, 2020).

Berdasarkan penelitian terdahulu, terdapat beberapa metode yang digunakan untuk ekstraksi fitur media sosial seperti penggunaan *decision Tree* sebagai penunjang keputusan rekomendasi tempat berdasarkan banyaknya kata per konten pada label tertentu sehingga dihasilkan akurasi dengan algoritma C4.5 sebesar 92% (Subowo, Rosyadi and Kusumawardhani, 2020). Penelitian lain mengenai akurasi data dengan metode Support Vector Machine (SVM) untuk menentukan tingkat kemacetan pada suatu lokasi menggunakan data twitter dihasilkan tingkat akurasi mencapai 97% (Subowo, Sediyono and Farikhin, 2019). Hasil

akurasi tersebut didapatkan pada labelling manual dengan dua kelas (+) dan (-), sehingga ketika diaplikasikan pada data *unigram* dengan *multilabel* didapatkan penurunan akurasi menjadi 74% pada C4.5 dan 83% pada SVM (Krouska, Troussas and Virvou, 2016). Metode terbaru untuk analsis data text dengan multilabel dengan *skip-gram* hubungan antar kata yang diperkenalkan oleh google yaitu *Word2Vec* (Church, 2017). Model *Word2Vec* dapat memproses data teks tidak terstruktur dengan mengambil corpus kata sebagai input dan menghasilkan vektor kata sebagai output. Salah satu kelebihan utama model *Word2Vec* adalah model ini merepresentasikan fitur sebagai vektor padat daripada representasi renggang konvensional yang umumnya mampu mengatasi masalah *sinonim* dan *homonim* yang sering dijumpai pada tugas NLP sehingga metode ini menghasilkan akurasi sebesar 89% (Nawangsari, Kusumaningrum and Wibowo, 2019). Penelitian – penelitian tersebut menggunakan permasalahan tunggal saja, seperti kemacetan saja atau rating hotel, sehingga pada beberapa permasalahan terkadang terdapat ambiguitas pada konteks kata, sehingga sistem rekomendasi yang menangani ketersebaran data memerlukan faktorisasi matriks yang diperluas Secara umum, dokumen dengan sentimen serupa, akan saling berdekatan di ruang fitur *embeddings* (Tang *et al.*, 2016).

Pada penelitian ini akan dikaji metode *skip-gram Word2Vec* dengan Fitur *Embeddings* pada data *multilabel* dengan objek kajian polusi udara sebagai *keyword* untuk diaplikasikan sebagai sistem peringatan dini. Keefektifan konteks kata dan memanfaatkan sentimen teks, *embedding* tidak hanya memiliki kemiripan semantik tetapi juga memiliki polaritas sentimen yang sama, sehingga dapat memisahkan sentimen ke ujung spektrum yang berlawanan serta konteks kata dengan fungsi *loss* khusus dengan memanfaat positif dan negatif seperti yang ditunjukkan pada Gambar 1 (Yu *et al.*, 2017).



Gambar. 1. Ilustrasi embeddings sentimen dengan aplikasi untuk tugas analisis *sentiment*, termasuk analisis sentimen tingkat kata, klasifikasi sentimen tingkat kalimat, dan membangun leksikon sentimen (Yu *et al.*, 2017).

1.1. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut

1. Bagaimana *preprocessing* pada *tweet* yang didalamnya adalah proses *crawling* (pengambilan data), *stemming, stop word removal, ekstraksi fitur*, dan *labelling*
2. Bagaimana hasil data *preprocessing* dapat diaplikasikan pada metode *Word2Vec* untuk manajemen keluhan polusi.
3. Bagaimana metode *Word2Vec* mampu memberikan klasifikasi keluhan polusi udara untuk dibandingkan akurasi sistemnya dengan metode Bert, LSTM dan *TextBloob*.

1.2. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat metode *skip-gram Word2Vec* dengan Fitur *Embeddings* pada data *multilabel* dengan objek kajian polusi udara sebagai *keyword* untuk diaplikasikan sebagai sistem peringatan dini. Keefektifan konteks kata dan memanfaatkan sentimen teks, e*mbedding* tidak hanya memiliki kemiripan semantik tetapi juga memiliki polaritas sentimen yang sama, sehingga dapat memisahkan sentimen ke ujung spektrum yang berlawanan serta konteks kata dengan fungsi *loss* khusus dengan memanfaat positif dan negatif.

1.3. Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan manfaat, baik dari segi teori maupun praktisi. Secara teori, penelitian ini diharapkan dapat digunakan untuk melengkapi kajian yang berkaitan dengan sistem manajemen keluhan polusi udara. Secara praktis,bermanfaat sebagai sarana untuk menambah pengetahuan mengenai sistem NLP dengan algoritma *Deep Learning*. Luaran dalam penelitian ini adalah Publikasi ilmiah pada jurnal terakreditasi nasional

BAB II. TINJAUAN PUSTAKA

1.1 2.1. Kajian Teori

2.1.1 Berita

Berita adalah kumpulan informasi yang mengandung fakta terhadap peristiwa yang sedang terjadi. Berita dapat diperoleh oleh siapapun melalui banyak cara, contohnya adalah media cetak seperti koran, majalah, buku, spanduk. Contoh lainnya adalah internet, acara televise dan perbincangan masyarakat. Topik yang umum diliput dalam berita adalah politik, pemerintahan, peperangan, edukasi, kesehatan.

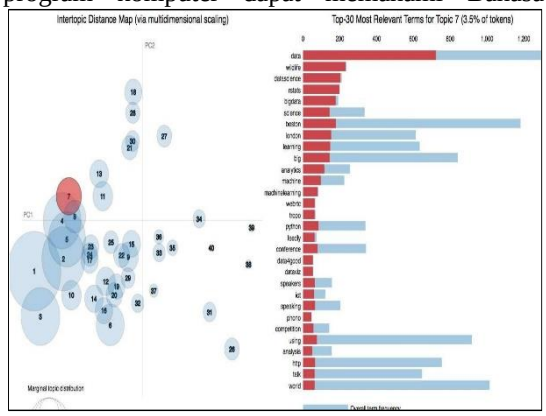
2.1.2 Artificial Intelligence

Artificial yang berarti buatan dan *Intelligence* yang berarti kecerdasan, *artificial intelligence* atau kecerdasan buatan adalah sebuah sistem buatan manusia yang memiliki kecerdasan sendiri, dimana sistem tersebut mampu belajar dari data yang diberikan sehingga dapat memprediksi secara akurat.

2.1.3 Natural Language Processing

Natural Language Processing atau Pemrosesan Bahasa Alami adalah sebuah cabang computer

science, artificial intelligence yang mempelajari interaksi antara komputer dengan bahasa manusia. Dengan adanya natural language processing, program komputer dapat memahami Bahasa



manusia sehingga program dapat membaca teks dan struktur linguistiknya. Contoh dari penggunaan Natural Language Processing pada umumnya adalah Chatbot yang digunakan pada *customer service*, *text classification* seperti *sentiment analysis*, *text-speech* atau *spell checking*.

2.1.4 Topic Modeling

Topic Modeling adalah tipe *statistical modeling* yang bertujuan untuk menemukan topik "abstrak" yang ada dalam sebuah dokumen. *Topic Modeling* biasanya digunakan dalam *text-mining* untuk mencari struktur semantic tersembunyi dari kumpulan teks. Sebuah dokumen umumnya dipandang oleh algoritma sebagai kumpulan topik dengan probabilitas semantik sendiri, dimana setiap topik memiliki hubungan antar kata. Contohnya adalah sebuah kalimat "Anjing Kucing Anjing Anjing" akan memiliki struktur semantik mengarah kepada kata "Anjing" dibandingkan dengan kata "Kucing". Melalui perhitungan matematika, algoritma akan dapat menentukan bahwa kalimat tersebut memiliki topik dengan struktur semantik yang mengarah ke kata "Anjing".

Topic Modeling juga disebut sebagai *Probabilistic Topic Models*, yang mengarah ke algoritma statistik untuk menemukan struktur semantik yang "*latent*", artinya struktur tersebut memang ada, namun belum ditemukan. *Topic Modeling* awalnya dibentuk sebagai alat untuk *text-mining*, namun dikembangkan sehingga dapat digunakan untuk mendeteksi struktur instruksif dari sebuah data seperti informasi genetik, gambar dan jaringan. (sumber: *https://en.wikipedia.org/wiki/Topic_model*).

Data yang akan dilakukan dalam *Topic Modeling* umumnya akan dilakukan *pre-processing* terlebih dahulu, umumnya disebut *Data Pre-Processing*. Tahapan-tahapan tersebut adalah:

1. *Tokenization*, data akan dipilah sehingga menjadi kumpulan kata.
2. *Stopwords Removal*, data akan dihilangkan kata stopword yang ada, yaitu kata yang tidak mengandung arti sama sekali.
3. *Lemmatizing*, kata yang mengandung sudut pandang orang ketiga atau *third-person view* akan diubah menjadi sudut pandang orang pertama atau *first-person view* dan kata dengan *past-tense* atau *future-tense* diubah menjadi *present-tense*.
4. *Stemming*, setiap kata akan diubah menjadi bentuk dasarnya.

Data Pre-Processing tersebut bertujuan agar pemrosesan data menjadi lebih mudah dan struktur semantik dokumen menjadi lebih mudah dicari. Algoritma yang umum digunakan dalam *Topic Modeling* adalah *Latent Dirichlet Allocation*, *Non-Negative Matrix Factorization*. Gambar 2.1 Contoh visualisasi *Topic Modeling* dengan menggunakan *Latent Dirichlet Allocation*, dimana algoritma mencari kata dengan probabilitas semantic tertinggi

2.1.5 Dataset

Menurut *Oxford Dictionaries*, *Dataset* adalah kumpulan informasi terkait yang terdiri dari elemen terpisah tetapi dapat dimanipulasi sebagai unit oleh komputer. *Dataset* adalah kumpulan dari data. Dataset mirip dengan sebuah tabel di dalam basis data atau sebuah statistik data matriks, dimana setiap kolom dalam tabel menggambarkan satu variabel. Kumpulan *dataset* yang begitu besar sehingga aplikasi pemrosesan data tradisional tidak memadai untuk mengatasinya dikenal sebagai *big-data*. *Training* set adalah data yang digunakan untuk membentuk sebuah model *classifier*. Model ini digunakan untuk memprediksi kelas data baru yang belum pernah ada.

Test set adalah Sampel data yang digunakan untuk memberikan evaluasi yang tidak bias dari model akhir yang sesuai pada training dataset. Membagi sebuah *dataset* menjadi *training* set dan *test* set Berguna agar model yang diperoleh nantinya mempunyai kemampuan generalisasi yang baik dalam melakukan generalisasi klasifikasi suatu data.

2.1.6 Web Scraping

Web Scraping adalah sebuah metode pengumpulan data melalui halaman internet dengan bahasa markup seperti *HTML* dan menganalisis data yang akan diambil dari halaman tersebut yang kemudian dikirim kembali ke pengguna. *Web Scraping* bekerja dengan menggunakan sebuah program kecil yang biasa disebut dengan istilah "*Scraper*" yang dibentuk dengan menggunakan bahasa pemrograman *Python* dengan menggunakan library *ScrapY* atau

BeautifulSoup. *Website* atau *domain* yang dituju oleh *Scraper* juga dapat dispesifikasi.

Scraper pertama – tama membaca halaman awal dari sebuah *website* atau *domain*, lalu membuka halaman pertama yang ditemukan. Kemudian data *HTML* dari halaman tersebut di-*parsing* sehingga *Scraper* mendapatkan data *HTML* yang diperlukan untuk penelitian yang kemudian disimpan kedalam sebuah variabel. Setelah *Scraper* mendapatkan data *HTML* dari halaman tersebut, *Scraper* akan mencari halaman baru yang dapat dibuka, dan kemudian proses yang sama akan diulangi sampai *Scraper* mendapatkan data dengan jumlah yang telah ditentukan. Kemudian variabel yang berisi data *HTML* tersebut disimpan kedalam sebuah *DataFrame* yang nantinya akan disimpan ke sebuah *file CSV*.

2.1.7 Machine Learning

Machine Learning adalah cabang dari ilmu *Artificial Intelligence* yang paling populer saat ini dikarenakan kegunaannya yang mampu menggantikan manusia dalam pekerjaannya. *Machine learning* adalah sebuah program dalam cabang *Artificial Intelligence* yang memiliki kemampuan untuk mempelajari sesuatu secara otomatis dan meningkatkan kemampuannya tanpa harus melakukan pemrograman ulang. *Machine learning* berfokus pada pengembangan dari program itu sendiri dan akses terhadap data yang dimiliki untuk dipelajari sehingga program akan terus berkembang.

Machine learning melakukan observasi terhadap sebuah data, mempelajari dan memahami pola dari data tersebut dan kemudian melakukan *decision making* berdasarkan pembelajaran yang telah dilakukan. Contoh umum dari *machine learning* adalah *data mining*, *image recognition*, *speech recognition*.

2.1.8 Decision Making

Decision Making adalah proses kognitif dari sebuah program *Artificial Intelligence* dalam menghasilkan sebuah *decision* atau hasil akhir. *Decision* didapatkan dari proses algoritma yang dirancang berdasarkan data *input* yang didapatkan, proses pembelajaran dari program dan aturan-aturan yang diberikan.

2.1.9 Clustering

Clustering merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan *cluster*. Objek yang di dalam *cluster* memiliki kemiripan karakteristik antar satu sama lainnya dan berbeda dengan *cluster* yang lain. Partisi tidak dilakukan secara manual melainkan dengan suatu algoritma *clustering*. Oleh karena itu, *clustering* dapat digunakan menemukan group atau kelompok yang tidak dikenal dalam data.

Clustering banyak digunakan dalam berbagai aplikasi seperti misalnya pada *business inteligence*, pengenalan pola citra, *web search*, bidang ilmu biologi, dan untuk keamanan (*security*). Dalam *business inteligence*, *clustering* bisa mengatur banyak pelanggan ke dalam banyaknya kelompok. Contohnya mengelompokan pelanggan ke dalam beberapa *cluster* dengan kesamaan karakteristik yang kuat. *Clustering* juga dikenal sebagai data segmentasi karena *clustering* mempartisi banyak *dataset* ke dalam banyak group berdasarkan kesamaannya. Selain itu *clustering* juga bisa digunakan sebagai *outlier detection*. Manfaat dilakukannya *clustering* adalah

1. *Clustering* merupakan metode segmentasi data yang sangat berguna dalam prediksi dan analisa masalah bisnis tertentu. Misalnya segmentasi pasar, *marketing* dan pemetaan zona wilayah.
2. Identifikasi objek dalam bidang berbagai bidang seperti *computer vision* dan *image processing*.

2.1.10 Spherical Clustering

Spherical Clustering adalah konsep *clustering* yang menggunakan *cosine similarity* sebagai dasar perhitungan jarak antar vektor (Jose, 2001).

Spherical Clustering menggunakan konsep vektor sebagai *centroid* atau nilai pusat dari setiap *cluster* yang telah dinormalisasi sehingga memiliki nilai *Euclidean* atau disebut *Concept Vectors*. *Spherical Clustering* memiliki beberapa keuntungan, salah satunya yaitu memanfaatkan *Data Sparsity* dari nilai vektor. Karakteristik dari *Spherical Clustering* ini memungkinkan *clustering* untuk metode penelitian ini.

Sebagaimana *Clustering* membutuhkan data *input* berupa nilai-nilai vektor, teks dan dokumen terlebih dahulu diubah menjadi nilai – nilai vektor. Setelah teks dan dokumen diubah menjadi nilai – nilai vektor, *Spherical Clustering* akan mengelompokkan setiap dokumen. Pada *Spherical Clustering*, setiap *node* yang memiliki sudut *cosine* yang berdekatan, cenderung memiliki nilai *similarity* atau kesamaan yang tinggi, mendekati angka 1. Begitu pula sebaliknya, apabila sudut *cosine* 2 buah *node* berjauhan, cenderung memiliki nilai similarity yang rendah atau mendekati 0.

2.1.11 Cosine Similarity

Cosine Similarity adalah sebuah metode pengukuran *similarity* antar dua buah vektor dengan mengukur nilai kosinus dari sudut derajat yang dibentuk oleh dua buah vektor tersebut dari titik pusat *spherical clustering*. *Cosine similarity* memiliki rentang nilai 0 sampai 1, dimana apabila nilai *cosine* mendekati angka 0, menunjukkan ketidakmiripan kedua data vektor. Sebaliknya apabila nilai *cosine* mendekati angka 1,

menunjukkan kemiripan antar dua buah vektor. *Cosine similarity* umum digunakan dalam analisa teks yang merupakan dimensi data yang tinggi. Hal ini dikarenakan sudut yang dibentuk antar 2 kata lebih baik daripada jarak *Euclidean* 2 buah kata.

2.1.12 High Dimensional Data

High Dimensional Data adalah data yang memiliki jumlah fitur yang banyak, contohnya adalah teks, gambar, informasi kesehatan seseorang. Dimensi data yang tinggi dapat menyebabkan *Curse of Dimensionality*, salah satunya yaitu *data sparsity*, data yang memiliki jumlah elemen *zero* (0) yang banyak. Dari *data sparsity*, proses ekstraksi informasi dari data menjadi lebih sulit. Umumnya jika menghadapi *data sparsity*, yang dilakukan adalah mengurangi jumlah dimensi data dengan menggunakan *Principal Component Analysis* (*PCA*). Namun pada penelitian ini, dengan menggunakan *metric cosine similarity* pada *spherical clustering*, permasalahan *data sparsity* dapat diatasi.

2.1.13 Curse of Dimensionality

Curse of Dimensionality merujuk ke berbagai permasalahan yang muncul dari menganalisa data dengan dimensi tinggi, contohnya adalah teks. Permasalahan yang umum ditemui adalah *Data Sparsity*, data dengan elemen *zero* (0) yang banyak. Contoh lainnya adalah rendahnya performa penggunaan *Euclidean Distance* sebagai metode pengukuran dalam dimensi data tinggi. Pada *machine learning*, dibutuhkan jumlah training data yang banyak untuk memastikan hasil training yang baik.

2.1.14 Data Sparsity

Data Sparsity adalah data yang memiliki jumlah elemen *zero* (0) yang tinggi. Data Sparsity cenderung merupakan hal buruk pada text-mining dikarenakan banyaknya elemen yang memiliki nilai zero (0) yang menyebabkan ekstraksi informasi menjadi sulit.

2.1.15 Labeled Data

Labeled Data adalah kumpulan sampel data yang telah diberi "*tag*" atau label yang memiliki informasi terhadap data tersebut. Label umumnya merupakan informasi yang menunjukkan kategori mana sebuah data berada. *Labeled Data* membantu program *machine learning* sehingga pembelajarannya menjadi lebih efektif dan *decision making* yang lebih akurat.

2.1.16 Word-Level Vector Embedding

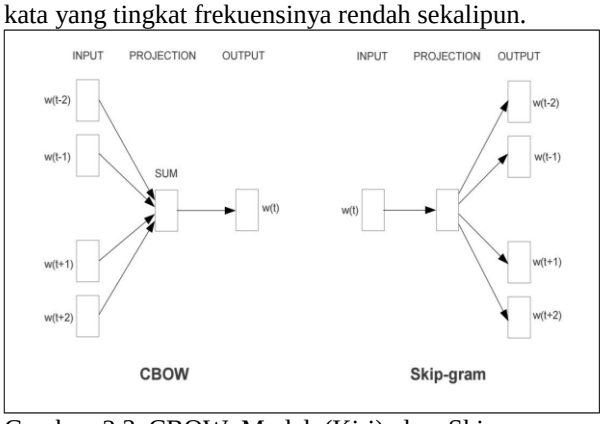
Dalam model *Natural Language Processing* saat ini, sering sekali menggunakan *Word-Level Vector Embedding*. Dalam *Word-Level Vector Embedding*, setiap kata diberikan sebuah nilai vektor yang mewakili kata tersebut. Nilai yang direpresentasikan dalam vektor kemudian dipelajari melalui *task* yang telah dioptimisasi.

Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) adalah model *neural network* pertama yang sukses menggunakan *Word-Level Vector Embedding*. Contoh lainnya adalah *FastText Embedding* (Bojanowski, Grave, Joulin, & Mikolov, 2017) dan *GloVe* (Pennington, Socher, & Manning, 2014).

2.1.17 Word2Vec

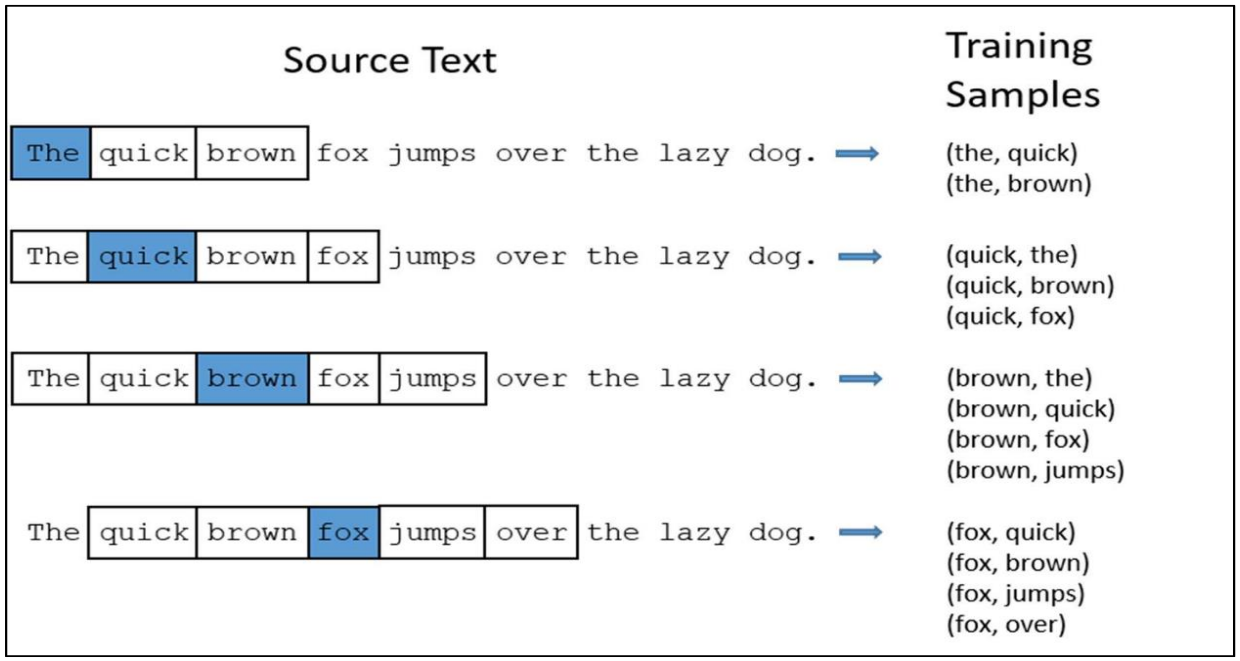
Word2Vec adalah sebuah *neural network* yang terdiri dari 2 *layer*, yang digunakan untuk mengubah setiap kata dalam *corpus* menjadi nilai – nilai vektor. Dibentuk oleh tim peneliti dari *Google* yang dipimpin oleh Tomáš Mikolov. Keuntungan dari menggunakan *Word2Vec* adalah arti semantik dari setiap kata unik dipreservasi, dimana tanpa menggunakan *Word2Vec*, kata hanya akan disimpan dengan *ID* unik tanpa arti dan memungkinkan terjadinya *Data Sparsity* atau banyaknya data kosong.

Word2Vec memiliki 2 metode pembelajaran, yaitu *Collection Bag of Words* (*CBOW*) dan *Skip-gram* Model. Awalnya *Word2Vec* melakukan inisialisasi *random* terhadap nilai vektor untuk setiap kata, yang kemudian diperbarui dengan menggunakan salah satu dari 2 metode pembelajaran. Dengan *CBOW* Model, algoritma mengambil 2 atau lebih kata dan algoritma akan memprediksi 1 kata yang sesuai dengan dokumen. Pada *Skip-gram* Model, algoritma mengambil hanya 1 kata dan akan memprediksi 2 kata atau lebih yang sesuai dengan dokumen. Hal ini menyebabkan seluruh kata mendapatkan pembelajaran yang sama, termasuk kata yang tingkat frekuensinya rendah sekalipun.



Gambar 2.3 CBOW Model (Kiri) dan Skip-gram Model (Kanan) Architecture (sumber: *Efficient Estimation of Word Representations in VectorSpace*, 2013, p5)

Gambar 2.4 Ilustrasi Metode Pembelajaran Word2Vec yaitu Skip-gram dan CBOW Model



(sumber: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/)

Pada Gambar 2.2, dapat dilihat ilustrasi dari metode pembelajaran *Word2Vec*. *Skip-gram* Model mengambil kata '*The*" dan kemudian akan memprediksi 2 kata selanjutnya yaitu kata "quick" dan kata "*brown*". Jika *Skip- gram* Model mengambil kata "*brown*", maka algoritma akan memprediksi kata "*The*", "*quick*", "*fox*" dan "*jump*".

Sebaliknya jika menggunakan *CBOW* Model, algoritma akan mengambil kata "*quick*" dan "*brown*" dan akan memprediksi kata "*the*". Atau contoh lainnya, adalah algoritma mengambil kata "*The*", "*quick*", "*fox*" dan "*jumps*" dan akan memprediksi kata "*brown*".

Pengembangan juga dilakukan untuk *Word2Vec*, contohnya adalah *Intelligent Word Embedding* (IWE) yang mengkombinasikan *semantic-dictionary mapping* dan *Word2Vec* untuk membentuk nilai vektor pada kata yang tidak ada dalam *vocabulary* untuk *Free-Text Radiology Report* (Banerjee et al, 2018). Contoh lainnya adalah *BioVectors* yang mengkarakterisasi *biological sequences* dalam interpretasi biokimia dan biofisik dari sebuah pola yang dapat digunakan untuk machine learning dalam bidang proteomik dan genomik (Asgari, & Mofrad, 2015).

Word2Vec juga diaplikasikan oleh peneliti lainnya dalam *sentiment analysis*, contohnya *Chinese Comment Sentiment Classification* (Zhang, Xu, Su, & Yu, 2015). Studi lainnya adalah *Sentiment Computing and Classification on Sina Weibo* (Xue, Fu, & Shaobin, 2014).

2.1.18 Document-Level Vector Embedding

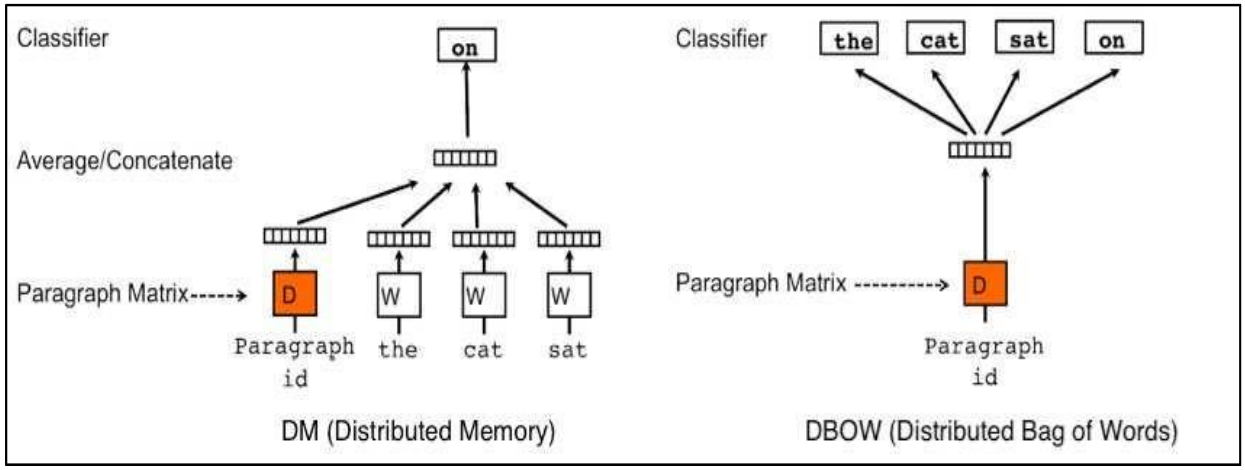
Pada tahun 2014, Le dan Mikolov mengembangkan *Word2Vec* sehingga dapat mempelajari dokumen juga, yang kemudian dikenal dengan nama *Doc2Vec*. Dengan model ini, dokumen dianggap sebagai salah satu dari keseluruhan nilai vektor kata yang ada. Dengan adanya *Document-Level Vector Embedding*, memungkinkan metode untuk memproses sebuah dokumen dan direlasikan dengan dokumen lainnya (Quoc Le, 2014).

2.1.19 Doc2Vec

Doc2Vec ((Quoc Le, 2014) adalah pengembangan dari algoritma *Word2Vec*, dibentuk oleh tim peneliti dari *Google* yang dipimpin oleh Tomáš Mikolov. *Doc2Vec* tidak hanya menjadikan setiap kata menjadi vektor, setiap paragraf yang ada juga diberi nilai vektor (Gambar 2.2). Nilai vektor dari suatu paragraph adalah unik, dan kata – kata yang sama yang muncul di dokumen berbeda memiliki nilai vektor yang sama.

Doc2Vec memiliki peran penting dalam metode penelitian ini, dikarenakan metode membandingkan setiap dokumen dengan dokumen lainnya. Berbeda dengan *Latent Dirichlet Allocation* (Campbell, Hindle and Stroulia, 2015)yang hanya membandingkan setiap topik dalam sebuah dokumen tanpa membandingkannya dengan dokumen lain. Sama seperti *Word2Vec*, *Doc2Vec* juga melakukan inisialiasi *random* terhadap nilai vektor dokumen dan kata.

Doc2Vec juga memiliki 2 metode pembelajaran, yaitu *Distributed Memory of Paragraph Vector* (*PV-DM*) dan *Distributed Bag of Words of Paragraph Vector* (*PV-DBOW*). *PV-DM* merupakan pengembangan dari *CBOW* Model pada *Word2Vec*, dimana algoritma juga menambahkan Paragraph ID pada pembelajarannya.



Gambar 2.5 PV-DM (Kiri) dan PV-DBOW (Kanan) yang merupakan metode pembelajaran Doc2Vec (Quoc Le, 2014)

Pada *PV-DBOW*, yang merupakan pengembangan dari *Skip-gram* model memiliki 2 versi, yaitu melatih vektor dokumen saja atau secara bersamaan melatih vektor kata juga. Apabila hanya melatih vektor dokumen saja, maka vektor kata tidak akan mendapatkan pembelajaran lebih lanjut sehingga akan tetap sama dengan nilai inisialisasi. Apabila secara bersamaan melatih vektor dokumen dan vektor kata, vektor kata juga akan mendapatkan pembelajaran lebih lanjut, namun hal ini membuat pembelajaran menjadi sangat lama.

Sama seperti *Word2Vec*, *Doc2Vec* juga digunakan oleh para peneliti untuk membentuk sebuah aplikasi, contohnya penilaian otomatis untuk video *interview* (Chen, Feng, Leong, & Lehman, 2016). Ada juga *sentiment analysis* untuk *informal text* pendek (Maslova, & Potapov, 2017) atau klasifikasi berita dari media *Twitter* dengan menggunakan *Doc2Vec* dan *Automatic Query Expansion* (Trieu, Tran, & Tran, 2017). Contoh lainnya adalah kategorisasi berita berbahasa Korea dengan menggunakan *CNN* dan *Doc2Vec* (Kim, & Koo, 2017).

2.2 *Related Works*

2.2.1 TaxoGen

TaxoGen (Zhang et al, 2018) adalah model *neural network* yang tujuannya adalah memprediksi topik hierarkis dari sebuah *corpus*, yang sedikit berbeda dengan model metodologi penelitian ini dimana tujuan dari model metodologi ini adalah memprediksi topik sebuah dokumen dari kumpulan dokumen.

Dalam pengaplikasiannya, *TaxoGen* menggunakan *Word2Vec* dengan seluruh *corpus* sebagai dasar analisis vektor. Dalam metode yang digunakan dalam penelitian, digunakan *Doc2Vec* dimana dokumen juga memiliki peran dalam analisis vektor.

2.2.2 Term Frequency – Inverse Document Frequency

Term Frequency – Inverse Document Frequency atau biasa disingkat *TF- IDF* adalah sebuah metode *text-mining* yang umum digunakan. Sesuai namanya,*TF-IDF* bekerja dengan mencari frekuensi setiap kata yang ada dan dibandingkan dengan jumlah dokumen yang mengandung kata tersebut. Formula perhitungan TF-IDF digambarkan pada Rumus 2.1

Weight(t) = tf(t in d)df(t) (2.1)

Dimana:

- *Weight* adalah nilai *weight* dari kata *t*
- *t* adalah sebuah kata, *tf* adalah *term frequency* dari kata *t*
- *d* adalah dokumen yang dianalisa, *idf* adalah *inverse document frequency* dari kata *t*

Nilai *term frequency* kata *t* didapat dari membagi jumlah kata *t* dalam sebuah dokumen dengan jumlah kata dari dokumen tersebut, digambarkan pada Rumus 2.2

tf(t in d) = (number of t appears)/(total words of d) (2.2) Dimana:

- *tf*(*t in d*) adalah nilai *term frequency* dari kata *t*
- *t* adalah sebuah kata
- *d* adalah dokumen yang dianalisa

Nilai *Inverse Document Frequency* didapat dari membagi jumlah dokumen yang mengandung kata *t* dengan jumlah dokumen yang ada dalam *corpus*, digambarkan pada Rumus 2.3 (2.3)

$$idf(t) = \log\left(\frac{D}{D(t)}\right)$$

Dimana:

- *idf(t)* adalah nilai *inverse document frequency* dari kata *t*
- *t* adalah sebuah kata
- *D* adalah jumlah keseluruhan dokumen
- *D(t)* adalah jumlah dokumen yang mengandung kata *t*

BAB III. METODE PENELITIAN

3.1. Alat dan Media Percobaan

Alat yang digunakan adalah seperangkat komputer yang terhubung dengan Google Collabs. Project dapat dilihat di link berikut : https://colab.research.google.com/drive/1CadAgkiLReD3Rrax9fGZ0aQGRW8ptZnf?usp=sharing

Sedangkan data dapat dilihat di link berikut : https://drive.google.com/file/d/1MLffkMOP0c84djbyjdUeKRiFOCjK5WzP/view?usp=sharing

Sistem dibangun dengan bahasa python dengan library numpy,pandas, sklearn dan mathplotlib pada platform KERAS.

Algoritma pembelajaran embedding kata yang mendominasi didasarkan pada hipotesis distribusi yang menyatakan bahwa representasi kata dapat direfleksikan oleh konteksnya. Cara efektif untuk menyandikan konteks kata menjadi representasi kata adalah "prediksi konteks". Pada kata target w_i dan kata konteksnya h_i, "Prediksi konteks" bertujuan untuk memprediksi w_i berdasarkan h_i, yang dapat dilihat sebagai pemodelan bahasa. Konteks kata target bisa jadi sebelum ceding, kata-kata berikut atau sekitarnya terjadi dalam sebuah teks.

$$h_i = \{w_{i-c}, w_{i-c+1}, \dots, w_{i-1}, w_{i+1}, w_{i+c}\} \quad (3.1)$$

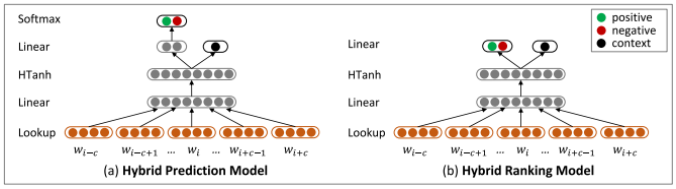
. Lapisan pencarian (juga disebut sebagai lapisan proyeksi) berisi tabel pencarian $LT \in R^{dx|V|}$ yang memetakan setiap kata ke vektor kontinu, di mana d adalah dimensi dari setiap vektor kata dan |V| adalah ukuran kosa kata. Operasi pencarian dapat dilihat sebagai fungsi proyeksi yang menggunakan file vektor biner idx saya yang nol di semua posisi kecuali di *i* index.

$$e_i = LT.idx_i \in R^{1xd} \quad (3.2)$$

Dimana e_i adalah embeddings kata konteks $\{w_{i-c}, w_{i-c+1}, \dots, w_{i-1}, w_{i+1}, w_{i+c}\}$ sebagai keluaran dari lapisan pencarian, yang diformalkan seperti di bawah ini

$$O_{lookup} = [e_{i-c}, \dots, e_{i-1}, e_{i+1}, \dots, e_{i+c}] \quad (3.3)$$

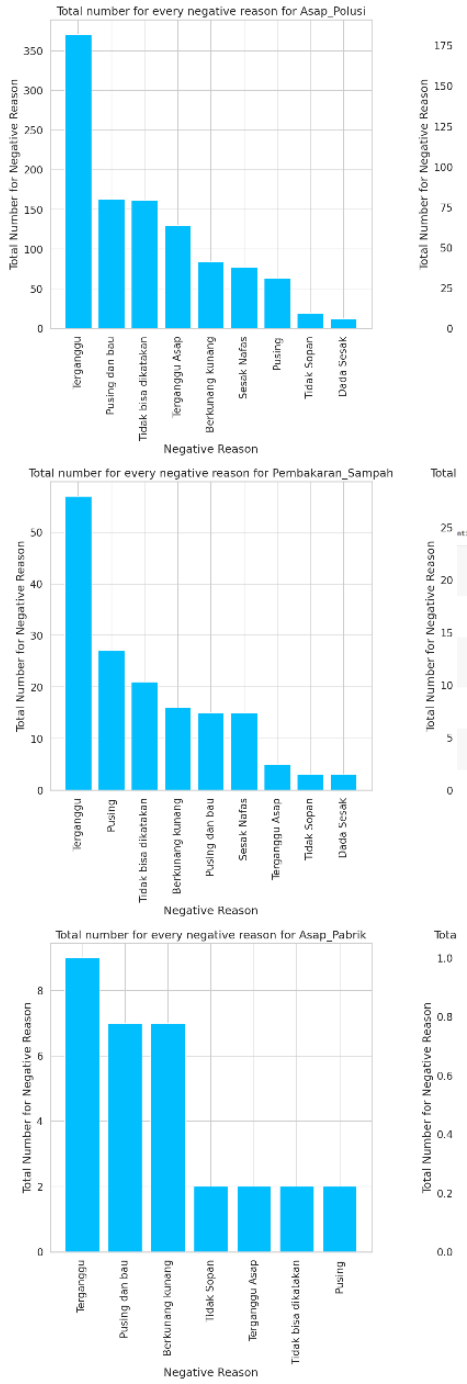
menggunakan pendekatan peringkat berpasangan untuk menangkap konteks kata untuk mempelajari embeddings kata. Ini memegang ide yang sama dengan estimasi kontrastif kebisingan tetapi tujuan pengoptimalannya adalah untuk menetapkan kata nyata pasangan konteks. Gambar 3.1 menunjukkan proses prediksi model.



Gambar 3.1. Ilustrasi model yang menangkap konteks kata serta sentimen kalimat

3.2. Data

Data yang digunakan adalah data twitter yang di crawl dari 1 Januari 2020 sampai 19 Desember 2020 sebanyak 3109 baris dengan 4 buah keyword, yaitu pembakaran sampah, asap kendaraan, asap polusi, dan asap rokok. Proses stemming dilakukan pada data Twitter yang didapatkan dengan memecah tiap kata dan menghilangkan simbol @, #, dan angka. Data hasil stemming tesebut kemudian dilakukan proses stop word removal untuk menghilangkan kata bantu, kata sambung sehingga dapat dilakukan ekstraksi fitur. Selanjutnya dilakukan pelabelan untuk menjabarkan kelas – kelas yang ada. Label yang digunakan adalah positif, netral, dan negatif. Dilakukan juga analisis fraksi sentimen untuk menunjukkan tingkat sentimen dari label yang diberikan. Gambar 3.2 menunjukkan ukuran sentimen dari tiap – tiap media penyumbang polusi terbesar.



Gambar 3.2. Ulasan negatif pengguna media sosial mengenai media penyumbang polusi

BAB IV. HASIL DAN PEMBAHASAN

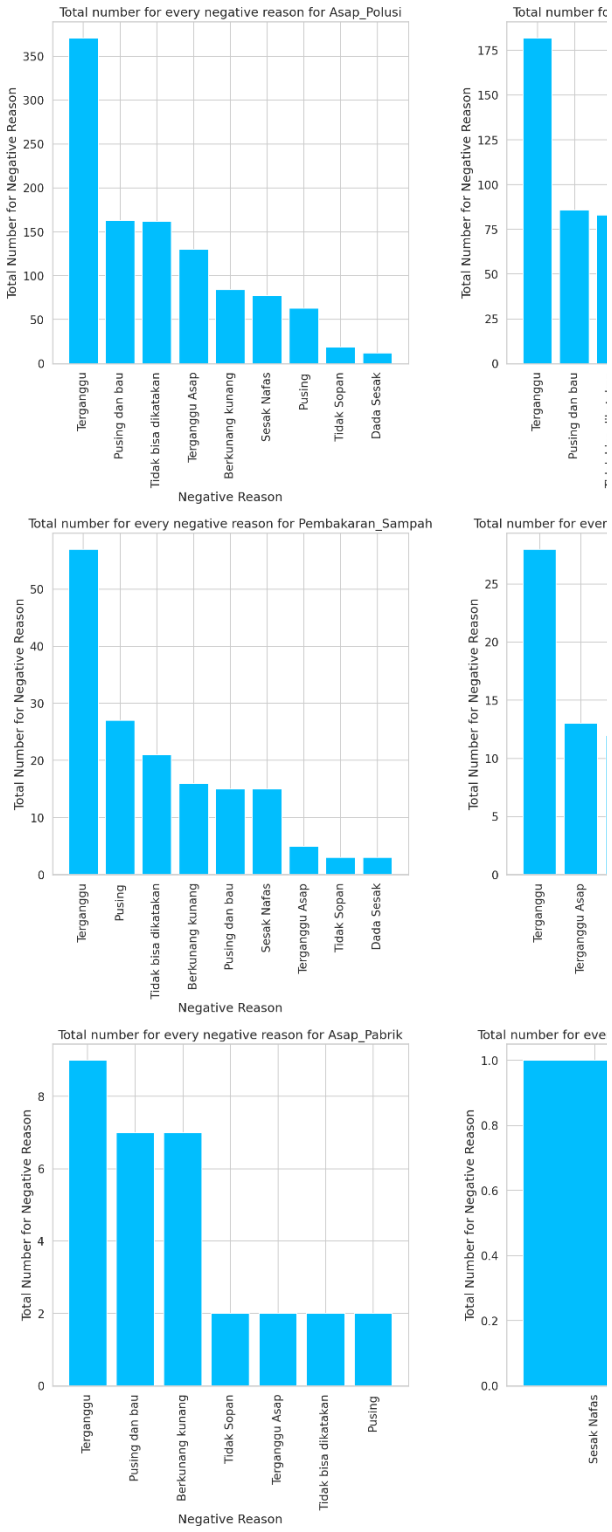
Data yang digunakan adalah data twitter yang di crawl dari 1 Januari 2020 sampai 19 Desember 2020 sebanyak 3109 baris dengan 4 buah keyword, yaitu pembakaran sampah, asap kendaraan, asap polusi, dan asap rokok. Langkah – langkah pre-processing dimulai dari stemming sampai labelling dapat dilihat pada Gambar 4.1.

Gambar 4.1 Text Pre-processing

Proses stemming dilakukan pada data Twitter yang didapatkan dengan memecah tiap kata dan menghilangkan simbol @, #, dan angka. Data hasil stemming tesebut kemudian dilakukan proses stop word removal untuk menghilangkan kata bantu, kata sambung sehingga dapat dilakukan ekstraksi fitur. Selanjutnya dilakukan pelabelan untuk menjabarkan kelas – kelas yang ada. Label yang digunakan adalah positif, netral, dan negatif. Dilakukan juga analisis fraksi sentimen untuk menunjukkan tingkat sentimen dari label yang diberikan. Gambar 5.2 menunjukkan struktur data yang digunakan.

Gambar 4.3 menunjukkan ukuran sentimen dari tiap – tiap media penyumbang polusi terbesar.

Gambar 4.2. Struktur Pelabelan pada Data.



Gambar 4.3. Ulasan negatif pengguna media sosial mengenai media penyumbang polusi.

Pada analisa awal dengan embedding sentiment didapatkan pengguna media sosial lebih menyukai menggunakan kata yang luas untuk menunjukkan sentimennya, pada Gambar 4.3, banyak pengguna media sosial menggunakan kata asap polusi dibandingkan dengan kata yang lebih spesifik seperti asap kendaraan besar, asap pabrik, maupun asap kendaraan bermotor dengan kelas mayoritas adalah yang menunjukkan sentimen "terganggu" dengan polusi udara. Sedangkan untuk koordinat pengguna media sosial, didapatkan hasil sebagai berikut yang menunjukkan contoh latitude dan longitude dari pengguna

[[106.812497, -6.324587], [106.863013, -6.324587], [106.863013, -6.262399], [106.812497, -6.262399]]

[[114.432276, -8.31779], [114.885827, -8.31779], [114.885827, -8.092484], [114.432276, -8.092484]]

[[112.246866, -7.607947], [112.307489, -7.607947], [112.307489, -7.538471], [112.246866, -7.538471]

Gambar 4.4. Kata sering muncul pada sentimen negatif

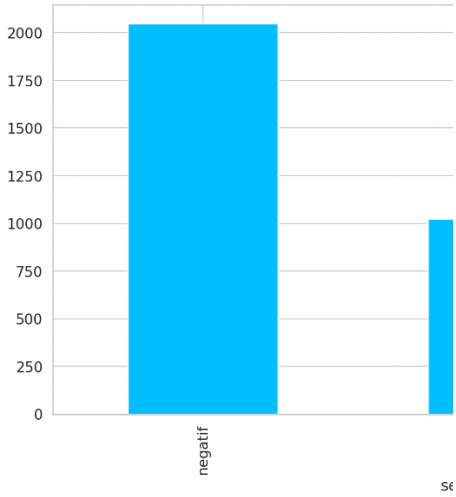
Dari Gambar 4.4, kata sentimen negatif paling banyak dimunculkan pada kata "polusi". Gambar 4.5 merupakan gambaran kata yang sering muncul pada kalimat sentimen netral terkait polusi udara.

Gambar 5.5. Kata sering muncul pada sentimen netral

Dari Gambar 4.5, kata sentimen netral paling banyak dimunculkan pada kata "rokok" dengan makna netral karena beberapa tweet memberikan sentimen yang tidak terganggu mengenai asap rokok. Gambar 4.6 merupakan gambaran kata yang sering muncul pada kalimat sentimen positif terkait polusi udara

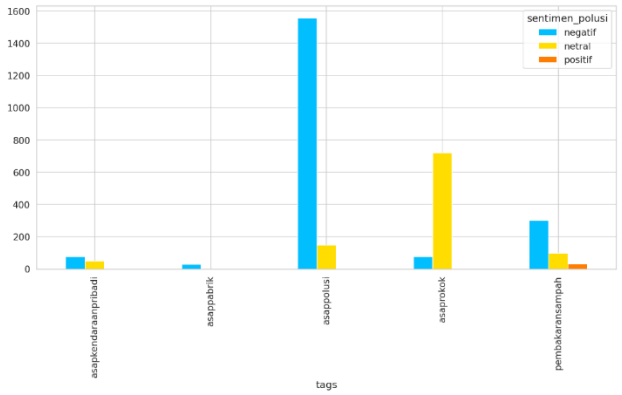
Gambar 4.6. Kata sering muncul pada sentimen positif

Dari Gambar 4.6, kata sentimen positif paling banyak dimunculkan pada kata “bakar sampah dan bahan bakar” hal ini dikarenakan adanya asumsi masyakat untuk membakar sampah agar lingkungan bersih dan penggunaan sampah sebagai bahan bakar misal bahan bakar energi sampah. Gambar 4.7 menunjukkan banyaknya data per sentimen. Dari Gambar 4.7 ditunjukkan bahwa sentimen negatif menghasilkan data yang paling banyak dari pada sentimen netral dan positif



Gambar 4.7 banyaknya data per sentimen

Gambar 4.8 menunjukkan tags sentimen paling banyak diulas oleh pengguna media sosial. Banyaknya pengguna media sosial yang mengulas kata “asap polusi” dengan sentimen negatif dengan rasio sentimen netral adalah 9,6% dan 0,01% untuk sentimen positif. sedangkan kata “pembakaran sampah” memiliki rasio sentimen netral lebih besar yaitu 45% dan 10% untuk sentimen positif.



Gambar 4.8 Tags sentimen paling banyak diulas oleh pengguna media sosial

Tujuan kami berikutnya adalah untuk memproses lebih lanjut data teks kami untuk NLP. Salah satu model yang akan kami kerjakan adalah BERT. BERT [12] adalah singkatan dari Representasi Bidirectional Encoder dari Transformers. Model Machine Learning tidak berfungsi dengan teks mentah. Anda perlu mengubah teks menjadi angka (atau semacamnya). BERT membutuhkan lebih banyak perhatian. Berikut adalah persyaratannya:

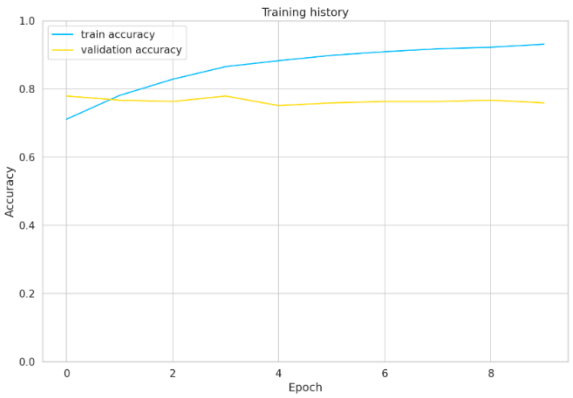
Tambahkan token khusus ke kalimat terpisah dan lakukan klasifikasi

Lewati urutan dengan panjang konstan (perkenalkan padding)

Buat array 0s (pad token) dan 1s (token nyata) yang disebut attention mask

Kita dapat menggunakan BERT versi cased dan uncased dan tokenizer. Secara intuitif, versi cased akan bekerja lebih baik, karena "BURUK" mungkin menyampaikan lebih banyak sentimen daripada "buruk". Berbeda dari pemrosesan teks klasik, yang akan kami lakukan pada data teks untuk model lain nanti, kami akan menggunakan BertTokenizer terlatih:

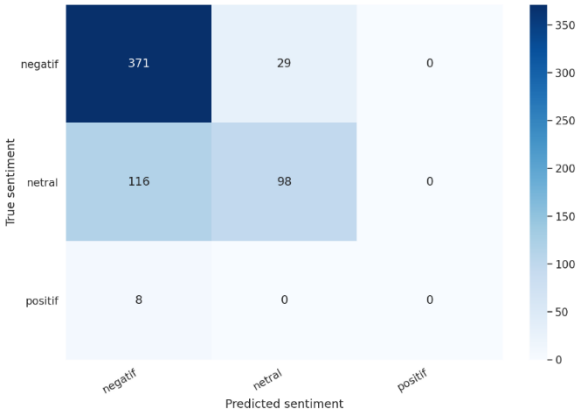
BERT bekerja dengan urutan panjang tetap. Kami menggunakan strategi sederhana untuk memilih panjang maksimal, sehingga kami mengisi setiap token hingga mencapai panjang maksimal. Gambar 4.9 menunjukkan hasil akurasi



Gambar 4.9. Akurasi pada Model Bert

Dari Gambar 10, nilai akurasi menunjukkan kenaikan akurasi berdasarkan nilai epoch, Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Netwok sampai dikembalikan ke awal untuk sekali putaran, karena satu Epoch terlalu besar untuk dimasukkan (feeding) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (batches). Gambar 4.9 menunjukkan Confusion matrik pada model Bert

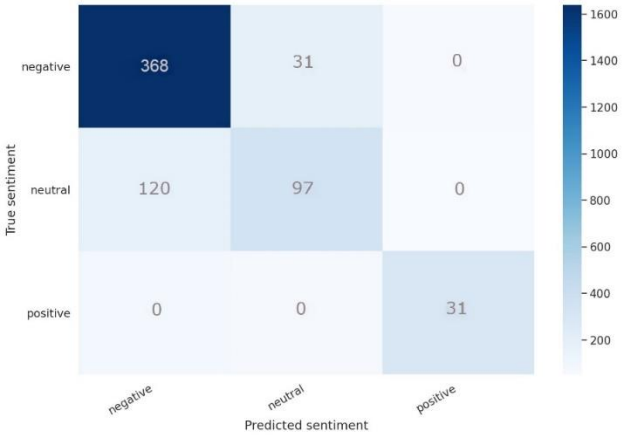
	Precision	recal	f1-score	support
Negatif	0.75	0.93	0.83	400
Netral	0.77	0.46	0.57	214
Positif	0.00	0.00	0.00	8
Accuracy			0.75	622
macro avg	0.51	0.46	0.47	622
weighted avg	0.75	0.75	0.73	



Gambar 4.10. Confusion Matrik pada Model Bert

Dari Gambar 4.10, dapat disimpulkan model Bert masih memberikan hasil yang tidak tepat pada ulasan netral yang diprediksi sebagai ulasan negatif sehingga menghasilkan akurasi maksimal tidak lebih dari 75%. Proses pelabelan dan pembobotan label pada tiap kalimat sentimen berpengaruh pada akurasi model Bert karena pada pengetesan yang ke-2 dimana dilakukan evaluasi pembobotan dengan Multi-label yang dilaporkan oleh [13], terdapat peningkatan akurasi sebesar 1%.

Model lain yang digunakan adalah LSTM. Jaringan LSTM (LSTM network) juga terdiri dari modul-modul dengan pemrosesan berulang. Konteks adalah sebuah vektor, yang jumlah elemennya kita tentukan sebagai desainer jaringan LSTM. Intuisinya adalah, masing-masing elemen kita harapkan bisa merekam suatu fitur dari input, misalnya dalam pemrosesan bahasa alami untuk bahasa Inggris, suatu elemen merekam gender dari subjek, elemen lain merekam apakah subjek tunggal atau jamak, dsb. Fitur-fitur ini akan ditemukan sendiri oleh LSTM dalam proses latihan. Gambar 4.11 menunjukkan Confusion matrik model dari LSTM.



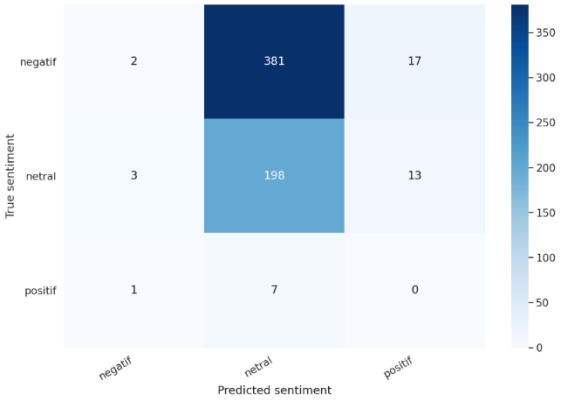
Gambar 4.11 Confusion matrik model dari LSTM.

	precision	recall	f1-score	support
negative	0.86	0.87	0.86	400
neutral	0.57	0.61	0.59	214
positive	0.72	0.64	0.68	8
accuracy			0.77	622
macro avg	0.72	0.70	0.71	622
weighted avg	0.78	0.78	0.78	622

Model lain yang digunakan adalah perpustakaan untuk analisis sentimen, TextBlob karena digunakan secara luas di industri. Pada definisi fungsi untuk prediksi, parameter ambang yang dipilih menggunakan rekomendasi web dari pengguna perpustakaan. Karena prediksi sentimen dibuat dalam kalimat, mean dari total skor sentimen diambil untuk ditinjau. Gambar 4.12 menunjukkan confusion matrik model TextBlob yang menunjukkan banyaknya kesalahan pada prediksi sentimen netral pada data sentimen negatif.

	precision	recall	f1-score	support
negatif	0.33	0.01	0.01	400
netral	0.34	0.93	0.49	214
positif	0.00	0.00	0.00	8
accuracy			0.32	622

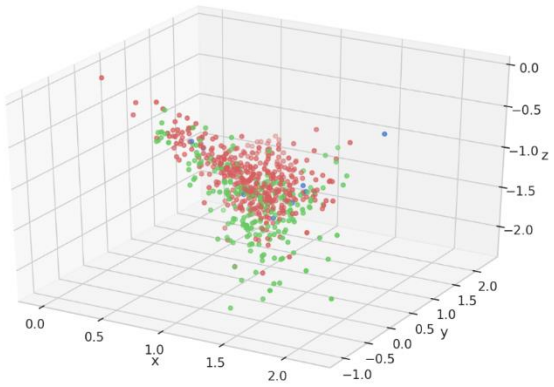
macro avg	0.22	0.31	0.17	622
weighted avg	0.33	0.32	0.18	622



Gambar 4.12 Confusion matrik model TextBlob.

Pada penelitian ini menggunakan kamus default leksikon SentiStrength yang diterjemahkan dan disesuaikan dengan aturan bahasa Indonesia, akan tetapi terms di kamus default belum sepenuhnya sesuai dengan kebutuhan kosakata bahasa Indonesia. Sehingga nilai akurasi textBlob cenderung rendah.

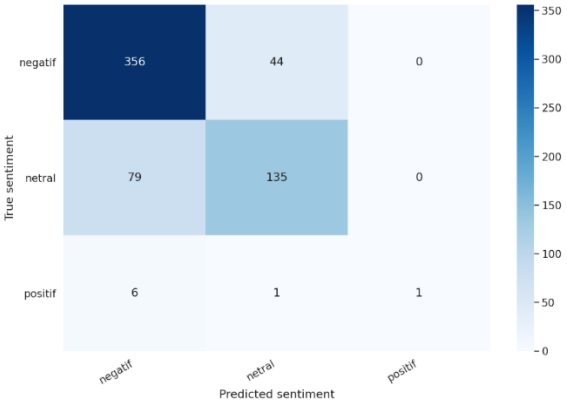
Di bagian ini, kita akan memiliki fokus besar untuk merepresentasikan ulasan kita dengan vektor fitur, menggunakan Doc2vec. Doc2vec adalah pendekatan tanpa pengawasan yang dibangun di atas word2vec, tidak seperti word2vec yang dapat membuat representasi vektor dari kata-kata sambil mempertimbangkan konteks akun, doc2vec dapat membuat representasi vektor dari dokumen. Tujuan dari langkah selanjutnya ini adalah untuk memvisualisasikan semua dokumen pengujian kami dalam ruang fitur dokumen, yang disorot oleh sentimennya. Karena model doc2vec hanya berfungsi dalam data pelatihan, sehingg dapat menguji seberapa baik model itu digeneralisasikan. Seperti yang terlihat lebih jauh di bawah ini, kita dapat melihat bahwa tiga kelas sentimen sangat terpisah dalam ruang fitur dokumen pengujian, yang membuktikan bahwa model doc2vec kami dapat menggeneralisasi pada data yang tidak terlihat seperti yang ditunjukkan pada Gambar 4.13



Gambar 4.13. Model doc2vec pada kata-kata yang didistribusikan (PV-DBOW) dengan tiga fitur vektor output untuk mengkodekan informasi dokumen.

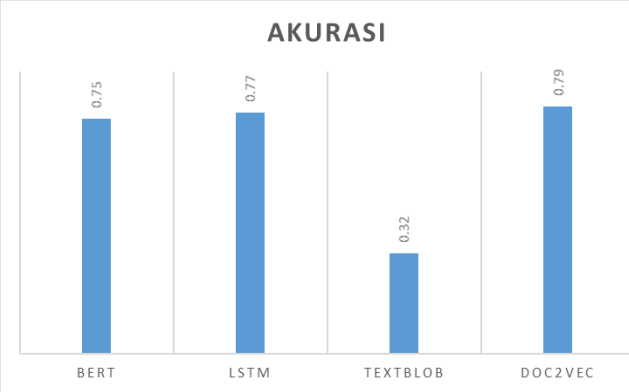
	precision	recall	f1-score	support
negatif	0.81	0.89	0.85	400
netral	0.75	0.63	0.69	214
positif	1.00	0.12	0.22	8
accuracy			0.79	622
macro avg	0.85	0.55	0.58	622
weighted avg	0.79	0.79	0.78	622

dari data yang disajikan, akurasi untuk sentimen embedding adalah 89% untuk sentimen negatif, mengetahui bahwa itu menjadi kelas utama dalam dataset kami dengan kemunculan terbanyak. Sepertinya sangat sulit untuk mengklasifikasikan ulasan positif dengan akurasi 12% karena sedikitnya data ulasan positif pada asap polusi. Untuk akurasi total pada Doc2Vec adalah 79%. Gambar 15 menunjukkan hasil confusion matric pada Model Doc2Vec



Gambar 4.15 Hasil confusion matric pada Model Doc2Vec

Dari data akurasi yang didapatkan dapat dibandingkan hasil akurasi dari model yang digunakan seperti yang ditunjukkan pada Gambar 4.16.



Gambar 4.16. Perbedaan Tingkat Akurasi Model

BAB IV. KESIMPULAN

1. Kebutuhan masyarakat akan kondisi yang sehat dan bebas polusi udara menjadi dasar permasalahan dari sistem yang kaji. Sistem mengkaji sentimen pengguna media soal terkait jenis media penghasil asap polusi. Data yang digunakan adalah data media sosial twitter dimana didalamnya sudah tersedia fitur GeoTag untuk mendapatkan lokasi presisi dimana pengguna mengirimkan konten media sosial. Proses pelabelan dan pembobotan label pada tiap kalimat sentimen berpengaruh pada akurasi Secara umum, dokumen dengan sentimen serupa, akan saling berdekatan di ruang fitur embeddings sehingga dapat digunakan untuk menilai kinerja model analisis sentimen. sistem bertujuan untuk melakukan tolok ukur dari karya dan model analisis sentimen Sentiment Embeddings Analisis Word2Vec menghasilkan tingkat akurasi lebih tinggi dari pada LSTM, Bert, dan TextBlob.

DAFTAR PUSTAKA

P. Lei, G. Marfia, G. Pau, and R. Tse, “Can we monitor the natural environment analyzing online social network posts? A literature review,” Online Soc. Networks Media, vol. 5, pp. 51–60, 2018, doi: 10.1016/j.osnem.2017.12.001.

[2] C. Fan, M. Esparza, J. Dargin, F. Wu, B. Oztekin, and A. Mostafavi, “Spatial biases in crowdsourced data: Social media content attention concentrates on populous areas in disasters,” Comput. Environ. Urban Syst., vol. 83, no. May, p. 101514, 2020, doi: 10.1016/j.compenvurbsys.2020.101514.

[3] A. Ghermandi and M. Sinclair, “Passive crowdsourcing of social media in environmental research: A systematic map,” Glob. Environ. Chang., vol. 55, no. January, pp. 36–47, 2019, doi: 10.1016/j.gloenvcha.2019.02.003.

[4] Y. Zhang, P. Siriaraya, Y. Kawai, and A. Jatowt, “Predicting time and location of future crimes with recommendation methods,” Knowledge-Based Syst., vol. 210, p. 106503, 2020, doi: 10.1016/j.knosys.2020.106503.

[5] E. Subowo, I. Rosyadi, and H. H. Kusumawardhani, “Twitter Data as Decision Tree Parameter for Analysis of Tourism Potential Policies,” vol. 436, pp. 474–478, 2020, doi: 10.2991/assehr.k.200529.099.

[6] E. Subowo, E. Sediyono, and Farikhin, “Ant Colony Algorithm for Determining Dynamic Travel Routes Based on Traffic Information from Twitter,” E3S Web Conf., vol. 125, no. 201 9, 2019, doi: 10.1051/e3sconf/201912523012.

[7] A. Krouska, C. Troussas, and M. Virvou, “The effect of preprocessing techniques on Twitter sentiment analysis,” IISA 2016 - 7th Int. Conf. Information, Intell. Syst. Appl., no. July, 2016, doi: 10.1109/IISA.2016.7785373.

[8] K. W. Church, “Emerging Trends: Word2Vec,” Nat. Lang. Eng., vol. 23, no. 1, pp. 155–162, 2017, doi: 10.1017/S1351324916000334.

[9] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, “Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study,” Procedia Comput. Sci., vol. 157, pp. 360–366, 2019, doi: 10.1016/j.procs.2019.08.178.

[10] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment Embeddings with Applications to Sentiment Analysis,” IEEE Trans. Knowl. Data Eng., vol. 28, no. 2, pp. 496–509, 2016, doi: 10.1109/TKDE.2015.2489653.

[11] L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, “Refining word embeddings for sentiment analysis,” EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 534–539, 2017, doi: 10.18653/v1/d17-1056.

[12] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[13] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” pp. 46–57, 2019, doi: 10.18653/v1/w19-3506.