Proposal Skripsi

Pelatihan Ulang Model BERT untuk Representasi Teks yang Lebih Optimal dalam Masalah Pemeringkatan Teks

Diajukan oleh

Carles Octavianus

(2006568613, carles.octavianus@sci.ui.ac.id)

Dosen Pembimbing

Sarini Abdullah S.Si., M.Stats., Ph.D.

Program Studi S1 Matematika/Statistika/Ilmu Aktuaria

Departemen Matematika FMIPA UI

Depok, Juni 2023

Ringkasan Proposal Skripsi

Permasalahan pemeringkatan teks merupakan salah satu permasalahan yang penting dalam bidang pemrosesan bahasa alami(NLP). Dalam permasalahan ini, kita diberikan sebuah kueri dan sebuah koleksi teks, dan kita diminta untuk mengurutkan teks-teks dalam koleksi tersebut berdasarkan relevansinya terhadap kueri.

Model berarsitektur Transformers (selanjutnya disebut sebagai model Transformers) seperti BERT (Devlin et al., 2018) dan RoBERTa telah menjadi model yang menjadi *state-of-the-art* model di sejumlah besar tugas NLP, termasuk klasifikasi teks dan regresi pada kalimat atau pasangan kalimat. Dalam konteks penyelesaian masalah NLP, umumnya diperlukan sebuah kalimat (sebagai contoh, tugas klasifikasi teks) atau pasangan kalimat (sebagai contoh, kesamaan semantik teks). Kalimat atau pasangan kalimat yang telah digabungkan menjadi masukkan bagi model Transformers, dan kemudian model mengeluarkan prediksi.

kita bisa meninjau permasalahan pemeringkatan teks sebagai permasalahan regresi dengan dua input. Namun, pendekatan seperti ini tidaklah efektif karena Banyaknya kombinasi pasangan yang mungkin. Sebagai contoh, jika kita ingin mencari teks yang serupa secara semantik dalam sebuah koleksi dengan $n = 10000$ teks, kita perlu melakukan $n(n-1)/2 = 49995000$ komputasi. Bahkan dengan menggunakan GPU modern seperti A100 GPU, diperlukan waktu sekitar 13 jam untuk menyelesaikan masalah tersebut.

Untuk mengatasi permasalahan komputasi yang timbul, salah satu solusinya adalah dengan menggunakan model Transformers untuk memetakan setiap kalimat atau teks ke dalam ruang vektor, sehingga teks yang memiliki kesamaan semantik akan berdekatan dalam ruang tersebut. Dalam hal ini, terdapat dua pendekatan umum yang biasa digunakan. Pertama, menggunakan rata-rata keluaran model Transformers, yang merupakan representasi kata kontekstual dalam bentuk vektor. Kedua, menggunakan keluaran model pada token pertama, yaitu token [CLS], sebagai vektor yang merepresentasikan keseluruhan teks.

Dalam penelitian ini, kami akan memaparkan beberapa metode untuk melatih ulang (*fine-tuning*) model pra-latih dengan tujuan mengatasi masalah representasi teks yang tidak optimal. Metode-metode yang akan kami tunjukkan akan berbeda dalam hal fungsi objektif, cara pelatihan, dan *dataset* yang digunakan.

Model pra-latih Transformers yang digunakan adalah IndoBERT (Wilie et al., 2020) dan mBERT (Wu and Dredze, 2020), yaitu model BERT yang telah dilatih dalam bahasa Indonesia dan multibahasa (secara berurutan).l IndoBERT akan dilatih dengan menggunakan empat *dataset* berbeda, yaitu Indo-SNLI (Bowman et al., 2015), Indo-SNLI *triplet* Indo-STS (Cer et al., 2017), dan Mmarco (Bonifacio et al., 2021). mBERT akan dilatih dengan *dataset* berupa kalimat paralel antara inggris-indonesia dengan prosedur *knowledge distillation*. setiap *dataset* akan dilatih dengan menggunakan prosedur dan fungsi objektifyang berbeda.

Akhirnya, setiap model yang dilatih ulang akan diuji dalam tugas pemeringkatan teks menggunakan *dataset* uji mMarco, miracl (Zhang et al., 2022), dan mr.tydi (Zhang et al., 2021) untuk mengukur kinerja model pada tugas pemeringakatan teks.

Kata kunci: BERT, Representasi teks, Pemeringkatan teks

1 Pendahuluan

Pemeringkatan teks merupakan permasalahan yang penting dalam bidang Pemrosesan Bahasa Alami (NLP). Dalam permasalahan ini, kita diberikan sebuah kueri dan sebuah koleksi teks, dan tujuan kita adalah mengurutkan teks-teks dalam koleksi tersebut berdasarkan relevansinya dengan kueri.

Permasalahan ini telah ada sejak sebelum penggunaan *machine learning* menjadi populer. Pada masa-masa sebelum konsep *learning to rank* dan *deep learning* muncul, pencarian dan pemeringkatan teks dilakukan menggunakan fungsi yang membandingkan kata-kata dalam kueri dengan kata-kata dalam dokumen, seperti TF-IDF dan BM25. Pendekatan ini memiliki kekurangan, seperti ketidakcocokan kosakata dan ketidakmampuan menangani kata-kata yang tidak ada dalam kueri namun memiliki makna yang sama.

Era *learning to rank* muncul sebagai respons terhadap pentingnya mesin pencari sebagai alat navigasi di web. Direktori buatan manusia seperti Yahoo! pada awal tahun 1990-an menjadi tidak efektif mengingat perkembangan pesatnya konten web. Dengan memanfaatkan data *log* atau *metadata* yang mencatat perilaku pengguna, seperti kueri dan klik, model pembelajaran mesin digunakan untuk melakukan pemeringkatan. Pendekatan ini meningkatkan pengalaman pencarian yang lebih baik, menarik lebih banyak pengguna, dan menghasilkan *metadata* tambahan serta fitur perilaku pengguna yang meningkatkan kualitas pemeringkatan. Namun, pendekatan ini juga menghadapi tantangan dalam pembuatan fitur secara manual yang membutuhkan waktu dan biaya yang besar, bahkan membutuhkan lebih dari 100 fitur untuk mendapatkan hasil yang baik.

Pada era *deep learning*, penerapan metode *deep learning* telah menjadi pendekatan *end-to-end* dalam pemeringkatan teks. *Machine learning* langsung diterapkan pada teks kueri dan tidak lagi tergantung pada pembuatan fitur buatan seperti pada era *learning to rank*. Hal ini memudahkan proses pembuatan model pemeringkatan teks. Pada era ini, terdapat dua paradigma utama dalam membangun model pemeringkatan teks.

1. Model berbasis representasi: Model ini mempelajari representasi vektor dari kueri dan dokumen dengan tujuan pemeringkatan. Representasi vektor kueri dan dokumen digunakan untuk menghitung skor relevansi antara kueri dan dokumen saat pemeringkatan. Pendekatan ini memungkinkan perhitungan representasi dokumen dilakukan secara offline dan disimpan dalam database, karena kueri dan dokumen bersifat independen.

2. Model berbasis interaksi: Model ini fokus pada penangkapan "interaksi" antara istilah atau kata-kata dalam kueri dengan istilah dalam dokumen. Pada era sebelum transformers, interaksi antara kueri dan dokumen diimplementasikan melalui matriks kesamaan, di mana setiap entri dalam matriks merepresentasikan nilai kesamaan kosinus antara kata kueri dan dokumen yang sesuai. Model ini umumnya bekerja dalam dua tahap: ekstraksi fitur (membangun matriks kesamaan) dan penilaian relevansi, yang memungkinkan pemahaman yang komprehensif tentang interaksi antara istilah dalam proses pemeringkatan.

Penelitian sebelumnya telah mengungkapkan bahwa model berbasis interaksi memiliki tingkat efektivitas yang signifikan lebih tinggi, meskipun dengan kecepatan yang lebih lambat dibandingkan dengan model berbasis representasi. Di sisi lain, model berbasis representasi memberikan keuntungan dalam menyederhanakan proses pemeringkatan teks dengan

hanya menggunakan perbandingan kesamaan antara vektor kueri dan vektor dokumen yang telah dihitung sebelumnya. Kelebihan ini memungkinkan pemrosesan yang lebih efisien pada koleksi dokumen yang besar dibandingkan dengan model berbasis interaksi.

Model Transformers, seperti BERT dan RoBERTa, telah menjadi acuan dalam banyak tugas Pemrosesan Bahasa Alami (NLP), termasuk klasifikasi teks dan regresi pada kalimat atau pasangan kalimat. Dalam penyelesaian masalah NLP, kita perlu menggunakan kalimat tunggal atau pasangan kalimat sebagai input bagi model Transformers, yang kemudian menghasilkan prediksi.

Namun, model berbasis interaksi pada Transformers tidak seefisien dengan model berbasis interaksi pra-transformers dalam tugas pemeringkatan teks. Hal ini disebabkan oleh perhitungan nilai relevansi yang lebih mahal pada Transformers dibandingkan pada era pra-transformers (CNN, RNN). Sebagai contoh, jika kita ingin mencari teks yang serupa secara semantik dalam koleksi dengan $n = 10000$ teks, kita perlu melakukan $n(n-1)/2 = 49995000$ komputasi. Meskipun menggunakan GPU modern seperti A100 GPU, waktu yang diperlukan untuk menyelesaikan masalah tersebut adalah sekitar 13 jam.

Untuk mengatasi masalah komputasi yang timbul, salah satu solusinya adalah menggunakan model Transformers dengan pendekatan berbasis representasi. Pendekatan ini memetakan setiap kalimat atau teks ke dalam ruang vektor, di mana teks dengan kesamaan semantik akan berdekatan dalam ruang tersebut. Terdapat dua pendekatan umum yang sering digunakan. Pertama, menggunakan rata-rata keluaran model Transformers sebagai representasi teks. Kedua, menggunakan keluaran model pada token pertama, yaitu token [CLS], sebagai vektor yang merepresentasikan teks. Dengan demikian, model berbasis representasi lebih efisien secara komputasi dibandingkan model berbasis interaksi.

Namun, perlu diingat bahwa penggunaan model pra-latih seperti BERT atau RoBERTa dalam konteks ini menghasilkan representasi yang tidak optimal. Hal ini disebabkan oleh kenyataan bahwa model pra-latih tersebut tidak secara khusus dilatih untuk tujuan representasi teks yang terkait dengan kesamaan semantik.

Dalam penelitian ini, kami akan memaparkan metode-metode untuk melatih ulang (*fine tuning*) model pra-latih guna mengatasi masalah representasi teks yang tidak optimal. Metode-metode yang kami jelaskan berbeda dalam fungsi objektif, cara pelatihan, dan *dataset* yang digunakan. Kami akan melatih ulang model dalam bahasa Indonesia untuk tujuan pemeringkatan teks dalam bahasa tersebut. Saat ini, belum terdapat model transformers yang dilatih ulang secara khusus untuk tujuan pemeringkatan teks (berdasarkan hasil pencarian model di website huggingface.co).

Model pra-latih Transformers yang kami gunakan adalah mBERT (Wu and Dredze, 2020) dan IndoBERT (Wilie et al., 2020), yaitu model BERT yang telah dilatih secara multibahasa dan untuk dalam bahasa Indonesia (secara berurutan). IndoBERT akan dilatih ulang dengan empat *dataset* yang berbeda, yaitu Indo-SNLI, Indo-STS, Indo-SNLI trpilet, dan Mmarco Indonesia. Meskipun permasalahan NLI dan STS berbeda dengan pemeringkatan teks, namun keduanya memiliki kesamaan tugas dasar, yaitu dalam mengukur kesamaan semantik antara dua teks. Oleh karena itu, kami menggunakan *dataset* tersebut untuk melatih ulang model pra-latih dengan tujuan mengevaluasi dampak pemilihan *dataset* dan fungsi objektif yang berbeda terhadap performa model yang dilatih ulang dalam tugas pemeringkatan teks. Selain itu, pelatihan dengan *dataset* tersebut relatif mudah dilakukan.

Selanjutnya, kami akan melatih ulang model pra-latih mBERT (*multilingual* BERT) den-

gan menggunakan prosedur *knowledge distillation*. Dalam prosedur ini, kami hanya memerlukan *dataset* yang berisi pasangan kalimat dalam bahasa Inggris-Indonesia (yang umumnya tersedia dalam *dataset* untuk tugas mesin penerjemah) dan sebuah model pemeringkatan teks yang sudah baik dalam bahasa Inggris. Prosedur ini cocok untuk melatih model dalam bahasa dengan sumber daya terbatas, seperti bahasa Indonesia.

Akhirnya, setiap model yang dilatih ulang akan diuji dalam tugas pemeringkatan teks menggunakan *dataset* uji mMarco, miracl (Zhang et al., 2022), dan mr.tydi (Zhang et al., 2021) untuk mengukur kinerja mereka.

2 Masalah Penelitian

Bagaimana cara melakukan pelatihan ulang yang optimal pada model BERT untuk mendapatkan representasi teks yang lebih optimal dalam masalah pemeringkatan teks?

3 Tujuan Penelitian

Penelitian ini bertujuan untuk menguji beberapa teknik *fine tuning* model BERT guna memperoleh representasi teks yang lebih optimal dalam konteks pemeringkatan teks. Untuk mengevaluasi peningkatan yang diperoleh dari teknik *fine tuning* yang digunakan, perbandingan dilakukan antara model yang dilatih ulang dan model yang tidak dilatih ulang pada kualitas pemeringkatan teks menggunakan tiga *dataset* yang berbeda.

4 Batasan Penelitian

Penelitian ini berfokus pada kualitas representasi teks untuk tugas pemeringkatan teks dari model yang terbatas pada tiga *dataset* yang diuji. Harap dicatat bahwa performa model mungkin tidaklah representatif untuk *dataset* lain yang tidak diuji.

5 Tinjauan Pustaka

5.1 Permasalahan Pemeringkatan Teks

Dalam penelitian ini, fokus utama adalah pada permasalahan pemeringkatan teks berdasarkan sebuah kueri atau pertanyaan q. Tujuan dari permasalahan ini adalah menghasilkan sebuah daftar terurut yang terdiri dari kl teks $\{d_1, d_2, \ldots, d_k\}$ dari sebuah koleksi terbatas teks $\mathcal{C} = \{d_i\}_{i \in I}$, dengan tujuan memaksimalkan metrik yang diinginkan, seperti *normalized Discounted Cumulative Gain* (nDCG), *Average Precision* (AP), dan sebagainya. (Lin et al., 2020)

Metrik-metrik ini digunakan untuk mengukur relevansi dari daftar yang dihasilkan, yang akan dijelaskan secara lebih rinci pada bagian selanjutnya dari proposal skripsi ini. Dalam literatur, permasalahan pemeringkatan teks sering juga disebut sebagai permasalahan *top-k retrieval*, dengan k adalah jumlah teks yang ingin diambil dari koleksi $\mathcal{C}$.

Karena permasalahan ini melibatkan konsep keterurutan, keluaran dari model atau algoritma biasanya berbentuk pasangan $\{(d_1, s_1), (d_2, s_2), \ldots, (d_k, s_k)\}$, di mana d_i mewakili

teks yang diambil dari koleksi $\mathcal{C}$ dan s_i merupakan skor yang menunjukkan tingkat relevansi teks d_i terhadap kueri q, dengan $s_1 \geq s_2 \geq \ldots \geq s_k$.

5.2 Nilai Relevansi

Dalam permasalahan pemeringkatan teks, meskipun relevansi merupakan tujuan yang ingin dicapai dan diukur melalui metrik-metrik evaluasi, mendefinisikan dan mengukur relevansi teks secara objektif sehingga semua pihak memiliki persepsi yang serupa merupakan tugas yang kompleks dan menantang (silakan lihat Bab 2.3 pada (Lin et al., 2020)). Oleh karena itu, dalam penelitian ini, fokus tidak terlalu diberikan pada perdebatan mengenai relevansi itu sendiri. Sebagai gantinya, diasumsikan bahwa relevansi dari pasangan (q_i, d_i) dapat diukur dengan baik.

Data yang umumnya digunakan pada permasalahan pemeringkatan teks adalah himpunan pasangan (q, d, r), dengan r merupakan penilaian relevansi yang diannotasikan oleh sistem atau manusia. Dalam kasus yang sederhana, r dapat berupa variabel biner yang menunjukkan apakah d relevan terhadap q atau tidak. Namun, dalam kasus yang lebih kompleks, r dapat berupa skala ordinal yang mengindikasikan tingkat relevansi d terhadap q.

Penilaian relevansi r memiliki dua kegunaan utama. Pertama, dengan adanya penilaian relevansi pada pasangan (q, d), kita dapat melatih model pemeringkatan secara *supervised* menggunakan pendekatan pembelajaran mesin. Kedua, dengan adanya penilaian relevansi r memungkinkan kita untuk mengevaluasi model pemeringkatan tersebut. Dengan adanya penilaian relevansi, kita dapat mendefinisikan metrik-metrik evaluasi untuk mengukur kinerja model pemeringkatan, seperti yang akan dijelaskan lebih lanjut pada bagian berikutnya dari proposal skrispsi ini.

5.2.1 Metrik untuk Evaluasi

Berikut ini adalah beberapa metrik yang sering digunakan dalam mengevaluasi model. Untuk menghindari kebingungan, kita akan menggunakan notasi $R = \{(i, d_i)\}_{i=1}^{l}$ sebagai gantinya daripada $R = \{(s_i, d_i)\}_{i=1}^{l}$, yang berarti R adalah kumpulan l dokumen teratas dengan urutan i yang dihasilkan oleh model perangkingan untuk kueri q. Selain itu, banyak metrik yang dihitung pada posisi pemotongan k, kita akan menyebutnya sebagai (metrik)@k, dengan $k \leq l$. Ketika menggunakan metrik pemotongan, definisi dari R menjadi $R = \{(i, d_i)\}_{i=1}^{k}$.

1. Presisi didefinisikan sebagai fraksi dokumen dalam daftar terurut R yang relevan, yaitu:

$$Presisi(R, q) = \frac{\sum_{(i,d)\in R} rel(q, d)}{|R|} \tag{1}$$

   dengan $rel(q, d) = 1$ jika q dan d saling relevan dan 0 jika tidak.

   Presisi mengukur sejauh mana teks dalam daftar terurut R yang relevan. Dalam konteks ini, relevansi dokumen terhadap kueri q diasumsikan sebagai relevansi biner.

2. *Recall*, didefinisikan sebagai pecahan dari dokumen relevan (dalam koleksi teks $\mathcal{C}$ secara keseluruhan) untuk kueri q yang berhasil ditemukan dalam daftar terurut R. Secara matematis, rumusnya adalah:

$$Recall(R, q) = \frac{\sum_{(i,d)\in R} rel(q, d)}{\sum_{d\in\mathcal{C}} rel(q, d)}, \tag{2}$$

di mana $rel(q, d) \in \{0, 1\}$ menunjukkan apakah dokumen d relevan dengan kueri q.

3. *Normalized Discounted Cumulative Gain* (nDCG) adalah metrik yang umumnya digunakan untuk mengukur kualitas dari pencarian situs web. Tidak seperti metrik lain yang telah disebutkan sebelumnya, nDCG dirancang untuk penilaian relevansi r dengan skala ordinal. Sebagai contoh, jika relevansi diukur dengan skala 5, maka $rel(q, d) \in 1, 2, 3, 4, 5$. *Discounted Cumulative Gain* (DCG) dapat didefinisikan sebagai berikut:

$$DCG(R, q) = \sum (i, d) \in R \frac{2^{rel(q,d)} - 1}{\log_2(i + 1)} \tag{3}$$

Dalam perhitungan ini, ada dua faktor yang digunakan: (1) tingkat relevansi (yaitu, teks yang sangat relevan memiliki nilai yang lebih tinggi dibandingkan dengan teks yang hanya relevan), dan (2) peringkat dimana hasil tersebut muncul (hasil yang relevan yang muncul di posisi atas dalam daftar terurut R memiliki nilai yang lebih tinggi). Kata 'diskon' dalam DCG merujuk pada penurunan nilai *gain* saat dokumen muncul di posisi yang lebih rendah dalam daftar terurut, yang merupakan efek dari faktor (2). Selanjutnya, kita memperkenalkan metrik nDCG:

$$nDCG(R, q) = \frac{DCG(R, q)}{IDCG(R, q)} \tag{4}$$

Di mana IDCG adalah representasi ideal dari daftar yang telah diurutkan: di mana dokumen diurutkan dari penilaian relevansi r yang tertinggi hingga yang terendah. Dengan definisi ini, nDCG merepresentasikan DCG yang telah dinormalisasi dalam rentang [0,1] berdasarkan daftar terurut yang ideal.

4. *Reciprocal Rank* (RR) merupakan nilai peringkat terkecil dari sebuah dokumen yang relevan. Dengan kata lain, jika dokumen relevan muncul di posisi pertama, nilai reciprocal rank adalah 1, nilai 1/2 jika muncul di posisi kedua, nilai 1/3 jika muncul di posisi ketiga, dan seterusnya. Jika dokumen relevan tidak muncul dalam l atau k hasil teratas, maka query tersebut mendapatkan skor nol. Mirip dengan precision dan recall, RR dihitung berdasarkan penilaian relevansi biner. Secara matematis Reciprocal Rank dihitung sebagai berikut:

$$RR(R, q) = \frac{1}{rank_i} \tag{5}$$

5.3 Survei Metode Pemeringkatan Teks

5.3.1 TF-IDF dan BM25

Pada era sebelum *learning to rank* dan *deep learning*, pencarian atau pemeringkatan teks dilakukan dengan menggunakan fungsi yang membandingkan kata-kata dalam kueri dengan kata-kata dalam dokumen. Secara lebih spesifik, misalnya q adalah kueri

dan d adalah dokumen, dan T_q dan T_d adalah himpunan kata-kata dalam q dan d masing-masing, maka fungsi pemeringkatannya dapat dirumuskan sebagai berikut:

$$S(q,d) = \sum_{t \in T_q \cap T_d} f(t) \tag{6}$$

Dengan f adalah fungsi yang melibatkan kata dan statistik kata yang berkaitan dengan dokumen atau kueri. Sebelum melakukan perhitungan skor ini, terdapat proses seperti *stemming*, normalisasi, dan mengubah huruf menjadi huruf kecil agar $|T_q \cap T_d|$ dapat lebih besar, sehingga dapat meningkatkan mekanisme skoring.

Statistik yang sering digunakan untuk menghitung skor relevansi antara lain frekuensi kata (*term frequency*, $tf_{t,d}$) yang mengukur seberapa sering suatu kata t muncul dalam sebuah dokumen d, frekuensi dokumen (*document frequency*, df_t) yang mengukur jumlah dokumen yang mengandung kata t, dan panjang dokumen (banyaknya kata dalam dokumen). Dari statistik frekuensi kata dan frekuensi dokumen, dapat diturunkan penghitungan tf-idf (*term frequency-inverse document frequency*:

$$tfidf(t,d) = \sum_{t \in T_q \cap T_d} tf_{t,d} \frac{|D|}{df_t} \tag{7}$$

Fungsi skoring tf-idf juga dapat ditulis dengan skala logaritmik:

$$tfidf(t,d) = \sum_{t \in T_q \cap T_d} \log(1 + tf_{t,d}) \log(\frac{|D|}{df_t}) \tag{8}$$

Perlu diperhatikan bahwa nilai $\log(1 + tf_{t,d})$ akan lebih tinggi ketika kata t muncul lebih sering dalam dokumen d, sedangkan nilai $\log(\frac{|D|}{df_t})$ akan lebih tinggi ketika kata t muncul pada jumlah dokumen yang lebih sedikit. Dengan demikian, kata-kata yang muncul dalam jumlah dokumen yang lebih sedikit akan memiliki skor yang lebih tinggi, sedangkan kata-kata umum akan menghasilkan skor yang lebih rendah.

Salah satu fungsi skoring yang banyak digunakan hingga saat ini adalah BM25 (Best-Match25) yang diusulkan oleh Robertson et al. pada tahun 1994. Fungsi skoring BM25 dirumuskan sebagai berikut:

$$BM25(q,d) = \sum_{t \in T_d \cap T_q} \frac{tf_{t,d}}{k_1 \left((1-b) + b\frac{dl_d}{avgdl}\right) + tf_{t,d}} \log \frac{|D| - df_t + 0.5}{df_t + 0.5} \tag{9}$$

Di mana k_1 dan b adalah parameter yang dapat diatur, dl_d adalah panjang dokumen d, dan $avgdl$ adalah panjang rata-rata dokumen dalam koleksi D.

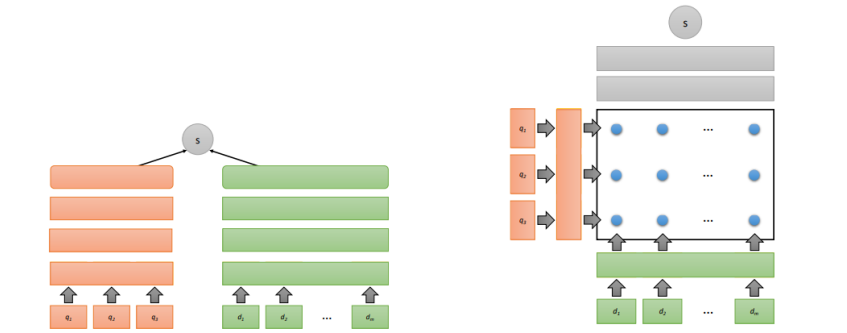
Hingga saat ini, BM25 tetap menjadi salah satu fungsi skoring yang paling banyak digunakan dalam pencarian atau pemeringkatan berbasis kata kunci.

Learning to Rank dan Deep Learning

Era *learning to rank* muncul karena meningkatnya pentingnya mesin pencari sebagai alat navigasi di web. Direktori buatan manusia seperti Yahoo! pada awal tahun 1990-an menjadi tidak efektif mengingat perkembangan pesat konten web. Dengan memanfaatkan data *log* atau *metadata* yang mencatat perilaku pengguna, seperti kueri dan

klik, model pembelajaran mesin digunakan untuk melakukan pemeringkatan. Hal ini meningkatkan pengalaman pencarian yang lebih baik, menarik lebih banyak pengguna, dan menghasilkan metadata tambahan dan fitur perilaku pengguna yang meningkatkan kualitas pemeringkatan (**??**).

Setelah era *learning to rank*, muncul era *deep learning*. Dalam konteks pemeringkatan teks, metode *deep learning* menarik perhatian dengan dua alasan utama. Pertama, representasi vektor yang kontinu dari teks memberikan alternatif dalam pemeringkatan teks mengatasi batasan sistem temu balik informasi yang hanya berdasarkan pencocokan kata yang tepat (bukan berdasarkan pencocokan secara makna). Kedua, *artificial neural network* menghilangkan kebutuhan akan fitur yang dibuat secara manual, yang merupakan tantangan besar dalam membangun sistem temu balik informasi pada era *learning to rank*.



(a) a generic representation-based neural ranking model (b) a generic interaction-based neural ranking model

Gambar 1: Ilustrasi dua kategori utama model pemeringkatan teks. (a) merupakan model berbasis representasi, di mana kueri dan dokumen direpresentasikan sebagai vektor dan skor relevansi dihitung berdasarkan perbandingan antara vektor kueri dan dokumen. (b) merupakan model berbasis interaksi, nilai relevansi dihitung langsung dari interaksi antara kueri dan dokumen (Lin et al., 2020).

Pada era *deep learning*, model untuk pemeringkatan teks dapat dibedakan menjadi dua kategori utama: model berbasis representasi dan model berbasis interaksi.

(a) Model berbasis representasi: Model ini mempelajari representasi vektor dari kueri dan dokumen untuk tujuan pemeringkatan. Representasi vektor kueri dan dokumen akan dibandingkan saat pemeringkatan untuk menghitung skor relevansi antara kueri dan dokumen. Pendekatan ini memungkinkan perhitungan representasi dokumen dilakukan secara offline dan disimpan dalam *database*, karena kueri dan dokumen bersifat independen. Contoh model awal dalam era *deep learning* adalah *Deep Structure Semantic Model* (DSSM) (Huang et al., 2013), yang membangun n-gram karakter dari input dan menggunakan *artifical neural network* untuk menghasilkan representasi vektor.

(b) Model berbasis interaksi: Model ini fokus pada penangkapan "interaksi" antara istilah atau kata-kata dalam kueri dengan istilah dalam dokumen. Pada era pra-transformers, interaksi antara kueri dan dokumen diimplementasikan melalui matriks kesamaan, di mana setiap entri dalam matriks merepresentasikan nilai

kesamaan kosinus antara istilah kueri dan dokumen yang sesuai. Model ini umumnya bekerja dalam dua tahap: ekstraksi fitur (membangun matriks interaksi) dan penilaian relevansi, yang memungkinkan pemahaman yang komprehensif tentang interaksi antara istilah dalam proses pemeringkatan.

5.4 Transformers dan BERT

5.4.1 Transformers

Transformers adalah model *deep learning* yang memanfaatkan mekanisme *attention*, yang pertama kali diperkenalkan sebagai mekanisme yang membantu meningkatkan performa model *seq2seq* (seperti RNN) dalam mesin penerjemah (Yang et al., 2016)

Transformers (Vaswani et al., 2017) merupakan model *seq2seq* yang bertujuan sebagai model mesin penerjemah yang menggunakan *purely* mekanisme *attention* (tanpa model *sequential* seperti RNN layer). Model ini terdiri dari *encoder* dan *decoder* yang terdiri dari beberapa blok *attention* dan *feed-forward layer* yang saling terhubung. *Encoder* bertugas untuk mengubah input menjadi representasi yang lebih abstrak (representasi kontekstual vektor dari kata, atau biasa kita sebut sebagai *contextual word embedding*). Sedangkan decoder bertugas untuk mengubah representasi tersebut menjadi output yang diinginkan, dalam kasus mesin penerjemah, output yang diinginkan adalah kalimat terjemahan.

Terdapat dua mekanisme *attention* pada arsitektur Transformers, yaitu *self-attention* pada *Encoder*, dan *encoder-decoder attention* pada *decoder*. *Encoder-decoder attention merupakan* mekanisme *attention* yang sama yang dijelaskan oleh Yang et al. (2016). Sedangkan, *self-attention* merupakan mekanisme *attention* yang baru diperkenalkan pada arsitektur Transformers.

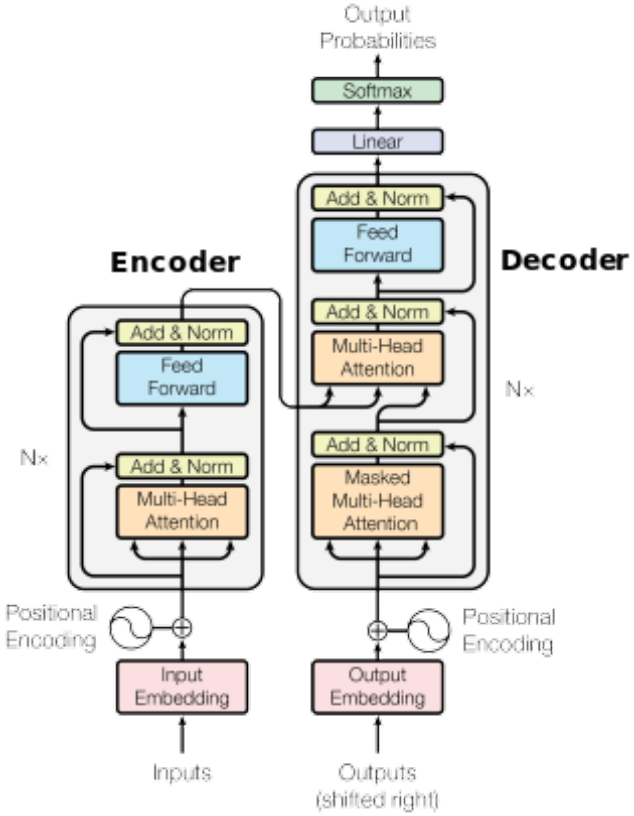
Layer self-attention ditambahkan dengan *layer feed forward*, menggantikan layer rekuren (*reccurent cell*) pada model sekuensial seperti RNN-encoder untuk menghasilkan representasi kontekstual dari kata-kata pada kalimat input.

Pada satu transformers-encoder terdiri dari beberapa layer, yaitu, *multi head self-attention layer, feed forward layer, layer normalization layer,* dan *residual connection.* tumpukan transformers-encoder inilah yang menjadi arsitektur dari model BERT.

5.4.2 BERT

Pada dasarnya, BERT (Bidirectional Encoder Representations from Transformers) adalah sebuah model *neural network* yang digunakan untuk menghasilkan vektor representasi kata dengan mengandung informasi kontekstual dari kalimat yang diberikan (Devlin et al., 2018). Meskipun pada awalnya BERT hanya dapat menghasilkan representasi kata dalam bahasa Inggris, saat ini telah ada model multibahasa dari BERT yang disebut mBERT (Wu and Dredze, 2020), dan bahkan telah ada model pra-latih untuk bahasa Indonesia yang disebut IndoBERT (Wilie et al., 2020).

BERT menerima masukan berupa urutan token (kata atau subkata) ditambah dengan beberapa elemen lainnya, seperti *positional embedding* dan *segment embedding* (lihat



Gambar 2: Ilustrasi arsitektur Transformers terdiri dari dua bagian: transformer-encoder dan transformer-decoder (Vaswani et al., 2017).

3), dan menghasilkan representasi untuk setiap token yang konteks dependen. Berbeda dengan model word *embedding* seperti word2vec (Mikolov et al., 2013) atau GloVe (Pennington et al., 2014) yang menghasilkan representasi yang bersifat konteks independen (statis, di mana setiap token memiliki representasi yang sama, tidak tergantung pada konteks). BERT memiliki tujuan yang serupa dengan model ELMo (Peters et al., 2018), yaitu menghasilkan representasi kontekstual, namun BERT didasarkan pada arsitektur transformers-encoder (tumpukan), sementara Elmo didasarkan pada arsitektur RNN.

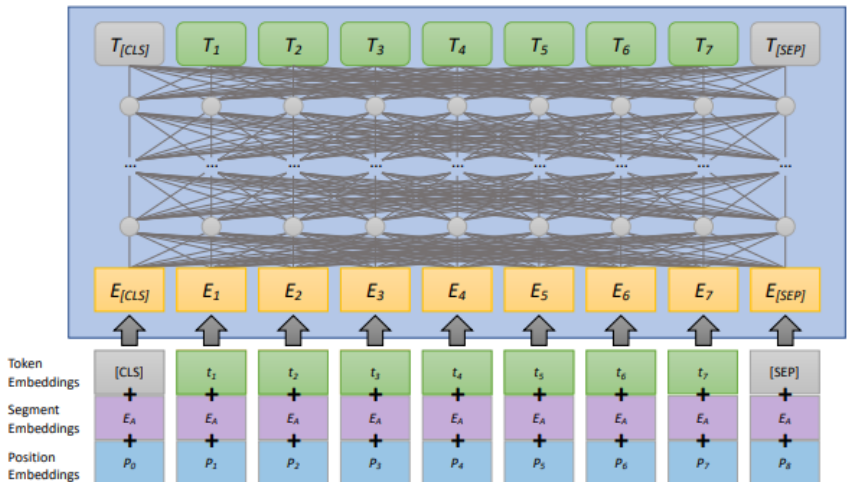
Tujuan dari representasi yang terkontekstual adalah untuk menangkap karakteristik kompleks dari sebuah bahasa, seperti sintaks dan semantik, serta bagaimana makna kata dapat berubah sesuai dengan konteks linguistiknya (polisemi).

Masukan dan keluaran dari BERT dapat diilustrasikan dalam gambar 3, di mana representasi token masukan ditulis sebagai:

$$\left[E_{[CLS]}, E_1, E_2, \ldots, E_{[SEP]}\right] \tag{10}$$

dan representasi keluaran (*embedding* kontekstual) ditulis sebagai:

$$\left[T_{[CLS]}, T_1, T_2, \ldots, T_{[SEP]}\right] \tag{11}$$

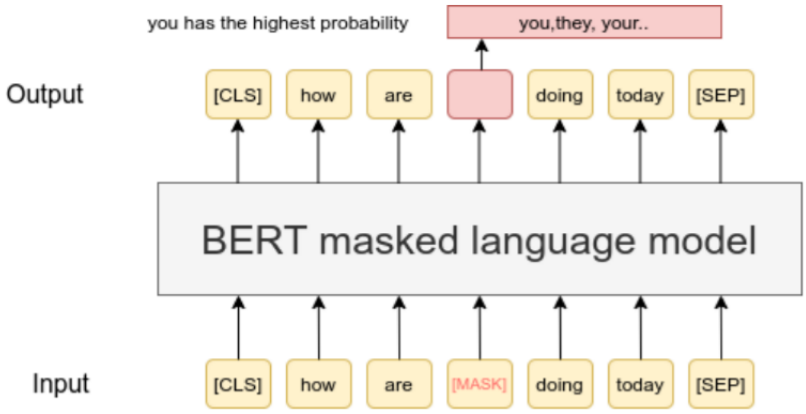


Gambar 3: Ilustrasi arsitektur BERT: barisan kata diubah menjadi token, *segment*, dan *positional embedding*. Jumlahan *embedding* ini menghasilkan *embedding* input, yang melewati 12 blok transformers-encoder. Representasi kontekstual vektor kata diambil dari blok terakhir (Lin et al., 2020).

Salah satu alasan mengapa BERT menjadi populer adalah karena cara BERT menggunakan ide dari ULMfit (Universal Language Model Fine Tuning) (Howard and Ruder, 2018) untuk melakukan *pre-training* model BERT secara unsupervised pada korpus yang sangat besar, seperti Wikipedia, sehingga *embedding* kata kontekstual telah terbentuk sebelumnya. Model BERT dilakukan *pre-training* dengan menggunakan objektif Masked Language Modeling (MLM) (lihat gambar 4). Pada tugas dengan objektif MLM, dalam suatu korpus, secara acak kita menghilangkan (mask) suatu token dari korpus tersebut dan kita meminta model untuk menebak kata yang dihilangkan tersebut. Kemudian, kita melakukan pelatihan dengan menggunakan *categorical cross entropy loss*. Selain itu, dalam artikel asli BERT, juga dilakukan *pre-training* dengan objektif *next sentence prediction*, tetapi pada penelitian selanjutnya, tidak ada indikasi bahwa objektif ini meningkatkan kinerja BERT pada tugas-tugas NLP.

Dalam konteks masukan untuk model BERT, sebuah barisan kata (kalimat) akan mengalami proses tokenisasi terlebih dahulu. Tokenisasi ini bertujuan untuk mengubah barisan kata menjadi representasi vektor kata (sparse vektor). Umumnya, kalimat akan di-tokenisasi menggunakan *tokenizer* seperti WordPiece, BPE (Sennrich et al., 2016), atau sentencepiece *tokenizer*. Tujuan dari penggunaan *tokenizer* adalah untuk mengurangi jumlah jenis kata yang harus diingat (*vocabulary*). Sebagai contoh, dengan menggunakan kosakata WordPiece yang digunakan oleh BERT, kata "scrolling" akan diubah menjadi "scroll" + "##ing". Penambahan tanda pagar ganda (##) pada subkata menunjukkan bahwa subkata tersebut "terhubung" dengan subkata sebelumnya. Pada BERT *original*, jumlah kosakata yang digunakan adalah 30.000 kata (atau subkata) dengan menggunakan wordpiece *tokenizer*.

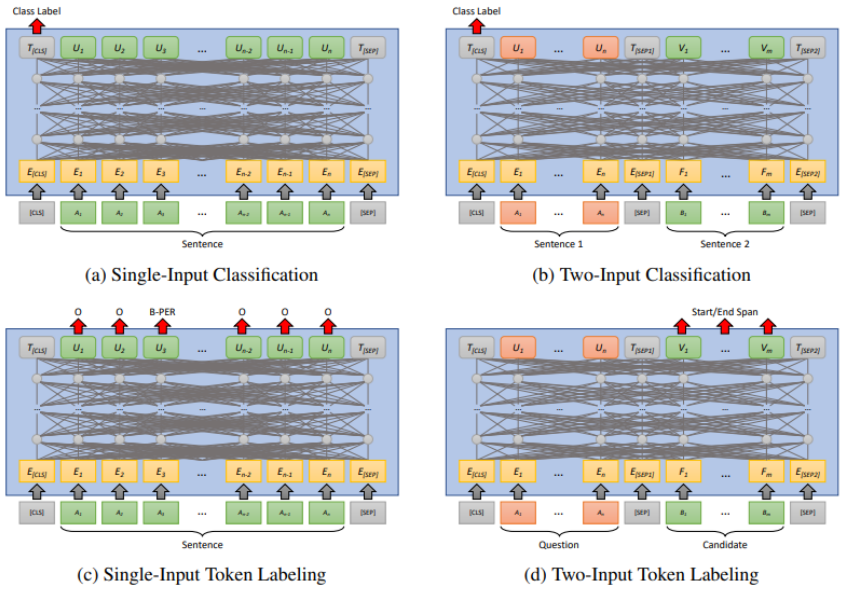
Selain token *embedding*, terdapat dua jenis masukan lainnya yang diperlukan oleh model BERT, yaitu *segment embedding* dan *positional embedding*, yang akan dijelaskan sebagai berikut:



Gambar 4: Ilustrasi objektif Masked Language Modeling (MLM) pada BERT: sebuah kata (token) secara acak di-hilangkan (mask) dan model diminta untuk menebak kata yang dihilangkan.

(a) Segment *embedding*, yaitu *embedding* yang dipelajari (*learned embedding*) yang menunjukkan apakah token tersebut termasuk dalam input pertama (A) atau input kedua (B) dalam tugas-tugas yang melibatkan dua input (ditandai dengan EA dan EB pada).

(b) Position *embedding*, yaitu *embedding* yang dipelajari (*learned embedding* yang mencerminkan posisi token dalam urutan, sehingga memungkinkan BERT untuk memahami urutan linear dari token-token tersebut



(a) Single-Input Classification

(b) Two-Input Classification

(c) Single-Input Token Labeling

(d) Two-Input Token Labeling

Gambar 5: Ilustrasi keempat tipe tugas turunan dengan menggunakan model pra-latih BERT. (Lin et al., 2020)

Meskipun pada dasarnya model BERT mengubah urutan token masukan menjadi urutan token keluaran yang kontekstual, dalam praktiknya, BERT terutama digunakan

untuk empat jenis tugas turunan (gambar 5):

(a) Tugas klasifikasi dengan satu input, seperti analisis sentimen pada segmen teks tunggal. BERT juga dapat digunakan untuk tugas regresi.

(b) Tugas klasifikasi dengan dua input, contohnya adalah mendeteksi apakah dua kalimat merupakan parafrase. Pada dasarnya, tugas regresi juga dapat dilakukan di sini.

(c) Tugas pelabelan token pada satu *input*, misalnya pengenalan entitas bernama (*named entity recognition*). Pada tugas ini, setiap token pada masukan akan diberi label yang sesuai dengan arti dari token tersebut (misalnya, kata "Jakarta" dapat diberi label "lokasi").

(d) Tugas pelabelan token pada dua input, seperti diberikan pasangan (query, teks), kita perlu mencari bagian teks yang menjadi jawaban untuk query tersebut. Masalah ini sering disebut sebagai *machine reading comprehension* (MRC) atau *question answering* (QA).

Terakhir, pada BERT terdapat token khusus seperti [CLS]. Nilai keluaran BERT untuk token CLS, yaitu $T_{[\text{CLS}]}$, sering digunakan sebagai representasi kalimat atau teks dalam tugas-tugas klasifikasi dan regresi yang diterapkan pada tingkat kalimat.

5.5 Representasi Teks

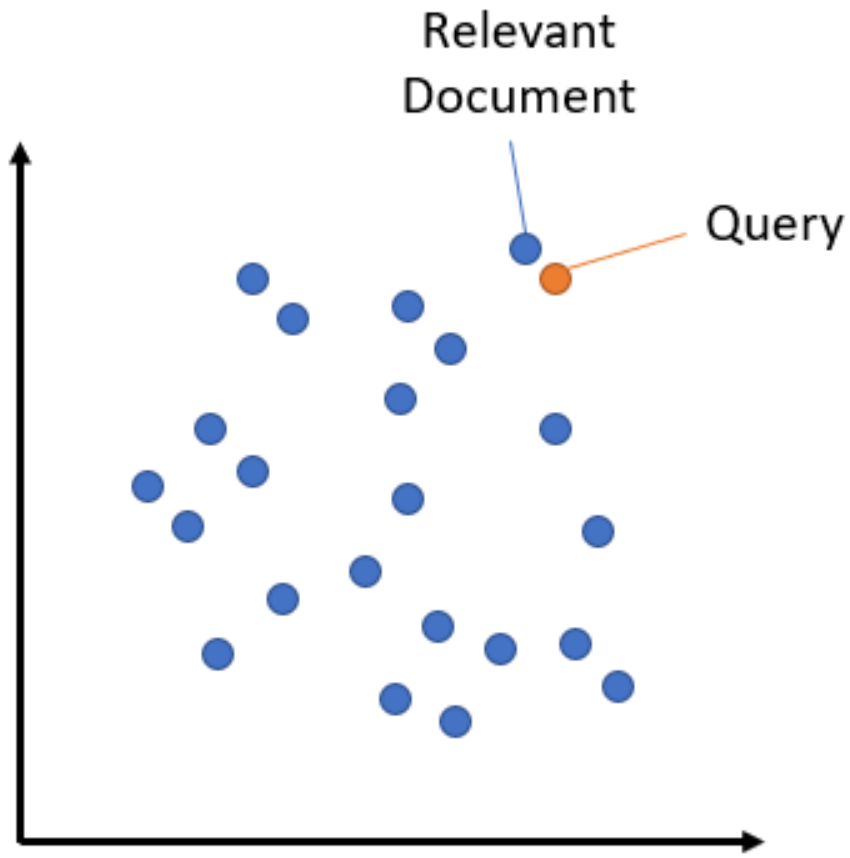
Dengan menggunakan model yang berbasis representasi, permasalahan pemeringkatan teks dapat disederhanakan menjadi permasalahan representasi.

Kita ingin mencari fungsi $\eta : [t_1 \dots t_n] \rightarrow \mathbb{R}^n$ yang mengubah sebuah teks menjadi vektor representasi. Fungsi tersebut diharapkan memenuhi sifat bahwa untuk kueri q, dokumen relevan d_p, dan dokumen irelevan d_n, kita ingin memaksimalkan $\phi(\eta(q), \eta(d_p))$ dan meminimalkan $\phi(\eta(q), \eta(d_n))$, di mana ϕ adalah fungsi untuk mengukur keserupaan antara vektor-vektor tersebut. Untuk tetap mempertahankan kecepatan dalam pemeringkatan teks, biasanya kita memilih ϕ sebagai fungsi yang sederhana (non-*machine learning*), seperti *cosine similarity, dot product,* atau jarak euclidean.

Dengan menggunakan nilai yang dihasilkan oleh ϕ antara sebuah kueri dan teks, kita dapat melakukan pemeringkatan teks dengan membandingkan nilai ϕ dari berbagai teks. Teks yang memiliki nilai ϕ yang lebih tinggi akan diberikan peringkat yang lebih tinggi.

Dalam konteks penggunaan model BERT, $\eta(input) = BERT(input)_{[CLS]}$, di mana $BERT(input)_{[CLS]}$ adalah nilai keluaran dari token [CLS] pada model BERT. Selain menggunakan representasi dari token [CLS], kita juga dapat merata-ratakan representasi dari semua token dalam teks, yaitu $\eta(input) = \frac{1}{n}\sum_{i=1}^{n} BERT(input)_i$, di mana $BERT(input)_i$ adalah nilai keluaran dari token ke-i pada model BERT.

Model berbasis representasi tidak hanya digunakan untuk permasalahan pemeringkatan teks, tetapi juga dapat digunakan untuk berbagai permasalahan NLP lainnya yang serupa dengan pemeringkatan teks.



Gambar 6: Ilustrasi representasi teks: fungsi η memetakan teks ke ruang vektor. Tujuan utama η adalah menghasilkan representasi yang saling berdekatan untuk teks dengan semantik yang serupa di dalam ruang vektor (SBERT.net).

5.6 Tugas yang Serupa dengan Pemeringkatan Teks

Sebelum menjelaskan proses pelatihan ulang model dan pemilihan *dataset*, perlu dibahas beberapa tugas atau masalah dalam bidang Pemrosesan Bahasa Alami (NLP) yang memiliki kesamaan dengan pemeringkatan teks. Kemiripan antara tugas-tugas ini dengan pemeringkatan teks menjadi dasar dalam pemilihan *dataset* yang akan digunakan dalam pelatihan ulang model.

5.6.1 Kesamaan Semantik Teks (Semantic Textual Similarity - STS)

Tugas STS (Kesamaan Semantik Teks) (Cer et al., 2017) merupakan suatu tugas dalam pemrosesan bahasa alami (NLP) yang bertujuan untuk mengukur tingkat kesamaan makna antara dua teks. Tugas ini difokuskan pada penentuan sejauh mana dua teks memiliki kesamaan atau keterkaitan semantik.

Tugas STS umumnya melibatkan pemberian skor atau nilai yang menggambarkan sejauh mana teks-teks tersebut serupa secara semantik. Skor ini dapat berupa angka dalam skala numerik, skala ordinal, atau menggunakan metode penilaian lainnya (lihat tabel **??** sebagai contoh). Tugas STS dapat mencakup berbagai jenis perbandingan

teks, seperti perbandingan antara kalimat-kalimat individu, pasangan kalimat, atau bahkan dokumen-dokumen secara keseluruhan.

5.6.2 Penalaran Implikasi *(Entailment)*

Tugas penalaran implikasi merujuk pada jenis tugas dalam pemrosesan bahasa alami (NLP) yang bertujuan untuk menentukan hubungan logis antara dua teks, yang biasanya disebut sebagai "premis" dan "hipotesis" (Giampiccolo et al., 2007). Tugas ini melibatkan evaluasi apakah makna hipotesis secara logis diimplikasikan oleh premis, yang berarti informasi dalam premis secara logis mendukung atau menyiratkan informasi dalam hipotesis.

Tugas penalaran implikasi sering melibatkan proses klasifikasi, di mana hubungan antara premis dan hipotesis dikategorikan ke dalam beberapa label, seperti "implikasi" (premis secara logis mengimplikasikan hipotesis), "kontradiksi" (premis bertentangan dengan hipotesis), atau "netral" (tidak ada hubungan logis yang jelas antara premis dan hipotesis). Tugas ini umum digunakan dalam penelitian dan evaluasi NLP untuk mengukur kemampuan model dalam memahami dan menyusun hubungan antara pernyataan tekstual.

5.7 Dataset yang Digunakan

Bagian berikut ini menjelaskan *dataset* yang digunakan untuk pelatihan dan pengujian model.

Untuk melatih ulang model pra-latih IndoBERT, digunakan beberapa *dataset* yang meliputi Indo-STS, Indo-SNLI, mMarco. model pra-latih mBERT dilatih dengan menggunakan kalimat paralel indonesia-inggris.

Dataset yang digunakan untuk pengujian model meliputi *dataset* uji yang sama dengan *dataset* pelatihan, yaitu Indo-STS, Indo-SNLI, Indo-SNLI *triplet* dan mMarco. Selain itu, terdapat dua *dataset* tambahan, yaitu Mr.TyDi dan Miracl, yang digunakan untuk menguji kinerja pemeringkatan teks model pada *dataset* yang berbeda dengan *dataset* pelatihan.

5.7.1 Indo-STS

Dataset STS(*Semantic Textual Similarity*) (Cer et al., 2017) digunakan untuk mengukur tingkat kemiripan antara dua kalimat. Setiap pasangan kalimat diberi label kemiripan antara 0 (tidak mirip) hingga 5 (sangat mirip). Indo-STS merupakan terjemahan dari *dataset* STS dalam bahasa Inggris ke dalam bahasa Indonesia.

5.7.2 Indo-SNLI

Dataset SNLI (Stanford Natural Language Inference) (Bowman et al., 2015) digunakan untuk mengukur kemampuan model dalam memahami hubungan antara dua kalimat. Setiap pasangan kalimat diberi label hubungan antara 2 (kontradiksi), 1(netral), dan

Skor Semantik	Teks 1	Teks 2
5	Sebuah pesawat lepas landas.	Pesawat udara lepas landas.
3.8	Seorang pria memainkan seruling besar.	Seorang pria bermain seruling.
3.8	Seorang pria menyebarkan keju parut di atas pizza.	Seorang pria menyebarkan keju parut di atas pizza mentah.
2.6	Tiga pria bermain catur.	Dua pria bermain catur.
4.25	Seorang pria bermain cello.	Seorang pria yang duduk bermain cello.
4.25	Beberapa pria sedang berjuang.	Dua pria sedang berjuang.
0.5	Seorang pria sedang merokok.	Seorang pria sedang bermain skating.
1.6	Pria itu bermain piano.	Pria itu bermain gitar.

Tabel 1: *Dataset* Indo-STS: Teks 1 dan Teks 2 adalah dua teks yang akan dibandingkan kemiripannya. Skor Semantik adalah nilai kemiripan antara dua teks tersebut dalam skala 0 hingga 5.

0(*entailment*), di mana *entailment* berarti kalimat pertama menyiratkan kalimat kedua, dan kontradiksi berarti kalimat pertama bertentangan dengan kalimat kedua. Indo-SNLI merupakan terjemahan dari *dataset* SNLI dalam bahasa Inggris ke dalam bahasa Indonesia.

5.7.3 Indo-SNLI Triplet

Dengan menggunakan *dataset* Indo-SNLI, dapat dibuat *dataset* triplet (kalimat 1, kalimat 2, kalimat 3) dengan kalimat 1 dan kalimat 2 adalah pasangan kalimat dengan label *entailment*, sedangkan kalimat 3 adalah kalimat dengan label kontradiksi terhadap kalimat 1. *Dataset* ini digunakan untuk melatih model dengan fungsi objektif *triplet loss*.

5.7.4 mMarco Indonesia

mMarco (Bonifacio et al., 2021) adalah versi multibahasa dari *dataset* MSMarco yang diterjemahkan ke berbagai bahasa termasuk Indonesia. mMarco Indonesia merupakan terjemahan dari mMarco dalam bahasa Inggris ke dalam bahasa Indonesia. *Dataset* ini terdiri dari pasangan kalimat pertanyaan dan jawaban yang diambil dari mesin pencari Bing.

5.7.5 Kalimat Paralel Indonesia-Inggris

Dataset kalimat paralel indonesia-inggris merupakan dataset yang sering digunakan dalam pelatihan mesin penerjemah. dataset ini akan digunakan untuk melatih model mBERT dengan prosedur *knowledge distillation*.

Premise (teks 1)	**Hypothesis (teks 2)**	**Label**
"Paduan suara gereja ini bernyanyi kepada massa seraya mereka menyanyikan lagu - lagu gembira dari buku di gereja."	"Gereja memiliki celah-celah di langit-langit."	1 (netral)
"Paduan suara gereja ini bernyanyi kepada massa seraya mereka menyanyikan lagu - lagu gembira dari buku di gereja."	"Gereja penuh dengan lagu."	0 (implikasi)
"Paduan suara gereja ini bernyanyi kepada massa seraya mereka menyanyikan lagu - lagu gembira dari buku di gereja."	"Sebuah paduan suara bernyanyi di pertandingan bisbol."	2 (kontradiksi)
"Seorang wanita dengan headscarf hijau, kemeja biru dan senyum yang sangat besar."	"Wanita itu masih muda."	1 (netral)
"Seorang wanita dengan headscarf hijau, kemeja biru dan senyum yang sangat besar."	"Wanita itu sangat bahagia."	0 (implikasi)
"Seorang wanita dengan headscarf hijau, kemeja biru dan senyum yang sangat besar."	"Wanita itu telah ditembak."	2 (kontradiksi)

Tabel 2: *Dataset* Indo-SNLI: Premise dan Hypothesis adalah dua teks yang akan dibandingkan kemiripannya, dengan Label sebagai label hubungan antara premise dan hypothesis.

5.7.6 Mr.TyDi

Mr.TyDi (Zhang et al., 2021) merupakan sebuah *dataset benchmark* multibahasa yang dibangun berdasarkan *dataset* TyDi (Clark et al.). *Dataset* ini mencakup sebelas bahasa yang memiliki keragaman topik pencarian. Tujuan dari *dataset* ini adalah untuk mengevaluasi kinerja model dalam pemeringkatan teks.

Metrik yang cocok digunakan untuk mengukur performa model pada *dataset* ini adalah *mean reciprocal rank* (MRR), karena pada *dataset* ini, secara kasar, setiap pertanyaan memiliki satu jawaban yang tepat.

5.7.7 Miracl

Miracl (Zhang et al., 2022) adalah *dataset benchmark* multibahasa yang juga digunakan untuk pemeringkatan teks. berbda dengan Mr. TyDi, pada miracl setiap pertanyaan memiliki lebih dari satu jawaban yang tepat yang diurutkan berdasarkan relevansinya dengan pertanyaan.

Pada *dataset* Miracl, metrik yang digunakan untuk mengukur performa model adalah *normalized discounted cumulative gain* (nDCG).

Teks anchor	Teks positif	Teks negatif
"Paduan suara gereja ini bernyanyi kepada massa seraya mereka menyanyikan lagu - lagu gembira dari buku di gereja."	"Gereja penuh dengan lagu."	"Sebuah paduan suara bernyanyi di pertandingan bisbol."
"Seorang pria tua dengan pose paket di depan iklan."	"Seorang pria berpose di depan iklan."	"Seorang pria berjalan dengan iklan."
"Sebuah Landrover sedang didorong menyeberangi sungai."	"Sebuah kendaraan menyeberangi sungai."	"Sedan terjebak di tengah sungai."

Tabel 3: *Dataset* Indo-SNLI Triplet: Teks anchor dibandingkan dengan teks positif dan teks negatif. Teks positif memiliki hubungan entailment dengan teks anchor, sedangkan teks negatif memiliki hubungan kontradiksi dengan teks anchor.

5.8 pelatihan ulang Model

Pada bagian ini, dijelaskan tentang proses pelatihan ulang (*fine tuning*) model pralatih IndoBERT dan mBERT untuk menghasilkan representasi kontekstual yang lebih baik dari kalimat atau dokumen. Pelatihan ulang ini bertujuan untuk meningkatkan performa model dalam pemeringkatan teks.

Fungsi objektif yang digunakan dalam pelatihan ulang model ini tergantung pada jenis *dataset* yang digunakan. Terdapat beberapa fungsi objektif yang digunakan dalam penelitian ini, yaitu *softmax loss* atau *categorical cross entropy loss* (untuk *dataset* Indo-SNLI), *triplet loss* (untuk *dataset* mMarco dan Indo-SNLI *triplet*), dan *mean squared error loss* (untuk *dataset* Indo-STS dan kalimat paralel Indo-Inggris).

5.8.1 Fungsi Objektif Softmax

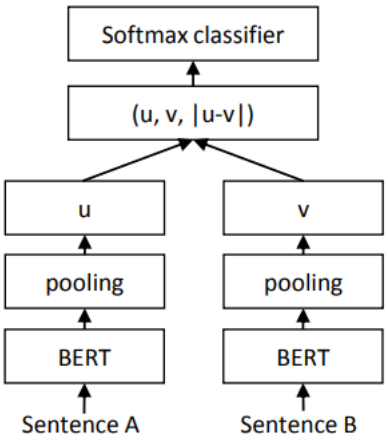
Fungsi objektif *softmax* digunakan pada *dataset* yang terdiri dari pasangan kalimat (kalimat pertama, kalimat kedua, label), dengan label $\in \{0, 1\}$. Arsitektur model yang digunakan adalah *siamese network* yang dilengkapi dengan *softmax classifier* (lihat gambar 7)

Kueri	**Teks positif**	**Teks negatif**
seberapa besar militer kanada?	Angkatan Bersenjata Kanada. 1 Misi penjaga perdamaian Kanada berskala besar pertama dimulai di Mesir pada 24 November 1956. 2 Ada sekitar 65.000 Pasukan Reguler dan 25.000 anggota cadangan di militer Kanada. 3 Di Kanada, 9 Agustus ditetapkan sebagai Hari Penjaga Perdamaian Nasional.	Canadian Physician Health Institute (CPHI) adalah program nasional yang dibuat pada tahun 2012 sebagai kolaborasi antara Canadian Medical Association (CMA), Canadian Medical Foundation (CMF) dan Asosiasi Medis Provinsi dan Teritorial (PTMA).

Tabel 4: *Dataset* mMarco: Kueri merupakan kalimat pertanyaan, teks positif adalah jawaban yang relevan dengan kueri, dan teks negatif adalah jawaban yang tidak relevan dengan kueri.

Teks inggris	**Teks indonesia**
The presence of communication amid scientific minds was equally important to the success of the Manhattan Project as scientific intellect was. The only cloud hanging over the impressive achievement of the atomic researchers and engineers is what their success truly meant; hundreds of thousands of innocent lives obliterated.	Kehadiran komunikasi di tengah pikiran ilmiah sama pentingnya dengan keberhasilan Proyek Manhattan seperti halnya kecerdasan ilmiah. Satu-satunya awan yang menggantung di atas pencapaian mengesankan dari para peneliti dan insinyur atom adalah apa arti kesuksesan mereka yang sebenarnya; ratusan ribu nyawa tak berdosa dilenyapkan.
The illeterate people in Papua were only given the stamp by the leader of the village	orang orang kampung yang tidak bisa membaca hanya diberikan cap saja oleh kepala kampung
Loyalties are fickle	Loyalitas mudah berubah

Tabel 5: *Dataset* kalimat parallel indonesia-inggris



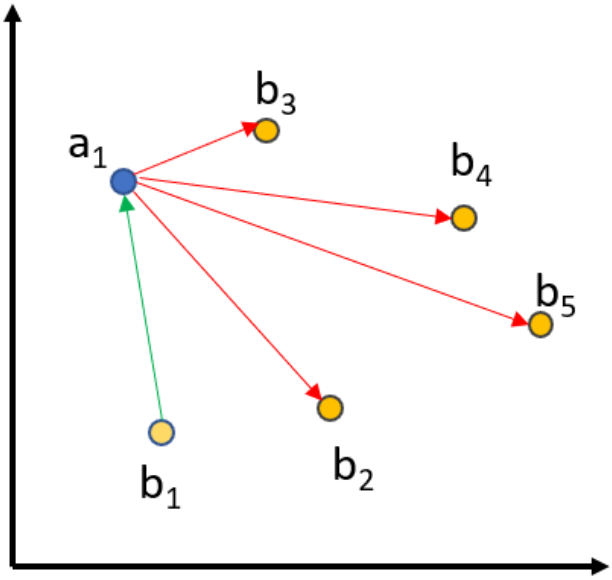
Gambar 7: Ilustrasi arsitektur model untuk fungsi objektif softmax: pasangan (teks 1, teks 2) dimasukkan secara independen ke dalam model BERT. Representasi teks dari kedua input digabungkan dan dimasukkan ke dalam softmax classifier. (Reimers and Gurevych, 2019)

Pada penelitian ini, fungsi objektif *softmax* digunakan pada *dataset* Indo-SNLI. Fungsi objektif *softmax* atau *categorical cross entropy loss* didefinisikan sebagai berikut:

$$loss = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log(p_{ij}) \tag{12}$$

di mana N merupakan jumlah sampel, $C = 3$ merupakan jumlah kelas, y_{ij} adalah label dari sampel ke-i terhadap kelas ke-j, dan p_{ij} adalah probabilitas prediksi model terhadap sampel ke-i terhadap kelas ke-j. Nilai p_{ij} dihitung dengan fungsi *softmax classifier* dari keluaran model IndoBERT.

5.8.2 Fungsi Objektif Triplet



Gambar 8: Ilustrasi fungsi objektif *triplet loss*: untuk pasangan teks yang relevan (a, b_1), tujuannya adalah untuk meminimalkan jarak antara a dan b_1, sehingga jarak tersebut lebih kecil dibandingkan dengan jarak antara a dan b_i yang lain (SBERT.net).

Dalam konteks fungsi objektif *triplet*, terdapat kalimat anchor a, kalimat positif p, dan kalimat negatif n. Fungsi *loss triplet* bertujuan untuk mengurangi jarak antara a dan p (dalam ruang representasi teks) agar lebih kecil dibandingkan dengan jarak antara a dan n. Secara matematis, fungsi *loss* tersebut didefinisikan sebagai berikut:

$$\max\left(|s_a - s_p| - |s_a - s_n| + \epsilon, 0\right) \tag{13}$$

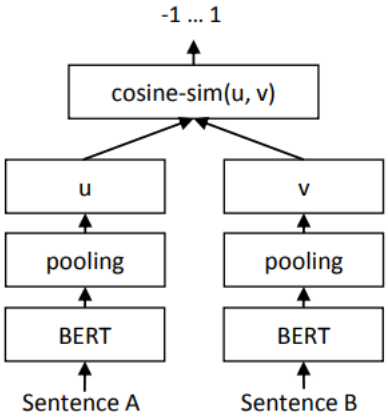
dengan s_x sebagai representasi vektor dari teks x. $|\cdot|$ merupakan metrik jarak dan margin ϵ digunakan untuk memastikan bahwa s_p setidaknya ϵ lebih dekat ke s_a dibandingkan dengan s_n. Dalam penelitian ini, digunakan jarak Euclidean sebagai metrik dan ϵ ditetapkan sebagai 1.

5.8.3 Fungsi Objektif Mean Squared Error

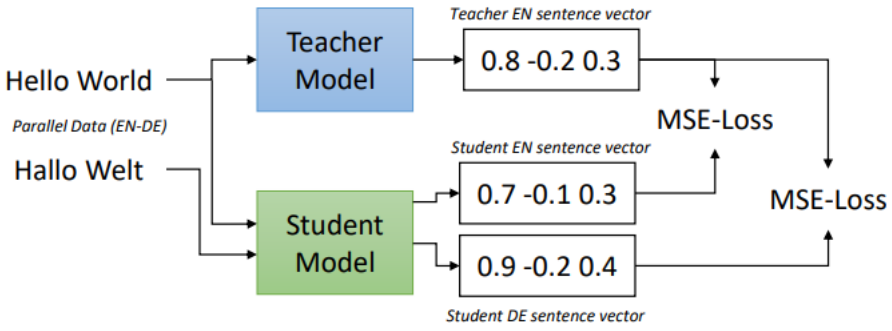
Fungsi objektif *Mean Squared Error* (MSE) digunakan untuk *dataset* indo-STS dan kalimat parallel *indonesia-inggris*. Fungsi objektif MSE didefinisikan sebagai berikut:

$$loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{14}$$

Pada *dataset* indo-STS, $\hat{y}_i$ merupakan prediksi model terhadap sampel ke-i mengenai kemiripan (dalam ruang *cosine*). Arsitektur model yang digunakan adalah *siamese network* dengan *cosine-similarity head* (lihat gambar 9)



Gambar 9: Arsitektur Model untuk Fungsi Objektif *Mean Squared Error* pada ruang *cosine* (Reimers and Gurevych, 2019)



Gambar 10: Diberikan data paralel (misalnya, Bahasa Inggris dan Indonesia), latih *student model* agar vektor yang dihasilkan untuk kalimat dalam Bahasa Inggris dan Indonesia mendekati vektor kalimat dalam Bahasa Inggris dari *teacher model* (Reimers and Gurevych, 2020)

5.8.4 Fungsi Objektif Mean Squared Error (*knowledge distillation*)

Pada *dataset* kalimat parallel antara bahasa Indonesia dan Inggris, dilakukan pelatihan ulang terhadap mBERT (*sebagai student model*) dengan menggunakan fungsi objektif *Mean Squared Error* (MSE) melalui prosedur *knowledge distillation*. Pada prosedur *knowledge distillation*, model yang sudah dilatih untuk tujuan pemeringkatan (*teacher model*) digunakan sebagai acuan. Sebagai contoh, sentence-transformers/msmarco-bert-base-dot-v5 dapat digunakan sebagai *teacher model*. Selanjutnya, model student (mBERT) dilatih ulang untuk meminimalkan fungsi objektif berikut:

$$\frac{1}{|N|}\sum_{j\in N}\left[\left(M\left(s_j\right)-\hat{M}\left(s_j\right)\right)^2+\left(M\left(s_j\right)-\hat{M}\left(t_j\right)\right)^2\right] \tag{15}$$

di mana M adalah *teacher model*, $\hat{M}$ adalah *student model*, s_j adalah kalimat sumber (dalam hal ini bahasa Inggris), t_j adalah kalimat target (dalam hal ini bahasa Indonesia), dan $|N|$ adalah jumlah sampel.

Setelah melalui proses pelatihan dengan menggunakan fungsi objektif *Mean Squared Error* (MSE) dan prosedur *knowledge distillation*, *student model* berhasil memperoleh ruang representasi yang serupa dengan *teacher model*.

6 Langkah kerja

Judul: Pelatihan Ulang Model BERT untuk Representasi Teks yang Lebih Optimal dalam Masalah Pemeringkatan Teks Kata kunci: BERT(Bidirectional Encoder Representations from Transformers), pemeringkatan teks, representasi teks Langkah kerja:

(a) Melakukan studi literatur mengenai BERT dan pemeringkatan teks.

(b) Mengumpulkan *dataset* yang akan digunakan untuk pelatihan ulang model BERT.

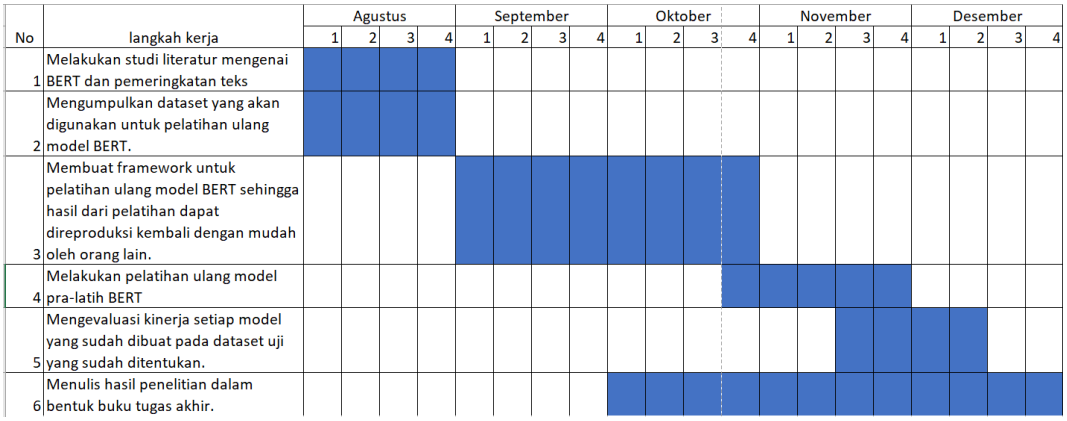
(c) Membuat framework untuk pelatihan ulang model BERT sehingga hasil dari pelatihan dapat direproduksi kembali dengan mudah oleh orang lain.

(d) Melakukan pelatihan ulang model BERT dengan menggunakan *dataset* yang sudah ditentukan

(e) Mengevaluasi kinerja setiap model yang sudah dibuat pada *dataset* uji yang sudah ditentukan.

(f) Menulis hasil penelitian dalam bentuk buku tugas akhir.

7 Jadwal penelitian



Gambar 11: Jadwal kegiatan penelitian tugas akhir

Daftar Pustaka

Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, , Roberto Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of ms marco passage ranking dataset, 2021.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326, 2015. URL `http://arxiv.org/abs/1508.05326`.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055, 2017. URL `http://arxiv.org/abs/1708.00055`.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/W07-1401`.

Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018. URL `http://arxiv.org/abs/1801.06146`.

Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *CoRR*, abs/2010.06467, 2020. URL `https://arxiv.org/abs/2010.06467`.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1301.htmlabs-1301-3781.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018. URL http://arxiv.org/abs/1802.05365.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL http://arxiv.org/abs/1908.10084.

Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.365. URL https:i//aclanthology.org/2020.emnlp-main.365.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.85.

Shijie Wu and Mark Dredze. Are all languages created equal in multilingual bert? *CoRR*, abs/2005.09093, 2020. URL https://arxiv.org/abs/2005.09093.

Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer, and Alexander J. Smola. Neural machine translation with recurrent attention modeling. *CoRR*, abs/1607.05108, 2016. URL http://arxiv.org/abs/1607.05108.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A multi-lingual benchmark for dense retrieval. *arXiv:2108.08787*, 2021.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Making a MIRACL: Multilingual information retrieval across a continuum of languages. *arXiv:2210.09984*, 2022.