X-BERT: Klasifikasi Teks Multi-label eXtreme menggunakan Representasi Encoder Dua Arah dari Transformers

Wei-Cheng Chang1 Hsiang-Fu Yu2 Kai Zhong2 Yiming Yang1 Inderjit Dhillon2,3

1Carnegie Mellon University, 2Amazon, 3University of Texas di Austin

Abstrak

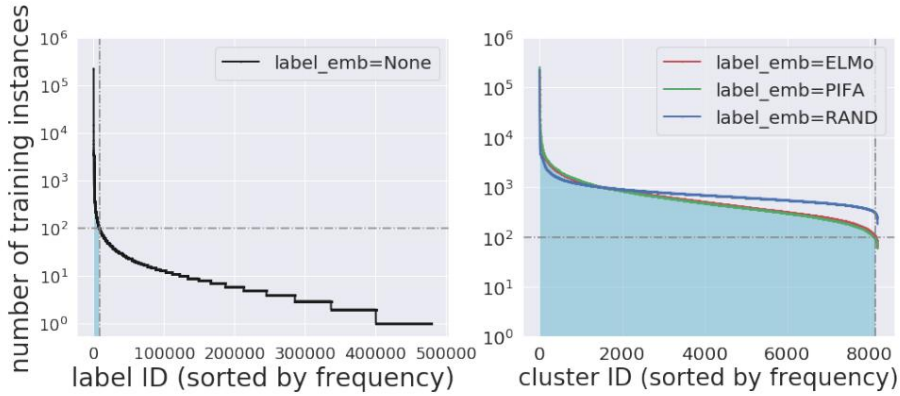
Klasifikasi teks multi-label ekstrim (XMC) menyangkut penandaan teks input dengan label paling relevan dari kumpulan yang sangat besar. Baru-baru ini, model representasi bahasa yang telah dilatih sebelumnya seperti BERT (Bidirectional Encoder Representations from Transformers) telah terbukti mencapai kinerja yang luar biasa pada banyak tugas NLP termasuk klasifikasi kalimat dengan set label kecil (biasanya kurang dari ribuan). Namun, ada beberapa tantangan dalam memperluas BERT ke masalah XMC, seperti (i) kesulitan menangkap ketergantungan atau korelasi antar label, yang fiturnya mungkin berasal dari sumber yang heterogen, dan (ii) kemampuan untuk menskalakan ke pengaturan label yang ekstrem. karena penskalaan bottleneck Softmax secara linier dengan ruang keluaran. Untuk mengatasi tantangan ini, kami mengusulkan X-BERT, solusi skalabel pertama untuk menyempurnakan model BERT pada masalah XMC. Secara khusus, X-BERT memanfaatkan label dan teks input untuk membangun representasi label, yang mendorong kluster label semantik ke dependensi label model yang lebih baik. Inti dari X-BERT adalah prosedur untuk menyempurnakan model BERT untuk menangkap hubungan kontekstual antara teks input dan kluster label yang diinduksi. Akhirnya, ansambel model BERT berbeda yang dilatih pada klaster label heterogen mengarah ke model akhir terbaik kami, yang mengarah ke metode XMC yang canggih. Secara khusus, pada kumpulan data Wiki dengan sekitar 0,5 juta label, presisi@1 X-BERT adalah 67,87%, peningkatan substansial atas fastText baseline neural dan pendekatan XMC canggih Parabel, yang mencapai 32,58% dan presisi 60,91%@1, masing-masing.

1. Perkenalan

Klasifikasi teks multi-label ekstrim (XMC) bertujuan untuk menandai setiap teks yang diberikan dengan subset label yang paling relevan dari koleksi label yang sangat besar, di mana jumlah label bisa mencapai jutaan atau lebih. Baru-baru ini, XMC telah menarik banyak perhatian karena pertumbuhan pesat data skala web di berbagai aplikasi industri, seperti kategorisasi produk untuk e-commerce [1], iklan pencarian dinamis Bing [28, 27], dan penandaan kategori Wikipedia di tantangan Klasifikasi Teks Hirarki Skala Besar PASCAL (LSHTC) [25], untuk menyebutkan beberapa saja.

XMC menghadirkan tantangan komputasi yang besar untuk mengembangkan pengklasifikasi yang efektif dan efisien karena jumlah instans dan label yang ekstrem. Gambar 1 mengilustrasikan contoh distribusi label ekor panjang dari kumpulan data Wiki-500K [31]. Banyak kemajuan telah dibuat untuk mengatasi tantangan skalabilitas dan label sparsity dalam masalah XMC. Sementara pendekatan one-vs-all (OVA) [2, 3] sering mencapai kinerja yang kuat, mereka mengalami masalah skalabilitas. Di sisi lain, metode berbasis pohon [28, 14] mempelajari ansambel pohon klasifikasi yang lemah tetapi cepat, yang bagaimanapun mengarah ke ukuran model yang besar. Pendekatan partisi label [24, 27, 13] membangun pohon label seimbang di mana hanya simpul daun yang dilatih dengan pengklasifikasi satu lawan semua, mencapai akurasi yang sebanding dengan pendekatan OVA tetapi tidak kehilangan kecepatan komputasi. Namun demikian, sebagian besar metode tradisional untuk klasifikasi teks multi-label ekstrim menggunakan varian bag-of-word (BOW) sebagai representasi teks, mengabaikan ketergantungan konteks kata yang lebih tinggi, dan dengan demikian tidak dapat menangkap semantik yang lebih dalam yang ada dalam data teks.

Model pembelajaran mendalam telah diusulkan untuk mempelajari representasi input yang kuat untuk klasifikasi teks [17, 15, 19] serta masalah XMC [20, 37]. Baru-baru ini, komunitas Natural Language Processing (NLP) menyaksikan perubahan paradigma yang dramatis menuju model representasi bahasa dalam yang telah dilatih sebelumnya, yang mencapai state-of-the-art di banyak tugas NLP seperti menjawab pertanyaan, pelabelan peran semantik, penguraian, klasifikasi kalimat. dengan sangat sedikit label, dan banyak lagi. Representasi Encoder Bidirectional dari Transformers (alias BERT [7]) merupakan salah satu perkembangan terbaru dalam bidang pekerjaan ini. BERT mengungguli pendahulunya, ELMo [26] dan GPT [29], melebihi state-of-the-art dengan margin yang lebar pada beberapa tugas NLP. Namun demikian, sulit untuk menyempurnakan model BERT pada tugas XMC. Tantangan utama adalah sulitnya menangkap label



Gambar 1: Di sebelah kiri, Wiki-500K menunjukkan distribusi label yang berekor panjang. Perhatikan bahwa hanya 1,6% dari label yang memiliki lebih dari 100 instans pelatihan, seperti yang ditunjukkan oleh rezim biru sian. Di sebelah kanan adalah distribusi cluster setelah pengindeksan label semantik, dengan representasi label yang berbeda. 99,4% cluster memiliki lebih dari 100 instance pelatihan, yang mengurangi masalah sparity untuk model pelatihan dalam tahap pencocokan.

ketergantungan dari sumber yang heterogen dan kemampuan untuk menskalakan ke pengaturan label ekstrem karena lapisan Softmax tambahan menskalakan secara linier dengan ruang keluaran yang besar.

Dalam makalah ini, kami mengusulkan X-BERT, pendekatan pembelajaran mendalam yang dapat diskalakan untuk menyempurnakan model bahasa pra-terlatih seperti BERT untuk masalah XMC. Sejauh pengetahuan kami, X-BERT adalah model BERT finetuned pertama yang sukses yang mengungguli tolok ukur XMC yang memiliki setengah juta label. Kontribusi dari makalah ini diringkas sebagai berikut:

- Kami mengusulkan X-BERT, model penyesuaian BERT yang dapat diskalakan untuk masalah XMC. X-BERT terdiri dari komponen Semantic Label Indexing, komponen Deep Neural Matching, dan komponen Ensemble Ranking.
- Kami memanfaatkan deskripsi teks label serta kata kunci masukan untuk membangun representasi label yang heterogen, yang menginduksi klaster label semantik. Dengan dependensi label yang dikodekan dalam klaster label, kami menyempurnakan model BERT agar lebih cocok dengan teks input ke satu set klaster label. Akhirnya, ansambel berbagai konfigurasi X-BERT semakin meningkatkan kinerja.
- X-BERT mencapai hasil mutakhir yang baru dibandingkan pendekatan XMC yang ada. Secara kuantitatif, pada dataset Wiki dengan sekitar 0,5 juta label, presisi@1 X-BERT mencapai 67,87%, peningkatan substansial atas fastText baseline deep learning [15] dan pesaing XMC mendekati Parabel [27], yang mencapai 32,58 % dan 60,91% presisi@1, masing-masing. Jumlah ini merupakan peningkatan relatif 11,43% dibandingkan Parabel, yang memang signifikan sejak pendekatan terbaru SLICE [13] dilaporkan mengarah pada peningkatan relatif 5,53%.
- Kumpulan data, kode, dan model pra-pelatihan tersedia untuk umum: https://github.com/OctoberChang/X-BERT.

2 Pekerjaan Terkait

2.1 Klasifikasi Multi-label Ekstrim

Kami mengkategorikan algoritma XMC ke dalam empat kategori: pendekatan satu lawan semua, metode partisi, pendekatan berbasis embedding , dan pendekatan pembelajaran mendalam.

Pendekatan One-Vs-All (OVA). Pendekatan satu lawan semua yang naif memperlakukan setiap label secara independen sebagai masalah klasifikasi biner. Pendekatan OVA [2, 20, 36, 35] telah terbukti mencapai akurasi yang tinggi, tetapi mereka mengalami komputasi yang mahal untuk pelatihan dan prediksi ketika jumlah label sangat besar. Oleh karena itu, beberapa teknik telah diusulkan untuk mempercepat algoritma. PDSparse [36]/PPDSparse [35] memperkenalkan sparsity primal dan dual untuk mempercepat pelatihan serta prediksi. DiSMEC [2], ProXML [3] dan PPDSparse [35] mengeksplorasi paralelisme dan sparsity untuk mempercepat algoritma dan mengurangi ukuran model. Pendekatan OVA juga

banyak digunakan sebagai blok bangunan untuk banyak pendekatan lainnya. Misalnya, dalam Parabel [27] dan SLICE [13], pengklasifikasi OVA linier dengan domain keluaran kecil digunakan.

Metode partisi Pertimbangkan matriks label-instance, Y {0, 1} N×L, di mana N adalah jumlah sampel pelatihan dan L adalah jumlah label. Ada dua cara untuk menggabungkan partisi. Salah satunya adalah mempartisi ruang input (baris Y) dan yang lainnya mempartisi ruang label (kolom Y). Ketika Y sangat jarang, partisi input hanya berisi subset kecil label dan partisi label hanya berisi subset kecil instance. Selanjutnya, dengan menerapkan pendekatan berbasis pohon pada partisi label memungkinkan prediksi waktu sublinier sehubungan dengan ukuran label. Misalnya, Parabel [27] mempartisi label melalui pohon label 2 rata-rata yang seimbang menggunakan fitur label yang dibangun dari instance. Baru-baru ini, beberapa pendekatan diusulkan untuk meningkatkan Parabel.

Bonsai [16] melonggarkan dua kendala utama di Parabel: 1) memungkinkan multiway alih-alih partisi biner dari ruang label di setiap node perantara; 2) menghapus batasan penyeimbangan yang ketat pada partisi. HAXMLNet [38] menggantikan fitur bag-of-words yang jarang digunakan di Parabel dengan representasi saraf dari jaringan perhatian.
SLICE [13] mempertimbangkan untuk membangun grafik perkiraan tetangga terdekat (ANN) untuk mengelompokkan label. Untuk contoh tertentu, label yang relevan dapat dengan cepat ditemukan dari tetangga terdekat dari contoh melalui grafik ANN.

Pendekatan Berbasis Embedding Model embedding [4, 39, 5, 6, 12, 33] menggunakan representasi peringkat rendah untuk matriks label, sehingga pencarian kesamaan untuk label dapat dilakukan dalam ruang dimensi rendah. Dengan kata lain, metode berbasis embedding mengasumsikan bahwa ruang label dapat diwakili oleh ruang laten berdimensi rendah di mana label serupa memiliki representasi laten yang serupa. Untuk mencapai kecepatan komputasi serupa dalam praktiknya, bagaimanapun, model berbasis embedding sering menunjukkan kinerja yang lebih rendah dibandingkan dengan pendekatan one-vs-all yang jarang, seperti PPD-Sparse [35], dan pendekatan partisi seperti Parabel [27], yang mungkin karena inefisiensi struktur representasi label.

Pendekatan Pembelajaran Mendalam Untuk input teks, representasi pembelajaran mendalam diharapkan dapat menangkap informasi semantik dalam input dengan lebih baik daripada fitur bag-of-words, seperti fitur TF-IDF. XML-CNN [20] menggunakan model CNN untuk mewakili input teks, sementara AttentionXML [37] dan HAXMLNet [38] menggunakan model perhatian untuk mengekstrak embeddings dari input teks. SLICE juga menggunakan penyematan pra-pelatihan yang diawasi dari model XML-CNN untuk pelatihan. Model bahasa dalam pra-pelatihan baru-baru ini seperti BERT [7], ELMo [26] dan GPT [29] telah menunjukkan hasil yang menjanjikan pada beberapa tugas NLP. Namun, pekerjaan sebelumnya belum dapat menggabungkan model besar yang telah dilatih sebelumnya untuk XMC, yang menghadirkan tantangan substansial baik dalam pelatihan maupun inferensi.

3 Algoritma yang Diusulkan: X-BERT

Pendekatan kami sebagian terinspirasi dari perspektif pencarian informasi (IR), di mana tujuannya adalah untuk menemukan dokumen yang relevan, dari kumpulan yang sangat besar, untuk kueri tertentu. Untuk menangani sejumlah besar dokumen, mesin IR biasanya melakukan pencarian dalam langkah-langkah berikut [10] — 1) pengindeksan: membangun struktur data yang efisien untuk mengindeks dokumen; 2) pencocokan: temukan indeks dokumen yang dimiliki oleh instance dokumen ini; 3) peringkat: mengurutkan dokumen dalam indeks yang diambil.

Masalah XMC dapat dihubungkan ke masalah IR sebagai berikut: sejumlah besar label dapat dilihat secara analog dengan sejumlah besar dokumen yang diindeks oleh mesin pencari; dan instance yang akan diberi label dapat dilihat sebagai kueri. Karena keberhasilan kerangka kerja tiga tahap IR untuk jumlah target yang sangat besar, beberapa pendekatan yang ada terkait erat dengan kerangka kerja ini, misalnya HAXMLNet dan Parabel. Dalam makalah ini, kami mengusulkan X-BERT (BERT for eXtreme Multi-label Text Classification) di bawah kerangka tiga tahap, yang terdiri dari tahapan berikut:

1. mengindeks label secara semantik,
2. pencocokan indeks label menggunakan pembelajaran mendalam,
3. memberi peringkat label dari indeks yang diambil dan mengambil ansambel konfigurasi yang berbeda dari sebelumnya Langkah.

3.1 Definisi Masalah

Notasi dan Definisi Secara formal, klasifikasi multi-label adalah tugas mempelajari suatu fungsi f yang memetakan suatu , yL] Y = {0, 1} masukan (atau instance) x X ke targetnya y = [y1, y2, · · · label. L, di mana L adalah jumlah unik total di mana (xi , yi) X × {0, 1} L. Kita Asumsikan bahwa kita memiliki satu set N sampel pelatihan {(xi , yi)} $_{saya=1,}^{n}$ gunakan untuk merepresentasikan matriks label. Y {0, 1} N×L, yang baris ke-i-nya adalah $_{saya.}^{>}$ Untuk beberapa kumpulan data khusus, kami memiliki tambahan informasi label y . Misalnya, setiap label dalam kumpulan data Wikipedia [25] diberi nama dengan kata-kata, seperti "Kebun Binatang di Meksiko" dan "Minuman daging babi". Jadi kita akan menggunakan { zj }j=1,2,⋯ ,L ZL sebagai representasi fitur dari label, yang dapat berasal dari informasi label itu sendiri atau dari pendekatan lain.

Perspektif Probabilistik. Kami merumuskan kerangka kerja kami untuk X-BERT dalam perspektif probabilistik. Asumsikan setelah pengindeksan, kami memiliki K cluster label, {Ik}k=1,2,⋯K, di mana setiap Ik adalah subset dari indeks label, yaitu, Ik [L]. Untuk contoh tertentu x, probabilitas label ke-l yl relevan dengan x adalah P(yl |x). Kita dapat membentuk model probabilistik sebagai berikut:

$$P(il\,|x) = \sum_{k=1}^{K} P(il\,\|lk, x, r)P(Ik|x, m). \tag{1}$$

Di sini P(Ik|x, m) adalah model yang cocok dengan m sebagai parameternya dan P(yl ‖lk, x, r) adalah model peringkat dengan r sebagai parameternya. Untuk model peringkat, kami memaksakan itu

$$P(yl = 1\|lk, x, r) = 0, \text{ jika } l \not\in Ik,$$

yaitu, selama tahap pemeringkatan, hanya label dalam cluster yang diambil yang dipertimbangkan. Intuisinya adalah ketika ukuran label sangat besar, akan ada banyak label serupa yang dapat dikelompokkan. Kerangka kerja kami memiliki keuntungan sebagai berikut:

1. Berbekal representasi label yang heterogen, kami mem-bootstrap model pencocokan dan pemeringkatan dengan berbagai indeks pengelompokan I, yang mengarah ke model pencocokan/pemeringkatan yang lebih beragam untuk sistem pengambilan yang lebih baik.
2. Didorong oleh ruang cluster yang diinduksi namun kompak, 12 lapisan BERT sekarang layak untuk komputasi opsi sebagai realisasi dari model pencocokan P(Ik|x, m).
3. Membatasi peringkat ke kumpulan label yang lebih kecil membantu mengecualikan label yang tidak relevan jika pengelompokan dan model yang cocok cukup baik.

Kami sekarang secara singkat menyentuh masing-masing tahap ini.

3.2 Pengindeksan Label Semantik

Mengindeks dokumen di mesin pencari membutuhkan informasi teks yang kaya sementara label XMC biasanya tidak memiliki informasi ini. Jadi, kami bertujuan untuk menemukan representasi label yang bermakna untuk membangun sistem pengindeksan semantik seperti itu.

Penyematan label melalui teks label Alih-alih menggunakan ID label, kita memerlukan beberapa informasi semantik tentang label. Diberikan informasi teks tentang label, seperti deskripsi singkat kategori dalam kumpulan data Wikipedia, kita dapat menggunakan teks singkat ini untuk mewakili label. Dalam karya ini, kami menggunakan salah satu representasi kata tercanggih, ELMo [26], untuk mewakili kata-kata dalam label. Perhatikan bahwa tanpa finetuning dengan kerugian yang tepat, penyematan token BERT atau varian lain mungkin tidak cocok untuk masalah pengelompokan. Penyematan label dibuat dengan cara menyatukan semua penyematan kata dalam teks label. Secara khusus, asumsikan urutan kata untuk label ke-l adalah {w1, ..., wk}, penyematan label untuk label ke-l adalah zl = ELMo(wt) di mana ELMo(wt) adalah representasi kata kontekstual dari berat $_{1}^{-}{}_{k}Pk$ t=1

Penyematan label melalui kata kunci dari contoh positif Namun, informasi teks pendek untuk label mungkin tidak berisi informasi yang cukup dan beberapa kata dalam teks pendek mungkin ambigu dan berisik. Oleh karena itu, kami mempertimbangkan representasi label lain yang berasal dari penyisipan teks sparse dari instance. Secara khusus, penyematan label zl adalah jumlah fitur TF-IDF sparse dari semua instance yang relevan untuk label ke-l:

$$z_l = v_l / \|v_l\|_k, \quad v_l = \sum_{i:y_{il}=1} \text{TF-IDF}(x_i),\ l = 1, \ldots, L,$$

Kami menyebut jenis penyematan label ini sebagai Agregasi Fitur Instance Positif, singkatan sebagai PIFA, yang juga digunakan dalam beberapa metode XMC yang canggih [27, 13, 38, 16].

Pengindeksan label Dengan representasi label di atas, kami membangun sistem pengindeksan dengan mengelompokkan label seperti pada metode partisi label [27, 13, 38, 16]. Untuk kesederhanaan, kami menganggap k-means clustering yang seimbang [22, 27] sebagai pengaturan default. Karena kurangnya representasi langsung dan informatif dari label, sistem pengindeksan untuk XMC mungkin berisik dibandingkan dengan masalah IR. Untungnya, contoh di XMC biasanya sangat informatif. Oleh karena itu, kami dapat memanfaatkan informasi yang kaya dari instans untuk membangun sistem pencocokan yang kuat serta ranker yang kuat untuk mengimbangi sistem pengindeksan.

3.3 Pencocokan Saraf Dalam

Fase pencocokan untuk XMC adalah untuk menetapkan cluster yang relevan (yaitu, indeks) untuk setiap contoh, yang direduksi menjadi masalah klasifikasi multi-label (MLC) lainnya. Kunci untuk mesin pencari yang sukses adalah model pencocokan ingatan tinggi karena fase peringkat berikutnya didasarkan pada dokumen yang diambil dari fase pencocokan. Untuk membangun sistem pencocokan MLC yang kuat, kami bertujuan untuk mengekstrak informasi diskriminatif dalam teks masukan. Banyak model pembelajaran mendalam telah diusulkan untuk masalah MLC seperti Seq2Seq [23], CNN [17] dan model self-attention [19, 32, 37] untuk mengekstrak informasi berurutan dalam teks input. Namun, model pembelajaran mendalam menderita kompleksitas komputasi yang tinggi dan sulit untuk diskalakan untuk XMC. Untungnya, di X-BERT, jumlah cluster dapat dikontrol oleh praktisi, sehingga kami dapat mengatur skala masalah pencocokan MLC sehingga model pembelajaran mendalam kami masih menikmati pelatihan yang wajar dan waktu inferensi.

Setelah pengelompokan label, label dipartisi menjadi K cluster $\{I_k\}_{k=1}^{K}$ dimana $I_k \subset [L]$. Tahap pencocokan saraf dalam bertujuan untuk menemukan encoder yang kuat g untuk membuat instance yang menyematkan $u = g(x)$, dan mempelajari jaringan saraf dangkal yang memetakan instance yang menyematkan u ke kluster yang relevan di $\{I_k\}_{k=1}^{K}$. Konkretnya, cluster I_k relevan dengan instance x_i jika instance memiliki label positif di I_k (yaitu, $y_{il} = 1, l \in I_k$).

3.3.1 X-perhatian

Model saraf dalam telah menunjukkan kesuksesan besar dalam banyak aplikasi NLP, seperti jaringan konvolusi yang terjaga keamanannya untuk pembelajaran urutan [9], mekanisme perhatian-diri untuk klasifikasi teks [19, 37], serta model Transformer dan variannya untuk terjemahan mesin [32]. Jadi, pertama-tama kita pertimbangkan mekanisme self-attention [19, 32] sebagai realisasi untuk encoder g(·), maka nama X-ttention.

Secara khusus, diberikan contoh token T , direpresentasikan sebagai urutan penyisipan kata $D = \{w_1, \ldots, w_T\}$, kami mempertimbangkan BiLSTM [11] untuk mengekstrak dependensi tingkat tinggi dalam teks: $H = (h_1, \ldots, h_T)$, $h_t = (\overrightarrow{h}_t, \overleftarrow{h}_t)$, di mana embedding $H \in \mathbb{R}^{2m \times T}$ dan $\overrightarrow{h}_t, \overleftarrow{h}_t$ adalah keadaan tersembunyi dari LSTM dua arah untuk token t. Untuk memiliki tetap ukuran H untuk instance panjang variabel, X-ttention mempelajari bobot self-attention untuk menggabungkan secara linear keadaan tersembunyi T di H. Secara konkret, mekanisme self-attention mengambil H sebagai input, dan mengeluarkan vektor dari bobot a: $a = \text{softmax}(w_{s2} \tanh(W_{s1} H))$, Perhatian multi-kepala mempertimbangkan beberapa vektor perhatian, $A = \text{softmax}(W_{s2} \tanh(W_{s1} H))$, $A \in \mathbb{R}^{r \times T}$ dengan r adalah jumlah kepala perhatian. Akhirnya, instance menyematkan $u \in \mathbb{R}^{2rm}$ menjadi $u = g(x; W_{s2}, W_{s1}, \text{BiLSTM}) = \text{vec}(HA^T)$.

3.3.2 X-BERT

Baru-baru ini, komunitas NLP telah menyaksikan perubahan paradigma dramatis dari arsitektur saraf khusus tugas ke model representasi bahasa mendalam yang telah dilatih sebelumnya. Di bawah paradigma ini, jaringan saraf pertama-tama dilatih sebelumnya pada sejumlah besar teks di bawah tujuan yang tidak diawasi dan kemudian disesuaikan dengan data khusus tugas, yang mencapai state-of-the-art di banyak tugas pemahaman bahasa alami seperti pertanyaan menjawab, pelabelan peran semantik , penguraian, klasifikasi kalimat dengan sangat sedikit label, dan banyak lagi. Representasi Encoder Dua Arah

Himpunan data	Ntrn	Nval	Ntst #fitur	#labels	#labels/instances	#instances/label	#clusters
Eurlex-4K	3.865 33.246 13.7945	1.544			5,32	19,93	64
Wiki10-28K	5.732 99.919 128.2659	1.251			18,68	7,47	512
AmazonCat-13K	1.067.616 118.623 306.782 161.925 13.234 1.411.760 156.396				5,04	406,77	256
Wiki-500K	676.730 517.631 479.315				4,90	14,44	8192

Tabel 1: Statistik Data. Ntrn, Nval, Ntst mengacu pada jumlah instance dalam pelatihan, validasi, dan set pengujian, masing-masing.

dari Transformers (alias BERT [7]) berdiri sebagai perkembangan terbaru dalam arah ini yang secara signifikan mengungguli banyak pendahulunya seperti Generative Pretrained Transformer (GPT) [29] dan Embeddings from Language Model (ELMo) [26].

Dalam makalah ini, kami mengusulkan untuk menyempurnakan model BERT sebagai encoder g(·) untuk masalah pencocokan XMC, oleh karena itu namanya X-BERT. Sejauh pengetahuan kami, BERT belum dieksplorasi untuk masalah XMC yang telah ratusan ribu label atau lebih. Mengikuti pengaturan [7], kita mulai dengan model BERT yang telah dilatih sebelumnya dengan: 12 lapisan sel Transformer dan ambil status tersembunyi terakhir dari token [CLS] sebagai instance embedding u R m. Selama fine-tuning, kami mengoptimalkan model end-to-end menggunakan Adam dengan pemanasan pada tingkat pembelajaran, dengan parameter pengklasifikasi linier tambahan W R K×m, diikuti oleh kehilangan engsel persegi sebagai fungsi kerugian.

Sementara kami menganggap model BERT sebagai contoh dari kerangka yang diusulkan, penting untuk dicatat bahwa X-BERT dapat mengakomodasi model pra-pelatihan lanjutan lainnya seperti XLNet [34] dan Roberta [21], yang kami tinggalkan eksplorasi masa depan.

3.4 Peringkat Ensemble

Setelah langkah pencocokan, subset kecil dari klaster label diambil dan tugas yang tersisa adalah memberi peringkat pada label cluster ini. Sebagai model peringkat, tujuan kami adalah untuk memodelkan relevansi antara instance dan label yang diambil. Secara formal, diberi label l dan instance x, kita ingin mencari pemetaan h(x, l) yang memetakan fitur instance x dan label l untuk skor. Dalam makalah ini, kami terutama menggunakan pendekatan linear one-vs-all (OVA), yang merupakan salah satu yang paling model langsung dan berkinerja terbaik. Model ini memperlakukan penugasan label individu ke sebuah instance sebagai masalah klasifikasi biner independen. Label kelas positif jika instance milik cluster; jika tidak, itu negatif. Jika fitur instance adalah teks, input ke pengklasifikasi linier dapat berupa fitur tf-idf. Dengan skor probabilitas yang dihitung melalui (1), kami selanjutnya menggabungkan skor dari model X-BERT yang berbeda, yaitu dilatih pada kluster label sadar semantik yang berbeda dengan menggunakan penyematan ELMo atau PIFA.

4 Hasil Empiris

4.1 Kumpulan Data dan Prapemrosesan

Kami mempertimbangkan empat set data klasifikasi teks multi-label dari Repositori Klasifikasi Ekstrim yang tersedia untuk umum [31] yang kami akses ke representasi teks mentah, yaitu Eurlex-4K, Wiki10-28K, AmazonCat 13K dan Wiki-500K. Ringkasan statistik dari kumpulan data diberikan pada Tabel 1. Kami mengikuti pelatihan dan pemisahan pengujian dari [31] dan menyisihkan 10% dari contoh pelatihan sebagai set validasi untuk penyetelan hyperparameter.

Kami mencatat bahwa statistik data dan jumlah label pada Tabel 1 sedikit berbeda dari [31] karena dua alasan. Pertama, karena hanya judul teks isi yang disediakan di Wiki10-28K dan Wiki-500K, kami memetakan judul dengan database dump Wikipedia terbaru, dan ekstrak teks mentah dokumen. Ini menciptakan subset dari aslinya dataset, menghasilkan jumlah label yang sedikit lebih kecil. Kedua, kami mematuhi prosedur pra-pemrosesan teks [23], mengganti nomor dengan token khusus; membangun kosakata uni-gram untuk TF-IDF; dan memotong dokumen setelah 300 kata.

Kami juga mempertimbangkan kumpulan data E-niaga eksklusif1, yaitu Prod2Query-1M, yang memetakan judul produk untuk pertanyaan pelanggan yang relevan. Prod2Query-1M terdiri dari 14 juta instans (produk) dan 1 juta label (kueri) di mana labelnya positif jika suatu produk diklik setidaknya sekali oleh kueri penelusuran pelanggan yang sesuai ke labelnya. Kami membagi dataset menjadi 12,5 juta sampel pelatihan, 0,8 juta sampel validasi, dan 0,7 juta sampel pengujian.

[1] "E-commerce" adalah pengganti untuk menjaga anonimitas selama periode peninjauan. Detailnya akan kami sertakan nanti.

Himpunan data	metode	Sebelum@1	Sebelumnya@3	Sebelumnya@5	Ingat@1	Ingat@3	Ingat@5
Eurlex-4K	PD-Jarang [36]	79,97	66,74	55,50	16.45	40.18	54.66
	teks cepat [15]	73,97	62,25	51,97	15.07	37.30	51.05
	FastXML [28]	76.17	61.86	50,75	15.54	37.01	49.75
	Parabel [27]	82.48	69,95	58,49	16,87	41,98	57.46
	X-BERT	**86.00**	**74,52**	**62,64**	**17,63**	**44,83**	**61.59**
Wiki10-28K	PD-Jarang [36]	82.12	71.00	60.47	5.04	12.86	18.04
	teks cepat [15]	65,28	53,48	45,36	3.97	9,62	13.42
	FastXML [28]	83,20	68,68	58,39	5.03	12,28	17.13
	Parabel [27]	82,78	71,70	62,48	5.02	12,88	18.46
	X-BERT	**85,75**	**75,19**	**65,13**	**5.24**	**13,57**	**19.24**
AmazonCat-13K	PD-Jarang [36]	89,18	69,95	55,46	25,44	54,72	67.55
	teks cepat [15]	81,56	70,65	58,35	22,81	54,36	69.91
	FastXML [28]	92.68	77.17	62.05	26.44	58.70	73.18
	Parabel [27]	91,42	76,34	61,68	25,82	57,84	72.53
	X-BERT	**95,17**	**80,65**	**65,19**	**27,15**	**61,15**	**76,32**
Wiki-500K	PD-Jarang [36]	-	-	-	-	-	-
	teks cepat [15]	32,58	23.00	18.60	10.67	19,89	25.30
	FastXML [28]	43,46	29.03	22.12	12.30	21,87	26.32
	Parabel [27]	60,91	41.33	31.67	18.74	33,21	39.75
	X-BERT	**67,87**	**46.73**	**35.97**	**21.21**	**37,70**	**45.20**

Tabel 2: Perbandingan Keseluruhan X-BERT atas metode canggih. Semua membandingkan baseline disajikan dalam ini Tabel dijalankan kembali pada rangkaian kereta/tes benchmark kami untuk perbandingan yang adil. Perhatikan bahwa kami menunda diskusi tentang membandingkan X-BERT ke metode XMC mutakhir yang lebih baru pada Tabel 4.

4.2 Algoritma dan Hyperparameter

Membandingkan Metode. Kami sekarang membandingkan X-BERT yang diusulkan dengan metode XMC canggih termasuk metode partisi input FastXML [28], metode partisi label Parabel [27], pendekatan berbasis OVA PD-Sparse [36], dan fastText model deep learning representatif [15] pada benchmark yang tersedia untuk umum kumpulan data multi-label [31]. Yang terpenting, semua hasil evaluasi metode pada Tabel 2 diperoleh dengan menjalankannya kode yang tersedia pada partisi dataset benchmark kami. Untuk evaluasi yang lebih komprehensif dengan state-of-the-art lainnya pendekatan yang belum merilis kodenya atau sulit untuk direproduksi, kami memiliki perbandingan terperinci dalam Tabel 4.

Metrik Evaluasi. Kami mengikuti [7] untuk mendapatkan representasi teks tokenized WordPiece untuk X-BERT dan menggunakan TF-IDF unigram untuk metode berbasis fitur (PD-Sparse, FastXML, dan Parabel). Kami mengevaluasi semua metode dengan ukuran peringkat berbasis contoh termasuk Precision@k (k = 1, 3, 5) dan Recall@k (k = 1, 3, 5), yang banyak digunakan dalam literatur XMC [28, 4, 14, 36, 27, 30].

hiperparameter. Untuk X-BERT, semua hyperparameter dipilih dari set validasi yang ditahan. Nomor cluster tercantum dalam Tabel 1, yang konsisten dengan pengaturan Parabel untuk perbandingan yang adil. Pencocokan saraf dari X-BERT adalah konfigurasi model pra-latihan BERTbase yang tidak terbungkus, dan menggunakan Adam [18] sebagai pengoptimal dengan kecepatan belajar dipilih dari $\{5 \times 10^{-5}, 8 \times 10^{-5}, 10^{-4}\}$. Hyperparameters dari metode lain diatur dengan mengikuti: pengaturan default mereka. Secara khusus, jumlah pohon di FastXML adalah T = 100, dan contoh maksimum dalam daun simpul adalah m = 10. Untuk Parabel, jumlah pohon adalah T = 1, 2, 3, dan jumlah maksimum label dalam simpul daun adalah m = 100. Baik FastXML dan Parabel menggunakan C = 1 sebagai penalti kerugian untuk SVM linier dengan engsel kuadrat kerugian, seperti yang diimplementasikan melalui LIBLINEAR [8]. Untuk PD-Sparse, suku regularisasi = 0,01, dan maksimum jumlah iterasi adalah 20. Untuk fasttext baseline deep learning, kami menetapkan kecepatan pembelajaran menjadi 1, jumlah unit tersembunyi hingga 100, dan jumlah maksimum epoch hingga 1000.

Himpunan data	ID konfigurasi	Konfigurasi Ablasi pengindeksan pencocokan peringkat	#trees	Metrik Evaluasi Prec@1	Prec@3	Prec@5	Recall@1	Recall@3	Recall@5
AmazonCat-13K	0	ELMo,PIFA BERT linier 6		**95.17**	**80** %		**27,15**	**61,15**	**76,32**
	1	ELMo,PIFA BERT linier 2					26,95	60,71	75.61
	2	ELMo BERT linier 2					26,89	60,66	75.51
	3	PIFA BERT linier 2					26,84	60,42	75.24
	4	ELMo BERT linier	1				26,54	59,71	74.08
	5	PIFA BERT linier	1				26,54	59,90	74,39
	6	PIFA BERT tf-idf	1				15,65	35,05	46.04
	7	ELMo Xttention linier	1				26,11	58,36	72.56
	8	PIFA Xttention linier	1				26,13	58,58	72.93
	9	ELMo linier linier	1				25,27	56,34	70.37
	10	PIFA linier linier					25,62	57,53	71,95
	11	Linier linier acak	1 1				24,99	55,32	68.84
Wiki-500K	0	ELMo,PIFA BERT linier 6		**67,87**	**46,73**	35,97 66,29	**21,21**	**37,70**	**45.20**
	1	ELMo,PIFA BERT linier 2		45,31	34,79	65,56 44,66	20,63	36,42	43.60
	2	ELMo BERT linier 2		34,17	65,45	44,75 34,38	20,27	35,76	42.71
	3	PIFA BERT linier 2		63,74	43,07	32,80 64,01	20,28	35,90	43.01
	4	ELMo BERT linier	1	43,48	33,26	46,47 27,33	19,61	34,32	40.80
	5	PIFA BERT linier	1	19,79	62,59	42,15 31,95	19,74	34,74	41.44
	6	PIFA BERT tf-idf	1	62,99	42,67	32,63 54,14	15,19	23,45	26.70
	7	ELMo Xttention linier	1	36,21	17,40,01		19,14	33,39	39,53
	8	PIFA Xttention linier	1				19,35	34,00	40.60
	9	ELMo linier linier	1				16,32	29,22	35.02
	10	PIFA linier linier	1				18,12	32,17	38.62
	11	Linier linier acak	1				11,35	18,99	22.00

Tabel 3: Studi ablasi X-BERT pada kumpulan data AmazonCat-13K dan Wiki-500K. Kami menguraikan empat take away pesan: (1) Indeks= {1, 2, 3} menunjukkan kinerja yang lebih baik dalam menggabungkan penyematan label yang berbeda; (2) Indeks= {4, 7, 9} dan Indeks= {5, 8, 10} keduanya menunjukkan bahwa, dari segi kinerja, transformator dalam yang telah dilatih sebelumnya > BiLSTM+perhatian diri > model linier (yaitu Parabel); (3) Indeks={5, 6} nyatakan pentingnya dan pembelajaran peringkat bukan peringkat TF-IDF tanpa pengawasan. (4) Indeks={10, 11} menunjukkan pentingnya label semantik pengindeksan alih-alih pengelompokan acak.

Eurlex-4K				Wiki-500K			
metode	Sumber	Peningkatan Relatif di atas Parabel (%) Sebelum@1	Sebelum@3 Sebelum@5	metode	Sumber	Peningkatan Relatif di atas Parabel (%) Sebelum@1	Sebelum@3 Sebelum@5
X-BERT	Meja 2	**+4.27%**	**+6.53%** **+7.10%**	X-BERT	Meja 2	**+11.43%**	**+13.07%** **+13.58%**
MENGIRIS	[13, Tabel 2]	+4.27%	+3.34% +3.11%	MENGIRIS	[13, Tabel 2]	+5,53%	+7,02% + 7,56%
AttentionXML	[37, Tabel 2]	+0,91%	+2,09% +0,86%	HAXMLNet	[38, Tabel 3]	+2,40%	+5,75% + 6,42%
ProXML	[3, Tabel 5]	+3,86%	+2,90% +2,43%	ProXML	[3, Tabel 5]	+2.22%	+0.82% + 2.92%
bonsai	[16, Tabel 2]	+0,97%	+1,46% +1,57%	bonsai	[16, Tabel 2]	+0,73%	+0,40% + 0,52%
PPD-Jarang	[27, Tabel 2]	+1,92%	+2.93% +2.92%	PPD-Jarang	[27, Tabel 2]	+2.39%	+2.33% + 2.88%
DiSMEC	[27, Tabel 2]	+1,73%	+2,90% +2,80%	DiSMEC	[27, Tabel 2]	+2,45%	+2,39% + 2,98%
PfastreXML	[27, Tabel 2]	-8,27%	-8,75% -8,73%	PfastreXML	[27, Tabel 2]	-13,13%	-18,58% -20,31%
XML-CNN	[27, Tabel 2]	-7.14%	-8,59% -10,64%	XML-CNN	[27, Tabel 2]	-12,65%	-20,52% -22,67%
fastText	fastText.zip	-3.04%	-4.59% -5.11%	-46.51%	-44.35%	-41.27%	Meja 2
SLEEC	[16, Tabel 2]	-3.53%	-6.40% -9.04%	SLEEC	[16, Tabel 2]	-29.84%	-40.73% -45.08%

Tabel 4: Perbandingan Peningkatan Relatif atas Parabel. Peningkatan relatif untuk setiap state-of-the-art (SOTA) metode dihitung berdasarkan metrik yang dilaporkan dari makalah aslinya seperti yang ditunjukkan di kolom Sumber. X-BERT yang diusulkan mengungguli semua pendekatan SOTA lainnya pada dua set data benchmark yang umum digunakan: Eurlex-4K dan Wiki-500K.

4.3 Hasil Empiris

Dalam subbagian ini, pertama-tama kami membandingkan pendekatan X-BERT yang diusulkan dengan metode XMC canggih, dan menyajikan studi ablasi rinci X-BERT.

4.3.1 Perbandingan Keseluruhan.

Tabel 2 membandingkan X-BERT yang diusulkan dengan baseline XMC kuat lainnya pada empat set data benchmark. Perhatikan lagi bahwa di sini kami hanya menyajikan hasil evaluasi metode XMC yang memiliki kode yang tersedia dan kinerja yang dapat direproduksi seperti yang dilaporkan dalam makalah mereka ketika kami menjalankan implementasinya pada partisi dataset benchmark kami. X BERT mengungguli model SOTA XMC Parabel pada semua set data. Perlu dicatat bahwa, pada kumpulan data paling menantang Wiki-500K, X-BERT meningkat di atas Parabel sekitar 7%/4% peningkatan mutlak untuk presisi@1 dan presisi@5. Keuntungan signifikan ini berasal dari dua teknik baru: deep neural matcher dan ansambel berbagai representasi label semantik. Dibandingkan dengan fastText dasar saraf yang terkenal, X-BERT mencapai kinerja yang jauh lebih baik, meskipun dengan mengorbankan waktu pelatihan yang lebih lama.

4.3.2 Hasil pada Dataset Prod2Query-1M .

Kami menerapkan model X-BERT ke sistem rekomendasi kata kunci untuk kampanye iklan e-commence, di mana pengiklan menawar kata kunci untuk produk mereka. Daftar kata kunci yang direkomendasikan ditampilkan dalam kelompok 25 per halaman. Jadi kami menghitung presisi pada 25, 50 dan 100. Kami menunjukkan peningkatan relatif X-BERT atas metode produksi saat ini pada Tabel 5.

Metrik	Sebelum@25	Sebelum@50	Sebelum@100
Peningkatan Relatif	+37,16%	+25,25%	+10,26%

Tabel 5: Peningkatan relatif X-BERT selama metode produksi saat ini pada dataset e-commence.

4.3.3 Studi Ablasi.

Kami dengan hati-hati melakukan studi ablasi X-BERT seperti yang ditunjukkan pada Tabel 3. Kami menguraikan kerangka kerja X-BERT menjadi tiga tahap: pengindeksan, pencocokan, dan peringkat, seperti yang diperkenalkan pada Tabel 3. Indeks konfigurasi 0 mewakili konfigurasi terbaik akhir sebagai dilaporkan pada Tabel 2. Ada empat pesan yang dapat diambil dari studi ablasi ini, dan kami menjelaskannya dalam empat paragraf berikut.

Konfigurasi id 0 hingga 3 menunjukkan efek penggabungan representasi label yang berbeda dalam tahap pengindeksan. Manfaat dari model ensembling dengan random seed yang berbeda dapat dilihat dari konfigurasi id 0 sampai 1. Selain itu, dari id 1 sampai 2 dan 3, kami mengamati bahwa ensembling menggunakan heterogeneous label embeddings lebih efektif daripada menggunakan embedding label yang sama dari random yang berbeda. seed, yang merupakan teknik yang digunakan dalam model Parabel. Ini menegaskan bahwa keragaman representasi label semantik membantu pencocokan saraf dalam menangkap label relevan yang berbeda.

Selanjutnya, kami menganalisis bagaimana model pencocokan yang berbeda mempengaruhi kinerja, seperti yang ditunjukkan pada konfigurasi id 4, 5, 7, 8, 9, 10. Jelas bahwa BERT yang di-finetuned (id 4, 5) lebih kuat daripada self-attention model (id 7, 8), dan model self-attention lebih efektif dibandingkan dengan model linear hierarkis (id 9, 10) yang digunakan dalam Parabel.

Pentingnya melatih model parametrik dalam tahap pemeringkatan ditekankan pada konfigurasi id 5, 6. TF-IDF pada dasarnya memberi peringkat ulang label dalam cluster yang diambil dengan pencocokan kata dengan dokumen yang ditanyakan tanpa mempelajari model apa pun. Kami melihat penurunan kinerja yang cukup besar dari pelatihan model linier ke pencocokan kata TF-IDF di tahap peringkat ulang. Temuan ini menunjukkan pentingnya mempelajari model parametrik yang lebih kuat di tahap peringkat; di masa depan kami berencana untuk melampaui model linear ke model saraf untuk pemeringkatan.

Akhirnya, kami mengkonfirmasi perlunya penyematan label dan algoritme pengelompokan dengan membandingkan dengan heuristik penugasan pengelompokan acak, seperti yang ditunjukkan pada id 10, 11. Sementara penugasan acak menghasilkan lebih banyak kluster yang terdistribusi secara merata (Lihat Gambar 1), ia gagal membentuk kluster yang bermakna. cluster, yang menimbulkan kesulitan dalam pencocokan dan tahap peringkat.

4.4 Perbandingan Lintas-Kertas

Banyak pendekatan XMC baru-baru ini diusulkan. Meskipun sebagian besar berisi perbandingan empiris pada beberapa kumpulan data yang umum digunakan, seperti Eurlex-4K dan Wiki-500K, metrik evaluasi dari metode yang sama pada "kumpulan data yang sama" bervariasi dari kertas ke kertas. Misalnya, presisi@1 DiSMEC terdaftar sebagai 63,70% di [13, Tabel 2] dan 70,20% di [16, Tabel 2]. Demikian pula, presisi@1 Parabel di Wiki-500K terdaftar sebagai 59,34% di [13, Tabel 2], 68,52% di [27, Tabel 2], sementara kita melihat 60,91% seperti yang terlihat pada Tabel 2. Kemungkinan inkonsistensi jatuh tempo

perbedaan dalam data preprocessing, data split, atau hyper-parameter. Dengan demikian, sulit untuk membandingkan metrik langsung dari makalah yang berbeda.

Di sini kami mengusulkan pendekatan untuk mengkalibrasi angka-angka ini sehingga berbagai metode dapat dibandingkan dengan cara yang lebih berprinsip. Secara khusus, untuk setiap metrik m(·), kami menggunakan peningkatan relatif atas metode jangkar umum, yang diatur menjadi Parabel karena banyak digunakan dalam literatur. Kemudian untuk metode X yang bersaing dengan metrik m(X) pada kumpulan data yang dilaporkan dalam makalah, kita dapat menghitung peningkatan relatif terhadap Parabel sebagai berikut: m(X)ÿm(Parabel) × 100%, di mana m(Parabel) adalah metrik yang diperoleh Parabel pada kumpulan data yang sama dalam m(Parabel) kertas yang sama. Mengikuti pendekatan di atas, kami menyertakan berbagai pendekatan XMC dalam perbandingan kami. kami melaporkan peningkatan relatif dari berbagai metode pada dua kumpulan data yang umum digunakan, Eurlex-4K dan Wiki-500K, pada Tabel 4.

Dari tabel ini, kita dapat dengan jelas mengamati bahwa X-BERT membawa peningkatan paling signifikan atas Parabel.

5. Kesimpulan

Dalam makalah ini, kami mengusulkan X-BERT, pendekatan pembelajaran mendalam pertama dengan model BERT yang disempurnakan yang mencapai kinerja canggih dalam masalah XMC. Tahap pengindeksan label semantik baru memberikan partisi label heterogen yang mem-bootstrap berbagai model BERT, menghasilkan model ansambel yang kuat untuk masalah XMC. Secara kuantitatif, pada dataset Wiki-500K, presisi@1 meningkat dari 60,91% menjadi 67,87% saat membandingkan X-BERT dengan metode XMC kuat Parabel. Jumlah ini merupakan peningkatan relatif 11,43% atas Parabel, yang memang signifikan dibandingkan dengan pendekatan mutakhir SLICE yang memiliki peningkatan relatif 5,53% dibandingkan Parabel.

Referensi

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, dan Manik Varma. Pembelajaran multi-label dengan jutaan label: Merekomendasikan frasa tawaran pengiklan untuk halaman web. Di WWW, halaman 13–24. ACM, 2013.

[2] Rohit Babbar dan Bernhard Scholkopf. DiSMEC: mesin sparse terdistribusi untuk klasifikasi multi-label yang ekstrim . Di WSDM, 2017.

[3] Rohit Babbar dan Bernhard Scholkopf. Kelangkaan data, ketahanan, dan klasifikasi multi-label yang ekstrem. Pembelajaran Mesin, 2019.

[4] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, dan Prateek Jain. Penyematan lokal yang jarang untuk klasifikasi multi-label yang ekstrem. Dalam NIPS, 2015.

[5] Yao-Nan Chen dan Hsuan-Tien Lin. Pengurangan dimensi ruang label yang sadar fitur untuk klasifikasi multi-label tion. Dalam NIPS, 2012.

[6] Moustapha M Cisse, Nicolas Usunier, Thierry Artieres, dan Patrick Gallinari. Filter mekar yang kuat untuk tugas klasifikasi multilabel besar. Dalam NIPS, 2013.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, dan Kristina Toutanova. BERT: Pra-pelatihan dua arah yang dalam transformer untuk pemahaman bahasa. Di NAACL, 2019.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, dan Chih-Jen Lin. LIBLINEAR: Sebuah perpustakaan untuk klasifikasi linier besar. Jurnal penelitian pembelajaran mesin, 9 (Agustus): 1871–1874, 2008.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, dan Yann N Dauphin. Urutan konvolusi ke pembelajaran berurutan. Dalam ICML, 2017.

[10] Google. Cara kerja pencarian. https://www.google.com/search/howsearchworks/, 2019. Diakses: 2019-1-18.

[11] Alex Graves dan Jurgen Schmidhuber. Klasifikasi fonem bingkai dengan lstm dua arah dan lainnya arsitektur jaringan saraf. Neural Networks, 18(5-6):602–610, 2005.

[12] Daniel J Hsu, Sham M Kakade, John Langford, dan Tong Zhang. Prediksi multi-label melalui penginderaan terkompresi. Dalam NIPS, 2009.

[13] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, dan Manik Varma. Slice: Linear yang dapat diskalakan pengklasifikasi ekstrim dilatih pada 100 juta label untuk pencarian terkait. Di WSDM, 2019.

[14] Himanshu Jain, Yashoteja Prabhu, dan Manik Varma. Fungsi kehilangan multi-label yang ekstrem untuk rekomendasi, penandaan, peringkat & aplikasi label lain yang hilang. Di KD, 2016.

[15] Armand Joulin, Makam Edouard, Piotr Bojanowski, dan Tomas Mikolov. Tas trik untuk klasifikasi teks yang efisien. Di EACL, 2017.

[16] Sujay Khandagale, Han Xiao, dan Rohit Babbar. Bonsai-beragam dan pohon yang dangkal untuk multi-label yang ekstrim klasifikasi. pracetak arXiv arXiv:1904.08249, 2019.

[17] Yoon Kim. Jaringan saraf convolutional untuk klasifikasi kalimat. Dalam EMNLP, 2014.

[18] Diederik Kingma dan Jimmy Ba. Adam: Sebuah metode untuk optimasi stokastik. Dalam ICLR, 2014.

[19] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, dan Yoshua Bengio. Penyematan kalimat perhatian-diri yang terstruktur. Dalam ICLR, 2017.

[20] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, dan Yiming Yang. Pembelajaran mendalam untuk klasifikasi teks multi-label yang ekstrem. Dalam SIGIR, 2017.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, dan Veselin Stoyanov. RoBERTa: Pendekatan pra-pelatihan bert yang dioptimalkan dengan kuat. pracetak arXiv arXiv:1907.11692, 2019.

[22] Mikko I Malinen dan Pasi Franti. K-means yang seimbang untuk pengelompokan. Dalam Lokakarya Internasional Bersama IAPR, 2014.

[23] Jinseok Nam, Eneldo Loza Menca, Hyunwoo J Kim, dan Johannes Furnkranz. Memaksimalkan akurasi subset dengan jaringan saraf berulang dalam klasifikasi multi-label. Dalam NIPS, 2017.

[24] Alexandru Niculescu-Mizil dan Ehsan Abbasnejad. Filter label untuk klasifikasi multilabel skala besar. Di AIST, 2017.

[25] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, dan Patrick Galinari. LSHTC: Tolok ukur untuk klasifikasi teks skala besar. pracetak arXiv arXiv:1503.08581, 2015.

[26] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, dan Luke Zettlemoyer. Representasi kata yang dikontekstualisasikan secara mendalam. Di NAACL, 2018.

[27] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, dan Manik Varma. Parabel: Pohon label yang dipartisi untuk klasifikasi ekstrim dengan aplikasi untuk iklan pencarian dinamis. Di WWW, 2018.

[28] Yashoteja Prabhu dan Manik Varma. FastXML: Pengklasifikasi pohon yang cepat, akurat, dan stabil untuk ekstrem pembelajaran multi-label. Di KDD, 2014.

[29] Alec Radford, Karthik Narasimhan, Tim Salimans, dan Ilya Sutskever. Meningkatkan pemahaman bahasa dengan pra-pelatihan generatif. 2018.

[30] Sashank J Reddi, Satyen Kale, Felix Yu, Dan Holtmann-Rice, Jiecao Chen, dan Sanjiv Kumar. stokastik penambangan negatif untuk pembelajaran dengan ruang keluaran yang besar. Di AISTATS, 2019.

[31] Manik Varma. Repositori klasifikasi ekstrim: Dataset & kode multi-label, 2018. Diakses: 10-05-2018.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, dan Illia Polosukhin. Perhatian adalah semua yang Anda butuhkan. Dalam NIPS, 2017.

[33] Jason Weston, Samy Bengio, dan Nicolas Usunier. WSABIE: Meningkatkan anotasi gambar kosakata yang besar. 2011.

[34] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, dan Quoc V Le. XLNet: Prapelatihan autoregresif umum untuk pemahaman bahasa. pracetak arXiv arXiv:1906.08237, 2019.

[35] Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, dan Eric Xing. PPDSparse: A metode paralel primal-dual sparse untuk klasifikasi ekstrim. Di KD, 2017.

[36] Ian EH Yen, Xiangru Huang, Kai Zhong, Pradeep Ravikumar, dan Inderjit S Dhillon. PD-Sparse: Pendekatan primal dan dual sparse untuk klasifikasi multikelas dan multilabel ekstrim. Dalam ICML, 2016.

[37] Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, dan Shanfeng Zhu. AttentionXML: Klasifikasi teks multi-label ekstrim dengan jaringan saraf berulang berbasis perhatian multi-label. pracetak arXiv arXiv:1811.01727v1, 2018.

[38] Ronghui You, Zihan Zhang, Suyang Dai, dan Shanfeng Zhu. HAXMLNet: Jaringan perhatian hierarkis untuk klasifikasi teks multi-label yang ekstrem. pracetak arXiv arXiv:1904.12578, 2019.

[39] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, dan Inderjit Dhillon. Pembelajaran multi-label skala besar dengan label yang hilang. Di ICML, 2014.