PROYEK AKHIR: PETUNJUK TAHAP 1

Tim Asdos NLP

Pengolahan Bahasa Manusia - Semester Gasal 2024/2025

1 Hal yang Harus Dilakukan

Pada tahap ini, Anda perlu menjalankan kode dari topik yang Anda pilih. Pastikan bahwa semuanya bisa dijalankan tanpa masalah. Buatlah laporan berisikan penjelasan terkait keberhasilan menjalankan program. Sertakan *screenshot* sebagai buktinya. Gunakan format penamaan *file* sebagai berikut: *NLP-PA1–[topik]-[NamaLengkap].pdf* (misalnya *NLP-PA1-A-AndiBudi.pdf*). Selanjutnya, akan dijelaskan apa yang perlu dijalankan untuk masing-masing topik.

2 Mentor dari Setiap Topik

Dalam pengerjaan PA, setiap topik memiliki asdos PJ sebagai mentor. Anda bisa bertanya ke PJ topik Anda, dan mereka yang akan melakukan penilaian laporan dokumen setiap tahap PA. Berikut adalah mentor dari masing-masing topik:

- Topik A dan E: Luthfi Balaka
- Topik B dan F: Faisal Adi
- Topik C dan G: Alvin Xavier
- Topik D: Fadli Aulawi

3 Topik A: *Melatih Relational Word Embeddings*

Akses *repository* ini. Lalu, ikuti petunjuk pada README-nya. Tahapannya meliputi:

- Unduh model *embedding* dan *file* terkait (misal *dataset*) menggunakan `downloader.ipynb`.
- Latih *Relational Word Embedding* menggunakan `train-rwe.ipynb`. Anda perlu mengunduh hasilnya untuk lanjut ke tahap berikutnya. Perhatikan bahwa *embedding* yang dilatih ini menggunakan data yang berbeda dari yang di-*paper*,[1] meskipun Anda tetap mendapatkan yang aslinya dari *paper* (*file* `reference_rwe.txt` yang diunduh dari tahap sebelumnya).
- Lakukan klasifikasi menggunakan `classification.ipynb`. Perhatikan bahwa terdapat beberapa skenario eksperimen pada *file* ini.

Perhatikan bahwa Anda perlu mengemas program tersebut sedemikian sehingga bisa dijalankan pada *platform* yang Anda gunakan. Misalnya jika menggunakan Google Colab, semua *file* bisa dijadikan satu file Jupyter Notebook (ipynb).

4 Topik B: *Document-Level Relation Extraction*

Secara umum, pada tahap 1 ini Anda diharapkan melakukan dua hal, yakni memahami paper [1] serta menjalankan kode pada Github repository ini.

[1] https://www.kaggle.com/datasets/mikeortman/wikipedia-sentences

Tahap memahami *paper* tersebut diharapkan sudah dilakukan sebelum terlibat langsung pada program/model ini. Akan sangat menantang bagi Anda untuk langsung *hands-on* pada *source code* tanpa memahami terlebih dahulu alur program/model secara umum.

Tahap berikutnya adalah menjalankan *source code* tersebut sesuai dengan arahan yang terdapat pada README yang tersedia. Perlu diperhatikan bahwa pada README terdapat CUDA sebagai prasyarat untuk menjalankan program tersebut. Apabila Anda **tidak memiliki CUDA** di *hardware* atau layanan yang Anda sewa, Anda diperkenankan untuk melakukan **modifikasi** pada *source code* sedemikian sehingga program dapat berjalan dengan normal.

Tahap terakhir adalah Anda melaporkan aktivitas memahami paper serta eksplorasi *source code* sedemikian sehingga dapat berjalan dan melakukan fungsi *training*, *evaluation*, *testing* dengan normal. Teknis laporan harap mengacu pada point 1.

5 Topik C: *Unsupervised Document Classification via Learning from Neighbors*

Ada baiknya memahami terlebih dulu apa yang dikerjakan oleh paper [2] untuk mempermudah percobaan. Apabila ingin melihat versi lebih singkatnya bisa mengunjungi artikel berikut. *Source code* dari *paper* tersebut dapat diakses melalui tautan ini. Untuk mempermudah, dataset yang digunakan adalah AG's news topic classification, bisa diambil dari sini

Sebelum melakukan percobaan, jangan lupa meng-*install* package yang diperlukan sesuai requirements.txt. Pengerjaan dibebaskan menggunakan python atau jupyter notebook, asalkan dapat memberikan bukti *screenshot* untuk masing-masing *task*-nya. Untuk tahap ini, ada beberapa hal yang perlu dilakukan, yaitu:

- Buat 2 buah model yang berbeda:
  - Lbl2Vec, yaitu model *document classification* yang berbasis Doc2Vec
  - Lbl2TransformerVec, yang merupakan model *document classification* berbasis *sentence transformer*.
- Lakukan *training* terhadap kedua model tadi menggunakan dataset AG's news topic classification bagian train, lalu coba prediksi data trainingnya juga. Dari sini, akan dihasilkan suatu output berupa *similarity score* dari tiap dokumen terhadap masing-masing label topik
- Uji coba model yang sudah dilatih tadi dengan data testing (data yang tidak dipakai untuk *training*). Ini juga menghasilkan output yang sama, yaitu *similarity* dari tiap dokumen dan topik.
- Hitung performa di kedua *case* tersebut dengan metrik F1 Score (Task ini dapat dianggap sebagai klasifikasi *multiclass* dengan membandingkan *most similar label* dan *ground truth label*).

6 Topik D: *Dependency Parsing*

Silakan akses repository melalui tautan berikut, lalu baca dan pahami panduan menjalankan project ini, terutama tahapan-tahapan dari persiapan data, pelatihan model, hingga evaluasi.

Pada tahap pertama ini, Anda diminta mengumpulkan data *dependency tree* untuk proses pelatihan. Pastikan Anda memahami struktur dan format datanya. Lakukan pengumpulan dan pemrosesan data mengikuti script `go_data.sh` dan `prepare_data.sh`. Pahami setiap tahapan script tersebut dan pastikan prosesnya dapat dijalankan sampai selesai. Hasilnya, Anda akan mendapat direktori data dalam format CoNNL-U.

Karena script ini digunakan beberapa tahun kebelakang, datanya mungkin belum terbarui, khususnya dataset Bahasa Jawa. Anda dapat memperbarui sumber dataset dengan menggunakan sumber lainnya pada script atau mengunduh dataset Bahasa Jawa langsung dari repository berikut.

Output minimal tahapan ini adalah Anda dapat memproses data *dependency parser* secara utuh dan membuat laporannya. Namun **sangat disarankan** untuk mulai mengerjakan tahap berikutnya seperti setting environment dan mencoba proses training karena kompleksitasnya yang lebih tinggi.

7 Topik E: *Argument-Pair Extraction dengan Framework Machine Reading Comprehension*

Dengan menggunakan kode dari *repository* ini, Anda akan melatih dan menguji model untuk melakukan ekstraksi pasangan argumen dari teks. Setiap langkah yang perlu dilakukan sudah dijelaskan pada *file* README, tetapi formatnya mayoritas berupa *file* Python. Anda perlu menyesuaikannya dengan *platform* yang Anda gunakan. Sebagai contoh, jika Anda menggunakan Google Colab, Anda bisa mengubah *file-file* tersebut menjadi satu *file* Jupyter Notebook (ipynb).

8 Topik F: *Automated Concatenation of Embeddings for Structured Prediction*

Pada tahap 1 ini terdapat dua hal yang akan Anda lakukan, di antaranya memahami paper [3] serta menjalankan program yang terdapat pada repository Github ini.

Pada paper [3] terdapat beberapa topik yang telah dipelajari di kelas, seperti *dependency parsing*, *Named Entity Relation (NER)*, *Part-of-Speech Tagging (POS tag)*, serta *word embedding*. Mengingat Anda diasumsikan sudah familiar dengan topik-topik tersebut, maka saat memahami paper [3] diharapkan Anda tidak akan kesulitan.

Tantangan pada tahap 1 ini terdapat pada pemahaman format dataset *sequence labeling* NER serta corpus untuk *dependency parsing*. Anda diharapkan benar-benar memahami serta melakukan eksplorasi menyeluruh baik pada README maupun masing-masing *command* yang terdapat pada repo ini. Terdapat dua skenario yang ditawarkan oleh repo tersebut, yakni *reproduce* hasil pada paper yang terdapat pada README ini serta *train*

from scratch seperti disampaikan pada README ini. Kedua skenario tersebut wajib Anda eksplorasi sebagai syarat minimum pada tahap 1 ini.

Apabila pada *hardware* atau layanan *notebook* yang Anda sewa tidak terdapat/menyediakan CUDA, Anda diperkenankan untuk melakukan modifikasi pada *source code* tersebut sedemikian sehingga mampu berjalan tanpa kendala. Laporkan hasil eksplorasi Anda sesuai dengan instruksi pada point 1.

9 Topik G: *SimCSE: Simple Contrastive Learning of Sentence Embeddings*

Sebagai tahap awal, akan dilakukan training terhadap model SimCSE. *Source code* dapat diakses pada tautan berikut. Jika ingin menjalankan di local, disarankan untuk membuat *virtual environment* terlebih dulu. Opsi lain adalah menggunakan colab atau kaggle agar dapat memanfaatkan GPU mereka. Ada beberapa *package* yang perlu di-*install* sebagaimana tercantum di bagian *readme*-nya:

- Install pytorch 1.7.1 terlebih dulu
- Install library di requirements.txt

Setelah melakukan setup *environment* dan *packages*, dapat dilanjutkan ke tahap menjalankan program:

- Unduh dataset yang telah disediakan. Ini dapat dilihat pada script data/download_nli.sh.
- Latih model dengan menjalankan *script* run_sup_example.sh (ini sebenarnya hanya meng-*execute* kode train.py saja). Akan ada 2 *task* yang dijalankan, yaitu *train* dan *eval*. Jika dirasa sangat berat untuk melatih model menggunakan keseluruhan dataset, ambil sebagian kecil saja.
- Perlu diingat bahwa *script* tersebut secara *default* menjalankan program .py menggunakan 'cuda' sebagai device. Karena itu, dipersilakan melakukan perubahan pada kode agar dapat menyesuaikan spesifikasi *hardware* masing-masing. Untuk menjalankan di colab atau kaggle, perlu memindahkan kode yang ada di file .py ke .ipynb.

References

[1] Y. Ma, A. Wang, and N. Okazaki, "DREEAM: Guiding attention with evidence for improving document-level relation extraction," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1971–1983. [Online]. Available: https://aclanthology.org/2023.eacl-main.145

[2] T. Schopf., D. Braun., and F. Matthes., "Lbl2vec: An embedding-based approach for unsupervised document retrieval on predefined topics," in *Proceedings of*

the 17th International Conference on Web Information Systems and Technologies - WEBIST, INSTICC. SciTePress, 2021, pp. 124–132. [Online]. Available: https://www.scitepress.org/Papers/2021/107103/107103.pdf

[3] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu, "Automated concatenation of embeddings for structured prediction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 2643–2660. [Online]. Available: https://aclanthology.org/2021.acl-long.206