PROYEK AKHIR: PETUNJUK TAHAP 2

Tim Asdos NLP

Pengolahan Bahasa Manusia - Semester Gasal 2024/2025

Selamat! Anda sudah berhasil menjalan program untuk topik Anda. Selanjutnya, Anda akan diminta untuk mengulang uji coba yang dikerjakan di *paper* tersebut dan melaporkannya di bagian **Hasil Uji Coba dan Analisis** pada laporan PA Anda.

1 Hal yang Harus Dilakukan

Pada tahap ini, Anda perlu menjalankan eksperimen sesuai dengan skenario dari topik Anda yang akan dijelaskan di bagian berikutnya. Hasil eksperimen perlu Anda jabarkan di bagian **Hasil Uji Coba dan Analisis**. Berikut adalah informasi yang perlu ada di bagian tersebut:

1. **Metriks Evaluasi.** Jelaskan setiap metriks evaluasi yang digunakan, seperti Precision, Recall, atau F1-score. Tuliskan rumusnya dan berikan referensi terkait rumus tersebut.
2. **Dataset.** Sebutkan informasi yang relevan tentang dataset pada percobaan Anda. Contohnya adalah ukurannya (jumlah kalimat, jumlah token, dll), bahasa yang digunakan, referensi ke *paper* yang membuat dataset tersebut, dsb. Salah satu cara menyampaikan informasinya adalah dalam bentuk tabel.
3. **Skenario Eksperimen.** Sebutkan *requirements* yang dibutuhkan (*hardware* dan/atau *software*) yang diperlukan untuk menjalankan program. Untuk pendekatan *deep learning*, perlu juga disebutkan informasi *hyperparameter* seperti *batch size*, *epoch*, dsb.
4. **Hasil.** Tuliskan hasil percobaan di bagian ini. Anda bisa tampilkan hasil percobaan dengan visualisasi seperti tabel atau gambar. Jelaskan informasi/*insights* apa yang bisa didapatkan dari hasil tersebut. Jangan meminta pembaca untuk memikirkan sendiri kesimpulan apa yang bisa didapatkan.
5. **Diskusi.** Pada bagian ini, Anda diharapkan membahas kenapa suatu skenario mempunyai hasil yang lebih baik dibandingkan skenario lainnya. Anda juga diharapkan bisa menemukan kelemahan dari program yang Anda uji. Akan lebih jika bagian ini diisi dengan *error analysis*. Input seperti apa saja yang masih belum berhasil diproses dengan baik?

Kumpulkan laporan Anda dengan format penamaan *file* sebagai berikut: *NLP-PA2–[topik]-[NamaLengkap].pdf* (misalnya *NLP-PA2-A-AndiBudi.pdf*). Penilaian akan dilakukan **hanya** berdasarkan apa yang Anda tulis pada bagian **Hasil Uji Coba dan Analisis** dan referensi yang digunakan di bagian ini. Bagian lain yang tidak terkait agar "disembunyikan" dengan cara dijadikan *comment* pada file LaTeX. Selanjutnya, akan dijelaskan skenario eksperimen yang perlu dijalankan untuk masing-masing topik.

2 Skenario Eksperimen

2.1 Topik A: *Melatih Relational Word Embeddings*

Dari tahap sebelumnya, diharapkan Anda sudah mulai memahami *flow* dari eksperimen topik ini. Anda melatih model *relational word embeddings* (RWE) dengan `train-rwe.ipynb`, kemudian digunakan untuk klasifikasi dengan `classification.ipynb`.

Sekarang, Anda akan melatih model RWE dengan *hyperparameters* berbeda sehingga menghasilkan beberapa model RWE. Kemudian, Anda akan bandingkan beberapa skenario klasifikasi, baik yang melibatkan model RWE maupun tidak. Berikut adalah detail hal-hal yang perlu Anda lakukan.

2.1.1 Pelatihan Model RWE

Perhatikan pada fungsi `train\_rwe`, terdapat beberapa *hyperparameters* yang bisa diatur. Silakan Anda coba tiga skenario dalam melatih model RWE:

- **Skenario A1**: *Default Hyperparameters*
- **Skenario A2**: Skenario A1 dengan ubahan jumlah *epochs*
- **Skenario A3**: Skenario A1 degnan ubahan *learning rate*

Satu hal yang perlu dicatat adalah Anda melatih model RWE berbasiskan dataset Wikipedia yang berbeda dengan yang ada di *paper*. (Dalam proses pelatihan, dataset tersebut direpresentasikan sebagai `rel_embeddings_path.txt`.) Secara lebih spesifik, dataset yang digunakan sebagai basis berasal dari Kaggle.[1]. Meskipun sama-sama dari Wikipedia di tahun 2018, terdapat perbedaan dengan yang ada di *paper*. Meskipun demikian, pada `downloader.ipynb`, Anda tetap mendapatkan model RWE *pre-trained* seperti **Skenario A1** bawaan *paper* (nama *file*-nya `reference_rwe.txt`) untuk komparasi.

2.1.2 Klasifikasi Pasangan Kata

Untuk klasifikasi, Anda akan membandingkan akurasi yang didapatkan dari beberapa skenario berikut:

Word Embedding	**RWE**
FastText	*None*
	Skenario A1
	Skenario A2
	Skenario A3
	Bawaan *paper*
BGE-Base	*None*
	Skenario A1
	Skenario A2
	Skenario A3
	Bawaan *paper*

Table 1: Skenario Klasifikasi

Selamat bereksperimen dan pastikan untuk menggali *insights* dari hasil yang didapatkan. *Hint*: Generalisasi pelatihan RWE dengan *dataset* Wikipedia berbeda dan akurasi dengan dan tanpa RWE.

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

2.2 Topik B: *Document-Level Relation Extraction*

Objektif pada tahap kedua ini adalah melakukan eksperimen serta memberikan analisis terhadap eksperimen yang Anda lakukan. Terdapat dua pilihan yang dapat Anda lakukan, yakni

- Melakukan *training* dalam *fully-supervised setting* hingga mendapatkan hasil akhir berupa metrik evaluasi. Namun, pada pilihan ini Anda dipersilakan untuk mengatur ukuran data yang akan dilatih hingga diuji. Anda tidak diharuskan menggunakan seluruh data yang tersedia pada dataset.
- Anda melakukan *inference* terhadap *checkpoint* model yang telah disediakan oleh peneliti di bagian ini atau Google Drive. Harapan keluaran dari pilihan ini adalah Anda mampu melakukan *inference* sedemikian sehingga mengeluarkan hasil akhir berupa metrik evaluasi seperti yang dilaporkan pada *paper* [1].

Apabila Anda terkendala dengan kemampuan *hardware* atau hal lainnya, Anda dipersilakan untuk melakukan penyesuaian seperti implementasi *early stopping*, *batch processing*, *error tolerance*, atau hal lainnya. Namun, perlu dijadikan catatan bahwa terdapat ukuran minimal dataset agar mampu memberikan hasil akhir yang informatif. Dengan demikian, apabila ingin melakukan penyesuaian jumlah dataset, perlu dilakukan lebih bijak dan berhati-hati.

Extra Point: Bagi Anda yang senang eksplorasi dan/atau tantangan, silakan dieksplor kemungkinan dilakukannya *transfer learning* dengan memanfaatkan dataset yang tersedia di internet. Analisislah apakah dengan menerapkan *transfer learning* tersebut mampu meningkatkan metrik evaluasi atau tidak.

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

Setelah melalui tahap 1, alur kerja program secara umum diharapkan sudah bisa dipahami. Model Lbl2Vec berbasis Doc2Vec dan sentence transformers telah berhasil di-train dan digunakan untuk memprediksi dokumen (*testing*) dengan `model.predict_model_docs()`. Perlu diingat juga bahwa parameter pada tahap tersebut juga tidak dibatasi (masih dibebaskan).

Pada eksperimen kali ini, yang perlu dilakukan adalah menguji nilai-nilai parameter yang berbeda untuk melakukan improvisasi terhadap modelnya. Sebagai *baseline*, gunakan nilai-nilai paremeter berikut:

- `similarity_threshold = 0.30`
- `min_num_docs = 100`
- `epochs = 10`

Ekspektasi Eksperimen:

1. **Tujuan**
   Cobalah mengubah parameter model untuk meningkatkan performa klasifikasi dokumen dibandingkan dengan model *baseline*. Gunakan **f1-score** sebagai metrik evaluasi untuk mengukur performa model, dan bandingkan hasilnya dengan baseline.

2. **Dokumentasi Eksperimen**

   - Jelaskan setiap parameter yang diubah (misalnya `similarity_threshold`, `min_num_docs`, atau `epochs`), alasan Anda mengubahnya, dan bagaimana perubahan tersebut diharapkan dapat meningkatkan *f1-score*.
   - Jika perubahan parameter belum berhasil melampaui performa *baseline* setelah beberapa kali mencoba, tidak masalah. Cukup tuliskan parameter yang digunakan dan berikan penjelasan mengapa performanya belum meningkat.

Catatan: Tidak ada batasan khusus untuk nilai-nilai parameter yang diuji. Namun, pastikan setiap percobaan terdokumentasi dengan baik untuk membantu evaluasi dan analisis performa model.

2.4 Topik D: *Dependency Parsing*

Pada tahap sebelumnya, anda telah menyiapkan dataset *dependency tree* untuk keperluan eksperimen ini. Dataset tersebut akan digunakan dalam proses pelatihan (*training*) untuk melatih model dengan struktur kalimat pada tiap bahasa tertentu. Tahapan selanjutnya yang perlu dilakukan adalah sebagai berikut:

1. Mempersiapkan *environment* untuk melaksanakan proses pelatihan. Panduan tahapan ini telah tersedia dalam berkas `README.md`. Apabila versi Python dan *library* yang tercantum sudah tidak terbarui (*outdated*), anda dapat melakukan eksplorasi untuk mengimplementasikan versi yang lebih baru.

2. Menjalankan script pelatihan. Script-script tersebut dapat ditemukan dalam direktori `examples/run_more/go_train_*.sh`. Seluruh script memiliki alur (flow) yang identik, dengan perbedaan hanya pada bahasa yang digunakan sebagai target pelatihan.

Luaran (output) dari tahapan ini adalah terlaksananya satu proses pelatihan pada skenario spesifik menggunakan dataset Bahasa Jawa. Model yang dihasilkan akan dijadikan sebagai model acuan (baseline) untuk evaluasi dan model dasar untuk transfer learning pada tahap selanjutnya. Hasil dari proses pelatihan ini berupa berkas checkpoint dengan ekstensi *.pt.

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

Dari tahap sebelumnya, diharapkan bahwa Anda sudah lebih memahami *flow* dari eksperimen yang dilakukan. Secara lebih spesifik, kita melatih model berbasiskan Machine Reading Comprehension (MRC) pada dataset *RR-Submission-v2*.[2]. Namun, proses pelatihan sangat lama (sekitar 2.5 jam per-*epoch*) sehingga dataset dibuat lebih kecil (setengah dari ukuran asal).

Yang perlu Anda lakukan adalah bereksperimen dengan mengubah beberapa *hyperparameters* dari proses pelatihan model. (Anda bisa mengatur *hyperparameters* melalui *file* `config.py`.) Anda perlu memilih suatu nilai `epochs` dan `layers` yang berbeda dengan nilai

[2] https://github.com/LiyingCheng95/ArgumentPairExtraction/tree/master/data/RR-submission-v2

default-nya. Sebagai ilustrasi, misalkan Anda memilih nilai *epochs* 4 (*default*-nya 2) dan *layers* 3 (*default*-nya 2), maka Anda akan memiliki skenario eksperimen di Tabel 2.

Skenario	**Epochs**	**Layers**
A	2 (*default*)	2 (*default*)
B	2 (*default*)	3
C	4	2 (*default*)
D	4	3

Table 2: Skenario Eksperimen

Sebagai catatan, pastikan bahwa nilai *hyperparameters* lain tetap (cukup ubah `epochs` dan `layers`). (Anda diperbolehkan juga mencoba skenario lain di luar yang wajib.) Setelah melakukan eksperimen, Anda bisa menggali *insights* dari hasil yang didapatkan. Berikut adalah pertanyaan yang wajib Anda jawab dari hasil yang didapatkan:

- Apakah hasil yang didapatkan dengan dataset *training* dikecilkan masih *comparable* dengan saat menggunakan semuanya (bisa dilihat pada *paper*)? Sebagai referensi, waktu yang dibutuhkan untuk satu *epoch* pada dataset ukuran asal adalah 2.5 jam pada Kaggle—perhatikan adanya *trade-off* yang diambil.
- Bagaimana pengaruh ubahan *hyperparameters* yang dilakukan terhadap nilai F1-score yang didapatkan? Apakah masuk akal dengan makna dari ubahan yang dilakukan?

Tentunya banyak hal menarik lainnya yang bisa digali dari hasil yang didapatkan. Selamat bereksperimen!

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

Setelah melalui tahap pertama, Anda diharapkan telah memahami alur keseluruhan program serta cara kerja model tersebut. Pada tahap kedua ini, Anda diharapkan mampu melakukan eksperimen serta analisis terhadap hasil yang didapatkan. Terdapat dua pilihan yang dapat Anda lakukan, di antaranya,

- Melakukan inferensi pada model yang telah disediakan oleh peneliti pada bagian ini. Pada pilihan ini, Anda **diharuskan** melakukan inferensi pada konfigurasi yang disarankan pada paper [2]. Laporkan hasil yang Anda terhadap keempat bahasa yang tersedia, yakni *de*, *en*, *es*, serta *nl*.
- Melakukan reproduksi model secara manual dengan mengikuti langkah-langkah yang tersedia pada markdown ini. Pada pilihan ini, Anda dipersilakan untuk menyesuaikan ukuran dataset yang diproses dengan kemampuan *hardware* yang Anda miliki. Selain itu, Anda dipersilakan untuk melakukan beberapa penyesuaian seperti implementasi *early stopping*, *batch and epoch processing*, *error tolerance*, serta hal lainnya.

Anda diharapkan memperhatikan terkait penyesuaian ukuran dataset, dikarenakan terdapat beberapa hal seperti *overfit* serta *underfit* yang mampu berdampak pada memberikan hasil akhir yang kurang representatif.

Extra Point: Silakan eksplorasi terkait kemungkinan dilakukannya *transfer learning* antar bahasa, seperti dari *English* ke *Indonesian* atau hal lainnya. Juga kombinasi bahasa yang berasal dari satu *language family*, seperti kombinasi bahasa-bahasa yang tergolong ke dalam keluarga *germanic language* lalu diuji ke bahasa Indonesia.

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

Setelah menyelesaikan tahap 1, mahasiswa diharapkan sudah mengerti cara kerja/*flow* dari program ini. Singkatnya, kita telah berhasil melatih model dengan parameter *default* menggunakan *Simple Contrastive Learning* dari arsitektur `SimCSE` untuk menghasilkan *sentence embeddings*.

Pada tahap ini, tugas mahasiswa adalah menggunakan model yang telah dilatih di tahap 1 untuk melakukan evaluasi lebih lanjut pada kumpulan data yang belum dilihat sebelumnya. Sebagai acuan, Anda dapat melihat script `evaluation.py`.

Jika sudah berhasil, cobalah untuk bereksperimen dengan melatih model baru dengan beberapa parameter yang diubah. Anda dibebaskan untuk mengubah parameter di Model Arguments (di bagian SimCSE's arguments), ataupun di Data Training Arguments. Tujuannya adalah untuk melihat bagaimana perubahan ini mempengaruhi performa *sentence embeddings* dalam tugas tertentu (misalnya, pencarian semantik atau klasifikasi kalimat).

Sebagai dokumentasi, tulis laporan singkat mengenai hasil eksperimen, termasuk perubahan parameter, alasan perubahan, serta analisis hasil dari setiap percobaan. Jika performa model tidak meningkat, cukup jelaskan mengapa hasilnya mungkin tetap sama atau menurun.

Catatan: Eksperimen ini bertujuan untuk memberikan pemahaman lebih mendalam tentang bagaimana parameter pada model `SimCSE` memengaruhi kualitas representasi kalimat dalam tugas-tugas berbasis kesamaan semantik.

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] 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