TUGAS

PENELUSURAN TENTANG PENTINGNYA PREPROCESSING

PEMROSESAN BAHASA ALAMI - B

PIETER CHRISTY YAN YUDHISTIRA

235150201111068

Bahasa Alami

Bahasa yang dikembangkan oleh manusia yang disampaikan baik tulis/lisan/isyarat

Pemrosesan Bahasa Alami

Studi pemrosesan bahasa alami oleh komputer

Teknik Pemrosesan Teks Umum

Text cleaning
Umumnya *lowercasing*, penghapusan URL, non word/non white space, digit, emoji dll.
atau lebih umum seperti penanganan singkatan, akronim, koreksi kata

Tokenization
Split jadi kata individual
ex: “i”, “love”, “you”

Stopwords
Hapus kata
ex: "the," "is," and "and,"

Stemming
Ubah jadi kata dasar
ex: “membeli” “beli”

Lemmatization
Ubah jadi kata dasar
+ tapi lihat konteks

Lainnya:
- N-gram processing
- POS Tagging
- Grammatical parsing/chunking
- Bag-of-Words (BoW)
- TF-IDF
- Word/Sentence Embedding
- Document Indexing
- Word Sense Disambiguation
- Text Summarization

Text Preprocessing in NLP
Text Preprocessing | NLP | Steps to Process Text (Kaggle)
Data Collection and Preprocessing for Large Language Models
Lourdusamy, Ravi & Abraham, Stanislaus. (2018). A Survey on Text Pre-processing Techniques and Tools. International Journal of Computer Sciences and Engineering. 06. 148-157. 10.26438/ijcse/v6si3.148157.

Kenapa Text Preprocessing Penting

Diagram flow 4, terdapat lowercasing, stopword, token, negation and slang handling, dll

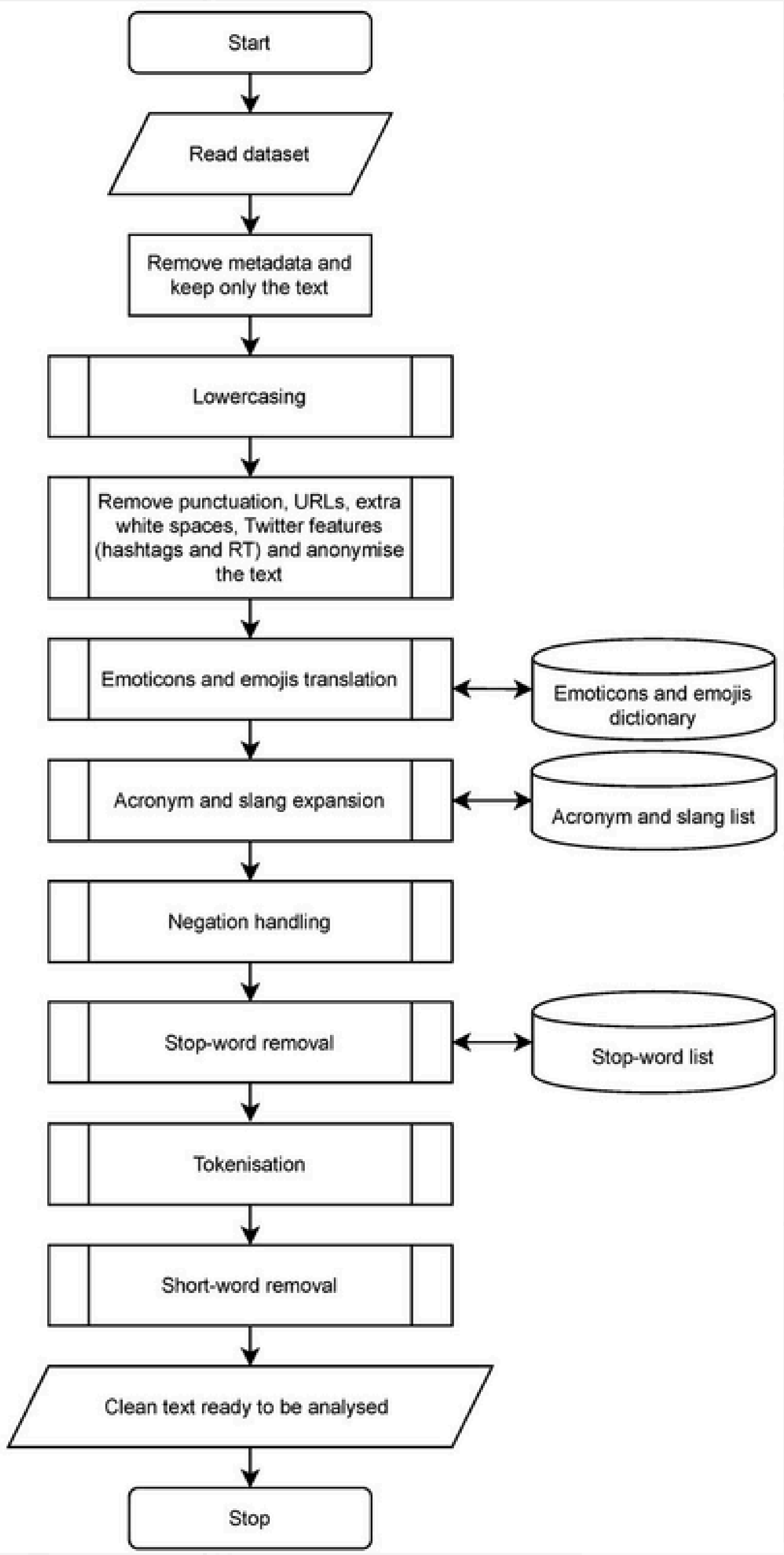


Table 5. Accuracy of the naïve Bayes classifier when testing, separately, with positive and negative tweets.

	Raw Data	Flow 1	Flow 2	Flow 3	Flow 4	Flow 5
Naïve Bayes Negative	**95.44**	93.52	90.63	91.63	93.03	92.09
Naïve Bayes Positive	41.48	54.15	55.11	52.02	**59.15**	48.10
Naïve Bayes Total	84.04	85.16	83.22	83.38	**86.21**	83.46

Table 6. Accuracy of the naïve Bayes classifier when testing, separately, with the positive and negative tweets included in the gold standard.

	Raw Data	Flow 1	Flow 2	Flow 3	Flow 4	Flow 5
Naïve Bayes Negative	**97.61**	97.49	94.70	95.72	96.53	96.81
Naïve Bayes Positive	33.33	42.50	56.96	53.33	**62.69**	45.83
Naïve Bayes Total	88.50	88.28	88.42	88.98	**91.72**	89.14

Dalam kasus analisis sentimen, text preprocessing dapat meningkatkan akurasi model.*

(Semakin tinggi flow, bertambah kombinasi teknik prep)

*Palomino, Marco & Aider, Farida. (2022). Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. Applied Sciences. 12. 8765. 10.3390/app12178765.

Tetapi Tidak Selalu Bagus

“Experimental results indicate that the removal of URLs, the removal of stop words and the removal of numbers **minimally affect the performance of classifiers**; ...
Therefore, removing stop words, numbers, and URLs is **appropriate to reduce noise but does not affect performance.”**
(Zhao, Jianqiang, 2017)

Table 6
Accuracies for the three non-deep models on the four test dataset used. In bold black and bold red are shown the best and the worst results, respectively, for each model. For NB on 20N, we avoid black bold for most of the column because of the same results.

	IMDB			PCL			FNS			20N		
Preprocessing	NB	SVM	LR	NB	SVM	LR	NB	SVM	LR	NB	SVM	LR
DON	0.767	**0.835**	0.798	0.726	**0.729**	**0.693**	0.685	**0.630**	0.640	0.040	**0.160**	**0.140**
LOW	0.771	0.831	0.801	**0.736**	0.696	0.668	0.695	0.665	0.650	0.040	0.140	0.100
RSW	0.787	0.831	**0.833**	0.719	0.651	0.686	0.705	**0.715**	0.660	**0.020**	**0.100**	0.060
STM	0.741	0.794	0.773	0.683	0.678	0.691	**0.675**	0.645	0.640	0.040	**0.160**	0.080
LOW → RSW	0.787	0.828	**0.833**	0.706	0.671	0.683	0.720	0.690	0.680	0.040	0.140	0.040
LOW → STM	**0.725**	0.803	**0.770**	0.678	0.668	0.688	0.700	0.665	**0.615**	0.040	0.120	0.100
RSW → LOW	**0.789**	**0.835**	0.820	0.721	0.663	0.691	**0.725**	0.690	0.675	0.040	0.120	**0.020**
RSW → STM	0.780	0.794	0.811	**0.671**	0.641	0.656	0.680	0.695	0.675	**0.020**	**0.160**	0.100
STM → LOW	**0.725**	0.803	0.800	0.678	0.668	0.673	0.700	0.665	0.635	0.040	0.120	0.060
STM → RSW	0.775	**0.790**	0.821	0.681	0.641	**0.646**	0.675	0.675	0.670	**0.020**	0.140	0.120
LOW → STM → RSW	0.750	0.799	0.820	0.678	**0.623**	0.648	0.695	0.680	0.645	0.040	0.140	0.080
LOW → RSW → STM	0.747	0.794	0.821	0.668	0.636	0.661	0.700	0.685	0.650	0.040	0.140	0.080
STM → LOW → RSW	0.749	0.797	0.814	0.678	**0.623**	0.661	0.690	0.675	0.645	0.040	0.140	0.080
STM → RSW → LOW	0.749	0.797	0.814	0.678	**0.623**	0.661	0.690	0.685	0.655	0.040	0.140	0.080
RSW → LOW → STM	0.757	0.797	0.807	0.673	**0.623**	0.678	0.720	0.670	0.655	0.040	0.140	0.120
RSW → STM → LOW	0.756	0.797	0.803	0.673	**0.623**	0.651	0.720	0.675	**0.685**	0.040	**0.160**	0.080

Dalam beberapa dataset, DON (Do Nothing) juga dapat sama bagusnya dengan kombinasi teknik preprocessing dan kombinasi prepocessing menurunkan akurasi (Siino, M, 2024)

	Embedding Preprocessing	Topic categorization				Polarity detection				
		BBC	20News	Reuters	Ohsumed	RTC	IMDB	PL05	PL04	SF
CNN	Vanilla	94.6	89.2	93.7	35.3	83.2	87.5†	**76.3**	58.7†	**91.2**
	Lowercased	93.9†	84.6†	**93.9**	**36.2**	83.2	85.4†	**76.3**	60.0†	91.1
	Lemmatized	94.5	88.7†	93.8	35.4	83.0	86.8†	75.6	62.5	**91.2**
	Multiword	**95.6**	**89.7**	**93.9**	35.2	**83.3**	**88.1**	75.9	**63.1**	**91.2**
CNN+LSTM	Vanilla	97.0	90.7†	**93.1**	30.8†	**84.8**	**88.9**	79.1	71.4	87.1†
	Lowercased	96.4	**91.8**	92.5†	30.2†	84.5	88.0†	79.0	**74.2**	87.4
	Lemmatized	96.6	91.5	92.5†	31.7†	83.9	86.6†	78.4†	67.7†	87.3
	Multiword	**97.3**	91.3	92.8	**33.6**	84.3	87.3†	**79.5**	71.8	**87.5**

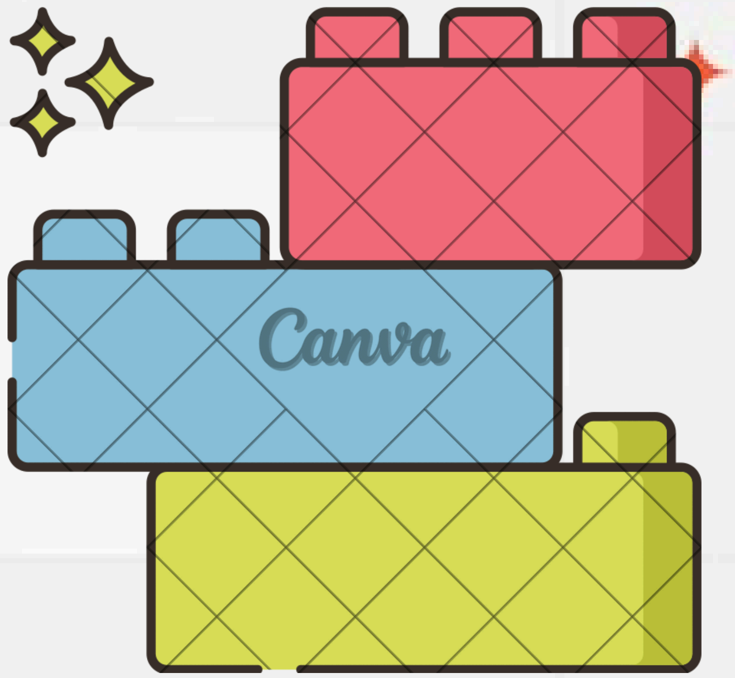
Dalam model DL (CNN/LSTM), teknik preprocessing sangat bergantung dengan konteks dataset, kompleksitas dan specific domain apa dll. (Camacho-Collados, J, 2017)

*Menurut saya, text preprocessing usefulness **case-by-case basis**, eksperimen dan eksplorasi teknik terbaik itu cara optimal untuk text prepocessing*

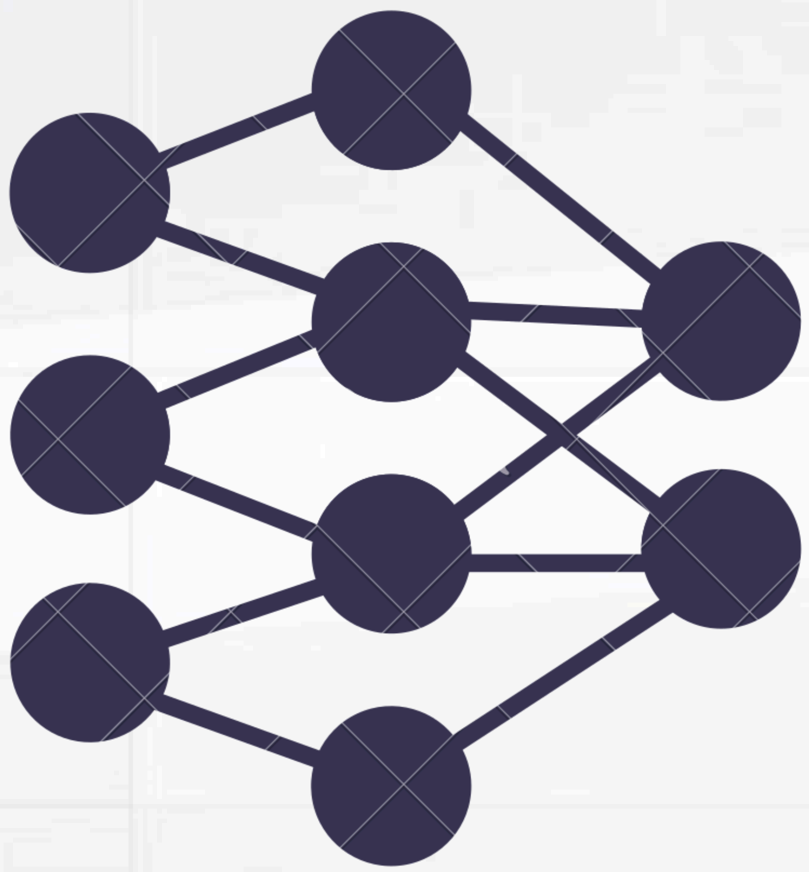
**Marco Siino, Ilenia Tinnirello, Marco La Cascia, Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers, Information Systems, Volume 121, 2024, 102342, SSN 0306-4379, https://doi.org/10.1016/j.is.2023.102342.*

***Zhao, Jianqiang & Gui, Xiaolin. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2672677.*

****Camacho-Collados, J., & Pilehvar, M. T. (2017). On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780.*



Bagaimana dengan konteks Large Language Model (LLM)?



Riset LLM di Text Preprocessing

Studi berkaitan LLM saat ini belum banyak yang eksplor teknik seperti “stemming” atau “POS Tagging”

Banyak yang lebih berfokus di arsitektur model, teknik training, skalabiltas hingga privasi dan halusinasi model

Tetapi kita dapat mengekpsplor beberapa aspek dari LLM dan menganalisis dari konteks text preprocessing

*Patil, R., & Gudivada, V. (2024). A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs). Applied Sciences, 14(5), 2074. https://doi.org/10.3390/app14052074

**Thennal, D. K., Fischer, T., & Biemann, C. (2024). Large Language Models Are Overparameterized Text Encoders. https://doi.org/10.48550/arxiv.2410.14578

Language Model In General

Scenario	Models	Accuracy	Macro Avg.			Weighted Avg.		
			precisions	*recalls*	*f1-scores*	*precisions*	*recalls*	*f1-scores*
WO	albert	0,86308	0,86435	0,86252	0,86249	0,86465	0,86308	0,86293
	bert	0,88733	0,88692	0,88692	0,88690	0,88727	0,88733	0,88729
	deberta	0,66983	0,68319	0,67118	0,66442	0,68328	0,66983	0,66364
	distilbert	**0,89263**	**0,89221**	**0,89257**	**0,89235**	**0,89227**	**0,89263**	**0,89241**
	electra	0,64238	0,65367	0,64226	0,63775	0,65283	0,64238	0,63728
	roberta	0,87242	0,87182	0,87256	0,87212	0,87172	0,87242	0,87199
	scibert	0,84654	0,84642	0,84665	0,84646	0,84634	0,84654	0,84637
	xlnet	0,79021	0,79197	0,79010	0,79044	0,79199	0,79021	0,79050
SW_STEM	albert	0,83742	0,83816	0,83748	0,83760	0,83804	0,83742	0,83751
	bert	0,85704	0,85699	0,85735	0,85703	0,85678	0,85704	0,85677
	deberta	0,62300	0,62897	0,62191	0,61863	0,62879	0,62300	0,61911
	distilbert	**0,86683**	**0,86731**	**0,86751**	**0,86728**	**0,86661**	**0,86683**	**0,86659**
	electra	0,58917	0,59915	0,58965	0,58206	0,59935	0,58917	0,58194
	roberta	0,83504	0,83472	0,83533	0,83481	0,83447	0,83504	0,83454
	scibert	0,82221	0,82177	0,82201	0,82137	0,82177	0,82221	0,82147
	xlnet	0,36467	0,37252	0,36490	0,35575	0,37246	0,36467	0,35564
SW_LEMMA	albert	0,85546	0,85641	0,85566	0,85579	0,85620	0,85546	0,85559
	bert	0,88108	0,88101	0,88131	0,88111	0,88079	0,88108	0,88089
	deberta	0,63863	0,65837	0,63783	0,63369	0,65822	0,63863	0,63409
	distilbert	**0,88304**	**0,88266**	**0,88327**	**0,88290**	**0,88246**	**0,88304**	**0,88269**
	electra	0,60596	0,62290	0,60666	0,60125	0,62352	0,60596	0,60129
	roberta	0,85492	0,85404	0,85505	0,85433	0,85398	0,85492	0,85423
	scibert	0,83833	0,83924	0,83856	0,83789	0,83951	0,83833	0,83790
	xlnet	0,76067	0,76088	0,76090	0,76052	0,76087	0,76067	0,76040

0.892

WA F1 Score Raw Text

VS

0.866

WA F1 Score with Stemming

0.883

WA F1 Score with Lemmatization

Model pre-trained seperti BERT, ROBERTA dan SciBERT **malah turun performanya** jika menerapkan Stemming dan Lemmatization.

Dapat diasumsikan model robust → seperti BERT sudah tidak perlu menyederhanakan training dgn text preprocessing seperti lemma

Haviana, S. F. C., Mulyono, S., & Badieah, B. (2023). The Effects of Stopwords, Stemming, and Lemmatization on Pre-trained Language Models for Text Classification: A Technical Study. https://doi.org/10.1109/eecsi59885.2023.10295797

Noise data dalam training

"Thorough extensive experiments on synthetic biographies data, we reveal that existing pretrained large language models have **established preferences** as human beings do, e.g. **preferring formal texts and texts with less spelling errors**." (Havrilla, A, 2024)

"Our findings suggest it is **critical to filter out documents** containing large amounts of dynamic, global noise during both pretraining and fine-tuning." (Singh, A., 2024)

Noise dalam data, baik model sebagus apapun seperti LLM akan mempengaruhi performanya*

Namun, banyak proposed solution tidak berkaitan dengan text preprocessing yang umum kita lakukan dalam kasus NLP

Preprocessing umum untuk LLM

Quality filtering

Deduplication

Memastikan tidak ada duplikat di data/keluaran model

Privacy scrubbing

anonymization, redaction, or tokenization

THE ELEPHANT ON THE ROOM

Computation dan time cost

"Typical training times for the lemmatizer models on UD treebanks **with 50 training epochs are 1–2 hours** on one Nvidia GeForce K80 GPU card. The largest treebanks (Czech-PDT 1.2M tokens and Russian-SynTagRus 870K tokens) **took approximately 15 hours** to train for the full 50 epochs."

Russian-SynTagRus memiliki sekitar 1 M token atau 66k kalimat dari teks 1960 and 2016 sedangkan LLM seperti GPT 4 diestimasi belajar dari 5 Trilion token atau 260B kata

Waktu dan komputasi yang dibutuhkan dalam LLM akan jauh lebih besar menggunakan teknik yang overly processed, untuk akurasi yang tidak menjamin lebih baik

*Singh, A., Singh, N., & Vatsal, S. (2024). Robustness of llms to perturbations in text. arXiv preprint arXiv:2407.08989.

**Havrilla, A., & Iyer, M. (2024). Understanding the effect of noise in llm training data with algorithmic chains of thought. arXiv preprint arXiv:2402.04004.

***Liu, Y., He, H., Han, T., Zhang, X., Liu, M., Tian, J., ... & Ge, B. (2024). Understanding llms: A comprehensive overview from training to inference. Neurocomputing, 129190.

***Droganova, K., Lyashevskaya, O., & Zeman, D. (2018). Data Conversion and Consistency of Monolingual Corpora: Russian UD Treebanks. In Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), December 13–14, 2018, Oslo University, Norway (No. 155, pp. 52-65). Linköping University Electronic Press.

*****How much LLM training data is there, in the limit?

Interesting Text Preprocessing on LLM Nowadays

Quality Filtering
Ultra-FineWeb: Efficient Data Filtering and Verification for High-Quality LLM Training Data

Image-Text Quality Enhancement
Beyond Filtering: Adaptive Image-Text Quality Enhancement for MLLM Pretraining

Deduplication
BaichuanSEED: Sharing the Potential of ExtensivE Data Collection and Deduplication by Introducing a Competitive Large Language Model Baseline

SGD LLM
SWAN: Preprocessing SGD Enables Adam-Level Performance On LLM Training With Significant Memory Reduction

Normalization
Flash normalization: fast RMSNorm for LLMs

Privacy Preserving
Exploring Federated Pruning for Large Language Models

LLM yang lebih robust:

- CABINET: Content Relevance based Noise Reduction for Table Question Answering
- Can Language Models Perform Robust Reasoning in Chain-of-thought Prompting with Noisy Rationales?
- dll

Conclusion

Apakah text preprocessing penting, dalam pengembangan LLM?

Iya, meskipun banyak studi menunjukan data mentah mirip performanya, text preprocessing dalam data teks diperlukan dan akan berpengaruh dalam performanya.
Bahkan untuk kasus LLM, kompleksnya data teks untuk dipahami komputer dan noise dari suatu data teks critical untuk dihadapi

Teknik preprocessing apa yang relevan dalam pengembangan LLM?

Kalau teknik costly seperti stemming atau lemmatization? tentunya tidak. Cost intensive dan terbukti gak berdampak besar
Teknik sederhana, seperti tokenisasi atau teknik kompleks seperti adaptive quality filtering? bisa jadi.

Text preprocessing sangat case-by-case basis, teknik apapun sesuai dalam kondisi

TERIMA KASIH!

Natural Language Processing