Embedding Meets Frequency: Novel Approaches to Stopword Identification in Burmese

Ye Kyaw Thu, Thepchai Supnithi

Language and Semantic Technology Research Team (LST),
National Electronics and Computer Technology Center (NECTEC), Thailand

November 27, 2023

1. Introduction
2. Related Work
3. Baseline Methods
4. Embedding Meets Frequency
5. Experimental Setting
6. Extracted Stopword List
7. Evaluation
8. Conclusion

Introduction

- Stopwords are words that, though frequent in a language, carry little to no meaning in specific contexts
- Their importance in NLP is twofold
- Firstly, by identifying and removing stopwords, computational efficiency can be dramatically improved as the focus remains on significant terms
- Secondly, the effectiveness of data analysis and machine learning models increases as noise is reduced.
- Although many languages have seen progress in stopword extraction, Burmese lags behind
- Addressing this gap, our research offers an in-depth study of automatic stopword extraction for Burmese

Related Work

- HaCohen-Kerner and Blitz delved into the development of a stopword list tailored for the Hebrew language, aiming to enhance the efficacy of NLP and IR
- Baseline: Term Frequency Normalized (TFN), Term Frequency Double Normalized (TFDN), and Inverse Document Frequency Normalized (IDFN)
- A comparison of the resultant Hebrew stopword list to its established English counterparts revealed an intriguing overlap: approximately 60% for the top 100 words
- Tursulistyono et al. deployed the Term Frequency–Inverse Document Frequency (TF-IDF) approach for Indonesian stopwords
- Comparative evaluations revealed the superiority of the TF-IDF approach over previous methods, especially in terms of revealing high-frequency words and words consistently present across all documents

Related Work: Burmese Stopwords

- Khine et al. employed supervised learning techniques using the Naïve Bayes and KNN algorithms for Burmese text classification
- their preprocessing methodology involved the removal of stopwords
- the authors did not elaborate on the methodology employed in the compilation of this list, leading to the assumption that it was manually curated
- Interestingly, the list (324 stopwords) also includes vowel characters or combinations, such as ‘ဦ’, ‘ု’, ‘ူ’, ‘ဲ’, ‘ဪ’, which are not typically considered stopwords

Baseline Methods: 1. Term Frequency (TF)

TF

Term Frequency (TF) is a fundamental concept in text analytics, representing the frequency of a term in a given document. It is computed using the formula:

$$\text{TF}(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}} \quad (1)$$

For instance, consider a document containing 100 words wherein the term "book" appears 5 times. The TF for "book" would be:

$$\text{TF}(\text{ "book"}) = \frac{5}{100} = 0.05$$

Typically, terms with high TF values in a majority of documents might be considered for stopword extraction

Baseline Methods: 2. Normalized Term Frequency

Normalized TF

The Normalised Term Frequency (Normalised TF) is a modification of the basic Term Frequency (TF) which aims to account for the variation in document lengths. It is typically computed as:

$$\text{Normalised TF}(t) = \frac{\text{TF}(t)}{\max_k \text{TF}(k)} \tag{2}$$

Where $\text{TF}(t)$ is the term frequency of term t and $\max_k \text{TF}(k)$ is the maximum term frequency among all terms in the document. Taking our previous example, if the term "book" has a TF of 0.05 and it's the term with the highest frequency in the document, then its Normalised TF would also be 0.05. However, if another term, say "paper", had the highest frequency with a TF of 0.08, then the Normalised TF for "book" would be:

$$\text{Normalised TF}(\text{"book"}) = \frac{0.05}{0.08} \approx 0.625$$

Baseline Methods: 3. Term-based Random Sampling

Term-based Random Sampling is another technique leveraged in text analytics to extract features or significant terms from large corpora. Instead of relying solely on frequency or statistical measures, this method randomly samples terms based on their frequency distribution, thereby accounting for both common and rare terms. Mathematically, the probability $P(t)$ of selecting a term t is proportional to its term frequency:

$$P(t) = \frac{\text{TF}(t)}{\sum_k \text{TF}(k)} \tag{3}$$

Where $\text{TF}(t)$ is the term frequency of term t and the denominator represents the sum of term frequencies of all terms in the document. For instance, continuing with our previous examples, if the term "book" has a TF of 0.05 in a document, and the sum of all term frequencies is 20, then the probability of selecting "book" would be:

$$P(\text{"book"}) = \frac{0.05}{20} = 0.0025$$

Baseline Methods: 4. Entropy

Entropy

Entropy measures the uncertainty or unpredictability associated with a term's distribution. A term that appears uniformly across all documents tends to have high entropy and is typically less informative; such terms are potential stopwords. The entropy $H(t)$ of a term t is given by:

$$H(t) = -\sum_{i=1}^{D} p(d_i|t) \log p(d_i|t) \tag{4}$$

Where D is the number of documents in the corpus, and $p(d_i|t)$ is the probability of term t appearing in document d_i. For our recurrent example with the term "book", if it appears uniformly across 10 documents such that $p(d_i|\text{“}book\text{”}) = 0.1$ for each document, its entropy would be:

$$H(\text{“}book\text{”}) = -10 \times 0.1 \log(0.1) \approx 2.302$$

Baseline Methods: 5. Co-occurrence Network

Co-occurrence Network

Formally, for a given window size around each word, co-occurrences are aggregated to construct a graph G, where each edge between word w_i and w_j carries a weight representing their co-occurrence frequency. Using graph metrics, especially degree centrality, potential stopwords can be identified. Degree centrality for a node v in graph G is given by:

$$C_d(v) = \frac{d_v}{n-1} \tag{5}$$

where d_v is the degree of node v and n is the total number of nodes in G. In our consistent example, if the term "book" has connections with a majority of other terms, its centrality score would be high, indicating its potential status as a stopword.

Baseline Methods: 6. Language Model

Language Model

In the context of stopword extraction, n-gram language models can be instrumental in gauging the significance, or lack thereof, of a term within a corpus. The principle behind using n-gram models for this task is straightforward: stopwords, being frequent and contextually neutral, typically exhibit high probabilities across various contexts.
Given an n-gram $(w_1, w_2, \ldots, w_n)$, the probability of the word w_n appearing after the sequence $(w_1, w_2, \ldots, w_{n-1})$ is given by:

$$P(w_n|w_1, w_2, \ldots, w_{n-1}) = \frac{\text{Count}(w_1, w_2, \ldots, w_n)}{\text{Count}(w_1, w_2, \ldots, w_{n-1})} \quad (6)$$

For example, if the bi-gram read "book" appears 10 times in our corpus and the word "read" appears 50 times, the probability of "book" appearing after "read" is $P(\text{"book"}|\text{"read"}) = \frac{10}{50} = 0.2$.

Embedding Meets Frequency: 1. Word2Vec and Frequency

Word2Vec and Frequency

Word2Vec provide dense vector representations of words by capturing their semantic meanings based on co-occurrence patterns in large corpora. These representations can be harnessed for the purpose of stopword extraction, especially when combined with traditional frequency-based measures. Given a term w, its normalized term frequency (TF_{norm}) is given by:

$$\text{TF}_{\text{norm}}(w) = \frac{\text{frequency of } w}{\text{maximum term frequency in the corpus}} \tag{7}$$

Let $\text{CS}(w)$ denote the average cosine similarity of the Word2Vec embedding of term w to all other terms in the corpus. The combined score $Score(w)$ for a term w can be computed as:

Word2Vec and Frequency

$$Score(w) = \alpha \times \mathrm{TF}_{\mathrm{norm}}(w) + (1 - \alpha) \times \mathrm{CosineSim}(w) \quad (8)$$

where α is a hyperparameter in the range [0, 1] determining the weightage of term frequency against embedding similarity. For instance, for the term "book", if its $\mathrm{TF}_{\mathrm{norm}}$ is 0.05 and average cosine similarity is 0.7 with $\alpha = 0.5$, its score would be $Score(\text{"book"}) = 0.5 \times 0.05 + 0.5 \times 0.7 = 0.375$. Terms with scores leaning more towards high similarity and frequency can be considered for extraction as stopwords.

Embedding Meets Frequency: 2. FastText and Frequency

FastText and Frequency

For Burmese stopword extraction, we employ a combined metric using both fastText embeddings and term frequency. Let $\mathbf{v}_w$ represent the embedding of word w obtained from a fastText model. The term frequency, normalized to its maximum value in the corpus, is denoted as $\mathrm{TF}_{\mathrm{norm}}(w)$. The average embedding for the entire corpus is computed as:

$$\mathbf{v}_{\mathrm{avg}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_{w_i} \tag{9}$$

where N is the number of unique words in the corpus.

Embedding Meets Frequency: 2. FastText and Frequency

FastText and Frequency

The cosine similarity between the word embedding $\mathbf{v}_w$ and the average embedding $\mathbf{v}_{\text{avg}}$ is:

$$\text{CosineSim}(w) = \frac{\mathbf{v}_w \cdot \mathbf{v}_{\text{avg}}}{\|\mathbf{v}_w\|_2 \times \|\mathbf{v}_{\text{avg}}\|_2} \tag{10}$$

Our composite score for a word, which combines its normalized term frequency and its cosine similarity to the average embedding, is then:

$$\text{Score}(w) = \text{TF}_{\text{norm}}(w) \times \text{CosineSim}(w) \tag{11}$$

Using our previously discussed document with the term "book", and supposing its normalized term frequency is 0.05 and its cosine similarity to the average embedding is 0.9, the composite score would be $\text{Score}(\text{"book"}) = 0.05 \times 0.9 = 0.045$.

Experimental Setting

- The dataset, or corpus, utilized for this research was prepared over a span of 4 years
- Sentences were sourced from BBC and VOA Burmese news articles, and some texts were taken from social media pages on Facebook
- Additionally, portions of the data were derived from the monolingual corpus of Lab projects
- The domain of this dataset is general. After the removal of non-Burmese words and symbols, the corpus comprises 212,836 sentences, 4,969,603 words, and 78,923 unique words.

Experimental Setting

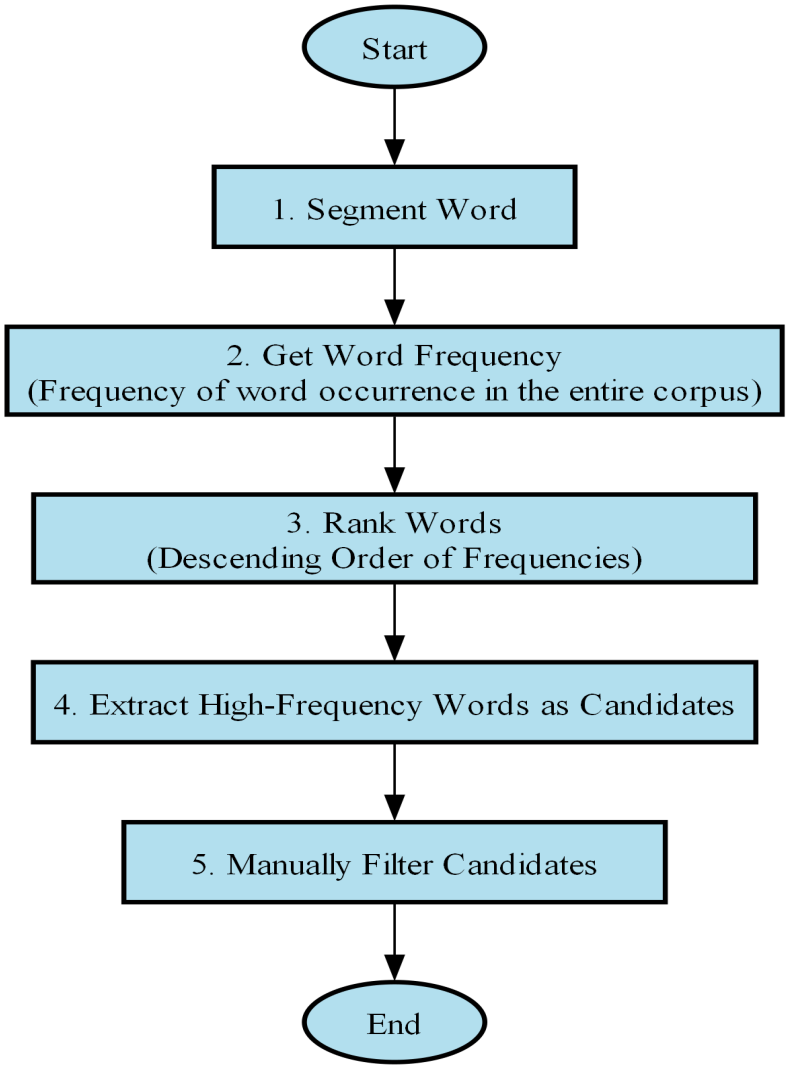

Figure: Overview of stopword extraction

Extracted Stopword List: with Word2Vec-Freq

The following are the 100 stopwords that we extracted with Word2Vec-Frequency based method:

[‘ပါ’, ‘က’, ‘တယ်’, ‘ကို’, ‘မ’, ‘သည်’, ‘နေ’, ‘တာ’, ‘တွေ’, ‘ရ’, ‘မှာ’, ‘တဲ့’, ‘များ’, ‘တော့’, ‘ဖြစ်’, ‘ပြီး’, ‘နဲ့’, ‘ရှိ’, ‘လို့’, ‘တို့’, ‘လည်း’, ‘တစ်’, ‘ခဲ့’, ‘ဘူး’, ‘သူ’, ‘ပဲ’, ‘ဆို’, ‘လိုက်’, ‘မှ’, ‘လာ’, ‘ပေး’, ‘၏’, ‘ကြ’, ‘နိုင်’, ‘သွား’, ‘ပြော’, ‘လား’, ‘ရင်’, ‘သော’, ‘ထား’, ‘မယ်’, ‘တွင်’, ‘နှင့်’, ‘လေး’, ‘မူ’, ‘ရေး’, ‘လေ’, ‘ရဲ့’, ‘ပြီ’, ‘လုပ်’, ‘ချစ်’, ‘လဲ’, ‘ချင်’, ‘ဒီ’, ‘သိ’, ‘ကောင်း’, ‘နှစ်’, ‘ဘာ’, ‘ပါစေ’, ‘လို’, ‘ဟုတ်’, ‘အားပေး’, ‘ဟာ’, ‘အရမ်း’, ‘နော်’, ‘မြန်မာ’, ‘ဖို့’, ‘အတွက်’, ‘ကြီး’, ‘ထဲ’, ‘ခြင်း’, ‘ခု’, ‘ကျွန်တော်’, ‘ပြန်’, ‘၍’, ‘သို့’, ‘ယောက်’, ‘နိုင်ငံ’, ‘ကြည့်’, ‘လူ’, ‘ပေါ့’, ‘အောင်’, ‘သူ့’, ‘ဦး’, ‘ရောက်’, ‘သေး’, ‘မည်’, ‘မင်း’, ‘သာ’, ‘လှ’, ‘ရေ’, ‘ကလေး’, ‘ဘယ်’, ‘မဂ်လာ’, ‘စေ’, ‘ရာ’, ‘သာမု’, ‘ဟု’, ‘မိ’, ‘စရာ’]

Extracted Stopword List: with FastText-Freq

The following are the 100 stopwords that we extracted with FastText-Frequency based method:

[‘ပါ’, ‘က’, ‘ကို’, ‘တယ်’, ‘မ’, ‘နေ’, ‘သည်’, ‘တာ’, ‘မှာ’, ‘တွေ’, ‘ရ’, ‘တော့’, ‘ဖြစ်’, ‘များ’, ‘တဲ့’, ‘ပြီး’, ‘လည်း’, ‘တို့’, ‘ရှိ’, ‘နဲ့’, ‘လို့’, ‘သူ’, ‘ခဲ့’, ‘ပဲ’, ‘ဆို’, ‘တစ်’, ‘ဘူး’, ‘လိုက်’, ‘မှ’, ‘လာ’, ‘ကြ’, ‘သွား’, ‘၏’, ‘ပေး’, ‘ပြော’, ‘နိုင်’, ‘လား’, ‘ရင်’, ‘သော’, ‘ထား’, ‘လေ’, ‘မယ်’, ‘တွင်’, ‘နှင့်’, ‘လေး’, ‘ပြီ’, ‘လုပ်’, ‘ချင်’, ‘လို’, ‘ချစ်’, ‘ဟုတ်’, ‘သိ’, ‘ရေး’, ‘ရဲ့’, ‘လဲ’, ‘ဘာ’, ‘မှု’, ‘ကောင်း’, ‘ဒီ’, ‘နော်’, ‘ဟာ’, ‘အရမ်း’, ‘နှစ်’, ‘ပြန်’, ‘ကြီး’, ‘ခု’, ‘ကျွန်တော်’, ‘အားပေး’, ‘ပါစေ’, ‘ထဲ’, ‘အတွက်’, ‘ဖို့’, ‘မြန်မာ’, ‘၍’, ‘ပေါ့’, ‘ခြင်း’, ‘အောင်’, ‘ယောက်’, ‘လူ’, ‘ကြည့်’, ‘သူ့’, ‘သေး’, ‘နိုင်ငံ’, ‘သို့’, ‘သာ’, ‘ဦး’, ‘မင်း’, ‘ရောက်’, ‘ကလေး’, ‘ရာ’, ‘ဟု’, ‘ရယ်’, ‘စရာ’, ‘တတ်’, ‘မည်’, ‘ရေ’, ‘စေ’, ‘ကြောင်း’, ‘တွေ့’, ‘ဘယ်’]

Evaluation: Manual

- In a comprehensive examination of stopword extraction methodologies focused on the top 100 stopwords
- an intersection of 63 stopwords was identified that was common across all extraction techniques, including both baseline and proposed methodologies
- Baselines: surfaced the highest number of unique stopwords (26), highlighting its distinct extraction approach
- Meanwhile, Co-occurrence Network followed with 8 unique stopwords
- The two newly proposed methods: exhibited an encouraging ability to extract meaningful stopwords, drawing a stopword set that's largely aligned with the baseline methods
- When juxtaposed, the Word2Vec-Frequency and FastText-Frequency proposals show minimal divergence, signaling their comparable performance in stopword extraction

Evaluation: Manual

- Consistent patterns emerged in multiple experiments for the top 100 and 300 stopwords.
- ‘part’ and ‘ppm’ tags ranked highest with frequencies of 12,617 and 13,873, respectively.
- Other tags like ‘v’, ‘conj’, ‘tn’, and ‘pron’ followed with frequencies between 3,000 to 5,600.
- Tags such as ‘adj’, ‘adv’, and ‘n’ were less frequently observed.

Evaluation: Zipf's Law

Zipf's Law

Zipf's Law is a statistical principle that postulates that in a given corpus, the frequency f of a word is inversely proportional to its rank r. This can be mathematically represented as:

$$f = \frac{C}{r^{\alpha}} \tag{12}$$

For the given data, using constants $C = 4766211.8692$ and $\alpha = 1.3720$, if a word's rank is 2 (i.e., $r = 2$), its predicted frequency f would be:

$$f = \frac{4766211.8692}{2^{1.3720}} \tag{13}$$

We compare the top 100 words in the corpus to various stopwords extraction methods, all in the context of Zipf's predictions.

Evaluation: for Word2Vec-Freq

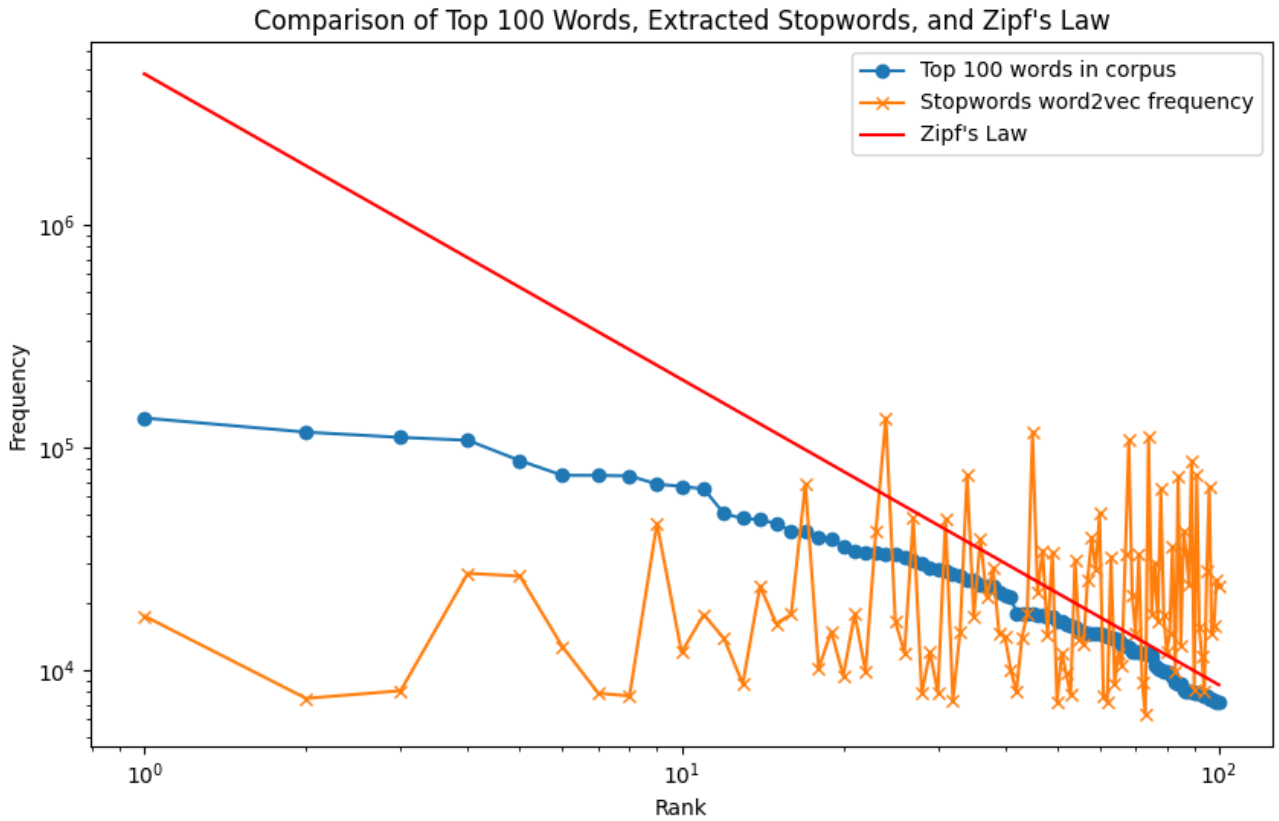

Figure: Comparison of extracted stopwords of Word2Vec-Frequency and the Zipf's law

Evaluation: for FastText-Freq

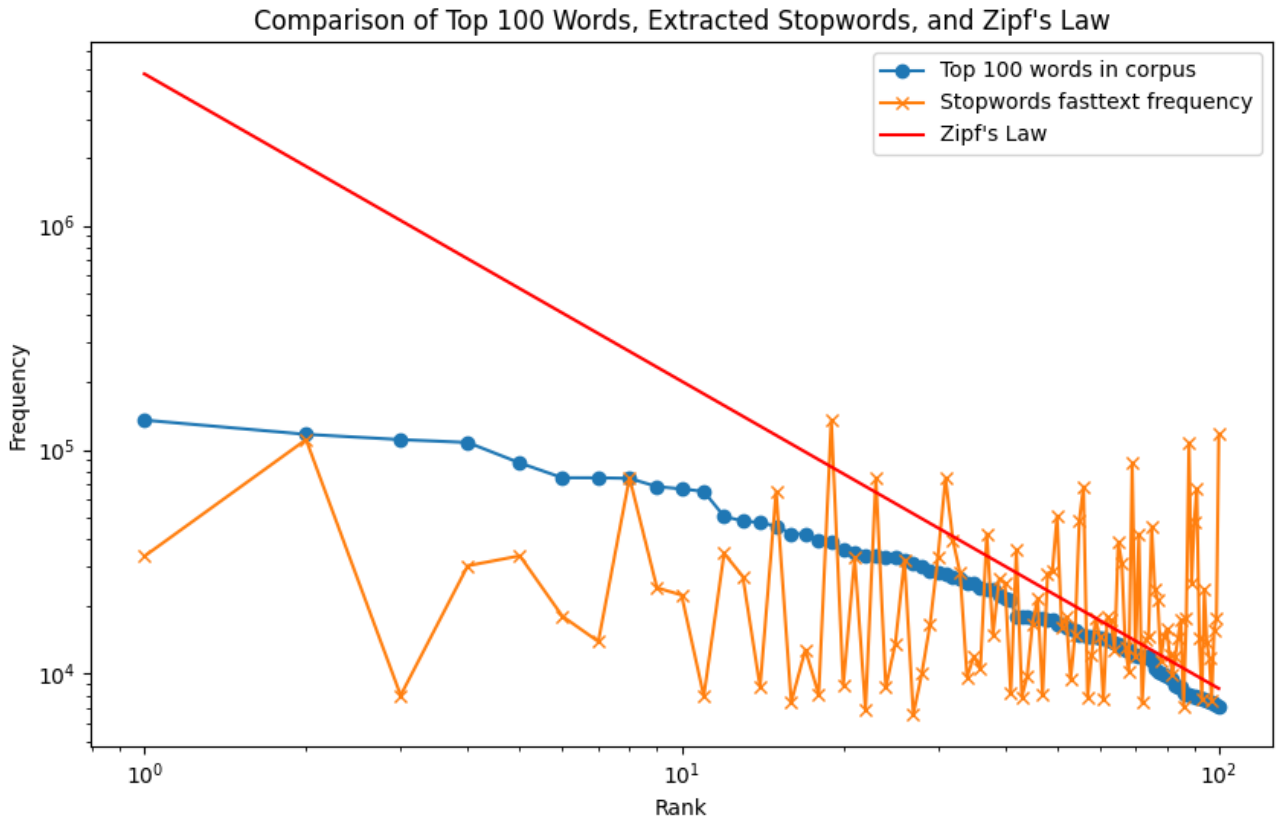

Figure: Comparison of extracted stopwords of FastText-Frequency and the Zipf's law

Conclusion

- We investigated six baseline approaches, which include well-known stopword extraction methods
- Additionally, we introduced two new proposals: a combination of word embedding and frequency, specifically word2vec-frequency and fastText-frequency
- We demonstrated that both of the proposed methods can extract a stopword list from a Burmese monolingual corpus
- Furthermore, we provided POS tag information for the extracted stopwords to facilitate future research
- In the near future, we aim to explore automatic stopword extraction techniques for domain-specific corpora, such as legal and technical texts

Thank you! Any questions?

Reference

1. HaCohen-Kerner, Yaakov & Blitz, Shmuel. (2010). Initial Experiments with Extraction of Stopwords in Hebrew. 449-453.
2. Achsan, H. T. Y., Suhartanto, H., Wibowo, W. C., Dewi, D. A., & Ismed, K. (2023). Automatic Extraction of Indonesian Stopwords. International Journal of Advanced Computer Science and Applications, 14(2), 166-171. https://doi.org/10.14569/IJACSA.2023.0140221
3. A.H.Khine, K.T.Nwet, K.M.Soe, Automatic Myanmar News Classification, 15th Proceedings of International Conference on Computer Applications, February 2017, pp. 401-408
4. Nwet, Khin & Darren, Seth. (2019). Machine Learning Algorithms for Myanmar News Classification. Journal of Natural Language Processing. 8. 17-24.
5. Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". Journal of Documentation. 28 (1): 11–21.

Reference

6. Breitinger, Corinna; Gipp, Bela; Langer, Stefan (2015-07-26). "Research-paper recommender systems: a literature survey". International Journal on Digital Libraries. 17 (4): 305–338.
7. Salton, G; McGill, M. J. (1986). Introduction to modern information retrieval.
8. Kenneth Ward Church and Patrick Hanks (March 1990). "Word association norms, mutual information, and lexicography". Comput. Linguist. 16 (1): 22–29.
9. Sobol,I.M. (2001), Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. MATH COMPUT SIMULAT,55(1–3),271-280.
10. Claude Shannon, pioneered digital information theory. FierceTelecom. Retrieved 2021-04-30.
11. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.

Reference

12. Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. Computational linguistics, 16(1), 22-29.
13. Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press: 1999. ISBN 0-262-13360-1.
14. Jurafsky, Dan; Martin, James H. (7 January 2023). "N-gram Language Models". Speech and Language Processing (PDF) (3rd edition draft ed.). Retrieved 12 Aug 2023.
15. Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781
16. Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg S.; Dean, Jeff (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems. arXiv:1310.4546

Reference

17. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov; Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 2017; 5 135–146.
18. G.K. Zipf. Human behavior and principle of least effort: an introduction to human ecology. addison wesley, cambridge, massachusetts, 1949.
19. David M. W. Powers. 1998. Applications and Explanations of Zipf's Law. In New Methods in Language Processing and Computational Natural Language Learning.