Journal of Information Systems Engineering and Management
2025, 10(16s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Incremental Learning Static Word Embeddings for Low-Resource NLP

Nathan J. Lee[1*], Nur Afny C. Andryani[2]

[1] *School of Computer Science, BINUS University, Jakarta, Indonesia (nathan.lee@binus.ac.id)*

[2] *Doctor of Computer Science Department, BINUS Graduate Program, BINUS University, Jakarta, Indonesia (nur.afny@binus.ac.id)*

**Corresponding Author: nathan.lee@binus.ac.id*

ARTICLE INFO

Received: 08 Dec 2024

Revised: 29 Jan 2025

Accepted: 10 Feb 2025

ABSTRACT

Natural Language Processing (NLP) development for Low-Resource Languages (LRL) remains challenging due to limited data availability, linguistic diversity, and computational constraints. Many NLP solutions rely on complex models and high volume/quality data, which makes them difficult to use in Low-Resource NLP. Inspired by the NLP challenges and insights revealed by various previous works, the underexplored Incremental Learning (IL) Static Word Embedding (SWE) system in the low-resource NLP case of Indonesia's local languages is proposed and presented. With basic-level models and hyperparameter sweeps, these models are tested in the scenario of incrementally incorporating 10 different local languages into themselves. The simulations indicate this type of model resists Catastrophic Forgetting (CF) very well and delivers competitive performance on the downstream task of sentiment analysis. In terms of f1 scores, the proposed model succeeds to exceed other baseline models and even rival heavy Transformer models. The proposed model can be considered as a prospective holistic solution for low-resource NLP. Future works could explore this model's behavior in finer-grained NLP tasks, different IL settings, or test more advanced models.

Keywords: Incremental Learning, Indonesian, Low Resource, NLP, Sentiment Analysis, Static Word Embedding.

1. INTRODUCTION

The development of NLP has been predominantly focused on a handful of high-resource languages (HRLs) [1]. Meanwhile, a lot of Low-Resource Languages (LRLs) remain out of reach of NLP technology [2, 3]. These languages present challenges in resource collection/availability, linguistic dissimilarity from HRLs, and various deviations in informal use [4–6]. However, NLP progress in LRLs can bring many benefits such as improving accessibility, bridging communication gaps, and supporting the language's preservation [4, 5, 7].

Researchers have explored many solutions to tackle this low-resource challenge. On the data side there are methods such as Data Augmentation, Machine Translation, Multilingual Models, and Transfer Learning from HRLs [1], [4]. On the model side, many hand-crafted and automatic methods have been used, with simpler models often outperforming complex models due to their low resource requirement [1, 2, 8–12].

Interestingly, barely any works in low-resource NLP used Incremental Learning (IL) Static Word Embedding (SWE) models. It itself is already quite a rare sight, and the works that do use them either focus on implementation [13, 14] or diachronic tasks such as topic modelling or semantic tracking [15–17]. The scattered research findings hint to this model being a great fit for a lot of the challenges in low-resource NLP.

This research intends to explore the research gap in the case study of Indonesia's local languages. We observe the behavior of SWE systems in incrementally incorporating different local languages into themselves and evaluating the model's downstream performance in Sentiment Analysis (SA). This stands out from previous works as the frames of IL data are all different languages. Through this exploratory experiment, the presented research aims to highlight the potential of this type of model as a valuable alternative for advancing low-resource NLP.

2. BACKGROUND

2.1. Static Word Embeddings

SWE models have been around for a while, with the famous Word2Vec [18] still seeing good use to this day. They work by learning dense vector representations for each word to capture their semantic meaning in a fixed Vector Space (VS). The resulting VS is relatively interpretable, with similar words being close together and word relationships being observable through simple vector arithmetics (a common example being "king - man + woman ≈ queen") [19–21].

Unlike contextual word embedding models that assign vectors on-the-fly affected by surrounding words, SWE models assign a single fixed vector for each word. This makes them computationally lighter but less capable of capturing complex linguistic features such as context. Despite this, there are many examples of SWEs holding their ground in low-resource scenarios [10–12]. Their relative computational efficiency, role in low-resource scenarios, and model interpretability are the main reasons for their persisting relevancy in NLP.

One of the interesting products of their interpretability is their usage in capturing semantic change [22–27]. A word's semantic representation across VSs can be compared with various techniques such as Cosine Similarity (CS) or word neighborhood-based methods (e.g., based on a word's nearest neighbors [26] or alignment with prototypical topic centroids [27]). Doing so across multiple VSs (e.g., trained on different time frames) enables us to observe the changes of a word's meaning over time. If these temporal VSs were equated to VS states, this semantic drift also implicitly happens in IL when a model's existing vectors move around to acommodate new incoming data [13, 14].

There are some known weaknesses of SWE models. Basic versions of them can't handle polysemy (where a word has multiple possible meanings) [18, 28] and context. This is important for fine-grained NLP tasks such as POS tagging, named entity recognition, question answering, entailment, and information retrieval [29–31]. It's not as important for coarse-grained tasks such as document classification and SA [32, 33].

There are also talks about the "instability" of SWE models, a term that refers to the non-deterministic nature of their resulting vectors after training on some data [34–36]. This is an intrinsic quality of the model, and though both are often evaluated, its connection with extrinsic quality (performance on downstream tasks) is rather uncertain. Results from [35–37] show some disconnect between model stability and downstream performance, and the widespread success of reportedly unstable models like Word2Vec and FastText cannot be simply undermined.

2.2. Incremental Learning

Incremental Learning (IL) is an ML paradigm where a model continually updates their base patterns over sequentially arriving data, unlike the common paradigm where a model is pretrained once on a complete, static dataset. One of the first instances of incremental learning in NLP was performed by [38] in their attempt to model yearly semantic shifts. In their work, the VS of year y is initialized with the trained VS of year y – 1, then allowed to train on the dataset of year y until converged.

Following that, several algorithmic advancements have been made [13, 14] and some works continue to explore IL

on SWEs [16]. However, as evidenced by the influential works and sheer research volume [39–47], the research interest in IL has long since shifted to more advanced model architectures.

These works have identified several advantages of IL over the popular paradigm of pretrain-finetune. Models that have been pretrained once have their base patterns stuck to the corpora they were trained on. This is not ideal in realistic scenarios, where language continues to evolve over time and they need to deal with things from beyond their training data [40, 41]. IL is able to continually evolve the model's base patterns, allowing constant adaptation to changing problems [43, 45], potential knowledge transfer/generalization [43, 47], and more effective use of limited and frequently changing data [39, 45].

However, there are also several unique challenges that come with using IL models:

- Catastrophic Forgetting (CF): A prominent problem where ML systems trained on a sequence of tasks will suffer performance drops on earlier tasks [42, 44–47]. Several popular mitigation strategies exist such as replay, regularization, and architecture-based methods, with various nuances in their implementation and success [39, 40, 42–44].
- Stability-Plasticity: Unlike conventional model training, IL models must refrain from fully converging on incoming data as that risks destabilizing their prior knowledge. They must balance the ability to incorporate new data (plasticity) while preserving old knowledge (stability) [45, 46, 48].

2.3. Indonesian Low Resource NLP

LRLs can be defined by several dimensions relating to their sociocultural vitality, digital or academic presence, as well as the resources available for NLP development [1, 4, 49]. Indonesia's local languages are a major example of low-resource NLP, lacking in data volume and representation especially compared to European counterparts [5, 50].

According to Ethnologue [51], many of Indonesia's local languages are at risk of dying out. Much of the written, formal, and institutional communication is based on Indonesian [3, 52]. Local languages are used more in informal communication [2, 3], many lacking standardized writing [5], and sometimes code-mixed with Indonesian or other languages [2, 6, 53–55]. Combined with the pressure of globalization, many Indonesians gravitate towards learning Indonesian and English [3], while both the intergenerational transmission [52] and number of speakers [3, 5] of local languages continue to decrease.

There are many potential benefits of NLP technology reaching Indonesia's local languages. It can bridge the communication and understanding gap between speakers, helping to mitigate ethnic conflicts [5]. It also promotes the language's use – and thus – preservation [4, 7]. With [52] predicting that Indonesia could shift to a monolingual society, this highlights the urgency of progress in Indonesian low-resource NLP.

There are several challenges in dealing with Indonesian low-resource NLP. There is the continued difficulty of collecting data and the high dialectical variation within languages [5]. Additionally, informal/colloquial usage usually contains code-mixing, a common occurrence [6] that's visible in various communication mediums such as social media [53, 54], e-commerce [55], etc.

Transformer models have dominated NLP with state-of-the-art performance in various benchmarks, however they focus more on high-resource languages [1], with examples like mBERT and mT5 having comparatively little training data for the Indonesian local languages they contain [5]. There are many examples of simpler models outperforming more complex models in low-resource scenarios due to their lower training data requirement [1, 2, 8–12]. There are also calls to minimize costs in terms of environmental impact [56, 57] and cost of development/adoption [5, 56].

3. OBJECTIVES

In the intersection of the insights and challenges highlighted in previous literature, the humble IL SWE system can be found offering a set of properties that can constitute a holistic solution for low-resource NLP:

- They can expand in increments, incorporating low volume and rapidly changing data as it becomes available, without requiring complete re-training or new model development.
- As they expand, they can represent many languages in a shared semantic space, allowing them to centrally model inter-language word relationships and naturally handle code-mixing.
- They are relatively efficient in terms of data requirement and computing cost.
- They additionally serve as versatile foundational features for many downstream NLP applications.

Therefore, this project aims to validate the robustness of this type of model in low-resource NLP scenarios. If it can maintain competitive performance for downstream tasks and not succumb to the theoretical model weaknesses, it would serve to unearth and highlight the underutilization of such models in low-resource NLP.

4. METHODOLOGY

4.1. Proposed Experiment Methodology

Basic-level SWE models are run through 10 different local language datasets, with each dataset being 1 step of the IL process. The SWE models have been pretrained in formal Indonesian beforehand as a sort of "anchor" language. In each step, the model's SA performance is tested against all 10 local language datasets. The model's generalization is also tested against an unseen dataset. The experimentation captures SA performance, and VS disruption during IL with CS Score as complementary data. The flow of each IL step is illustrated in Fig. 1:

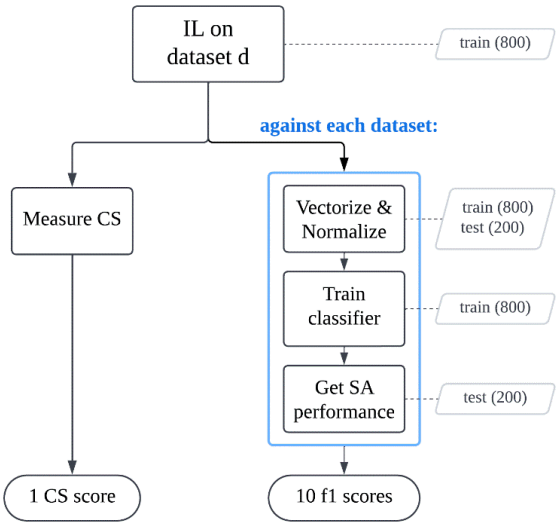

Figure 1. The flowchart of 1 IL step on dataset d

The independent variables in our test are the learning rate (alpha) and number of epochs used as the IL model's hyperparameters when training on new data. Alpha values (0.01, 0.05, 0.1, 0.25) and epochs (20, 60, 100, 140) were tested. For the SWE models, 2 models were tested (Word2Vec [18] which operates on the word-level, and FastText [58] which operates on the subword/n-gram level) and 3 model variants:

- Basic: it simply extends the model vocabulary and train on the new dataset.

- Initialization: it initially inserts a few (around 100) local language words at the vector positions of their Indonesian equivalents, then normally extend the model vocabulary and train on the new dataset. This simulates a slightly more ideal realistic scenario where we're able to supply some linguistic knowledge.
- Alignment: a separate vector space is trained on the new dataset, aligned to the IL model using Procrustes alignment based on their shared words, and then the aligned vectors are inserted into the IL model. Training and aligning separate vector spaces is more popular than IL for SWE models [4, 23–25, 27, 59, 60].

Support Vector Machine (SVM) with an rbf kernel is the classifier model applied to evaluate the task performance. The corresponding performances are evaluated using SA performance and macro f1 score. Additionally, CS score is used to measure the extent of VS disruption. The CS score was measured using 100 anchor words, comparing their vectors before and after IL training, and taking the average of the 100 words.

4.2. Data

The SA datasets of NusaX [50] are mainly used for the experimentation. There are 10 local language datasets, containing 1000 samples each in a train:test:validation split of 500:400:100. The dataset contains 3 classes (positive, neutral, negative) with a class distribution of approximately 10:6:10. Since our method does not require validation data, a train:test split of 800:200 was used instead. The datasets were arranged in lexicographic order for IL, as their names are unrelated to their internal properties. For the unseen dataset, the Minangkabau SA dataset by [2] was used. For the model pretraining, the Indonesian Wikidump data by [61] was used.

For the anchor words, 100 representative words were chosen, containing common Indonesian words and significant sentiment words. For the sentiment words, 44 words were selected from the NusaX sentiment lexicon that appeared most frequently in their Indonesian SA dataset. For the common Indonesian words, 56 out of the top 1000 most common Indonesian words from the Wikidump were randomly selected. Then the 100 words were cleaned from duplicates, numbers, abbreviations, english words, and noise.

4.3. Experimental Settings

This experiment was conducted on a standard Google Colab CPU runtime, which has 2 CPU cores. Stratified 5-Fold was used for all performance evaluations. As a comparison baseline model, one of NusaX's SA models was recreated based on TF-IDF feature extraction and SVM rbf classifier. This will be referred to as the "baseline model".

The SWE models used are the implementations of Word2Vec and FastText provided by Gensim [62]. These models are based on Skip-gram and Negative Sampling [63], which are relatively good in the realm of static word embeddings [11, 14, 24, 35]. They have been pretrained on Indonesian Wikidump data using the same hyperparameters as [58], who were also working with Wikidump data.

For the text preprocessing, the text was casefolded, cleaned from non-alphanumeric characters, and stripped of Indonesian stopwords according to Python's NLTK library. Also, hyphens (-) were replaced with spaces to keep duplicative plural words separated (for example, "kata-kata" becomes "kata", "kata" instead of "katakata".

5. RESULTS

Note: all line plots presented include shaded areas representing the 95% confidence interval of the values. Also, the ± symbol will be used as mean ± standard deviation.

5.1. Baseline Comparison

The baseline model's average macro F1 score across the 10 local language datasets was 0.761. This is slightly higher than the reported score from the original paper, likely due to differences in the train:test split and text preprocessing. Keep that in mind as the experimental settings here were slightly more favourable.

5.2. Learning Hyperparameters

Fig. 2 and Fig. 3 shows that learning rate (alpha) has a clear effect on F1 and VS disruption. Epochs has a smaller effect on F1, but clearly affects the training time as shown in Fig. 4 (Note: for Fig. 3 and Fig. 4, results of alignment models are excluded due to significantly different behavior). Both alpha and epochs have an "overshoot" range where it is counter-productive to have them too high. It is also apparent that FastText is slower and more sensitive to hyperparameter choices and VS disruption.

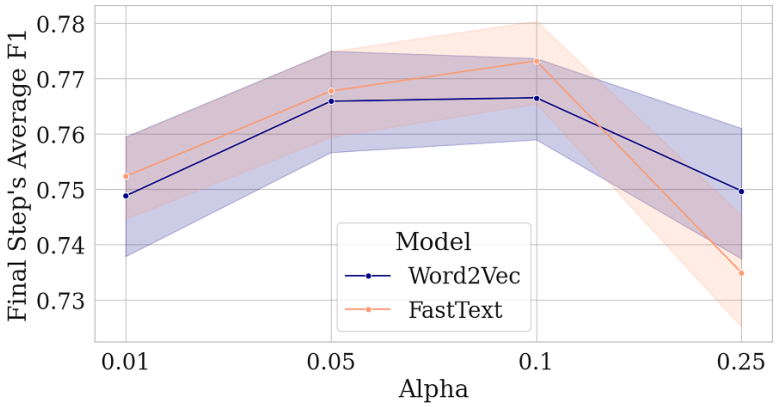

Figure 2. Effect of Alpha on final F1 performance

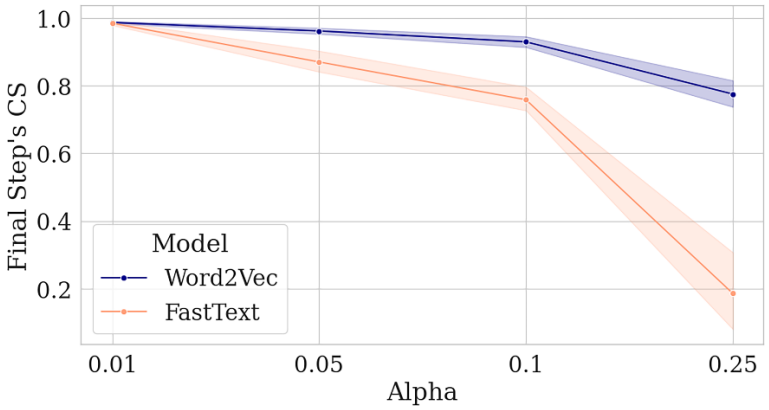

Figure 3. Effect of Alpha on ending CS score

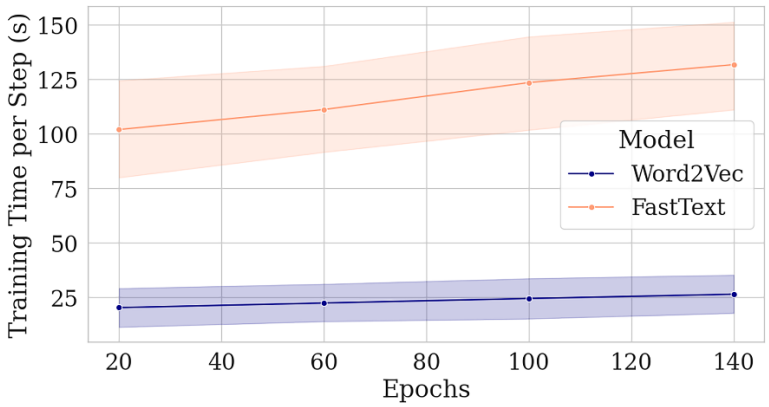

Figure 4. Effect of number of epochs on training time

5.3. Comparison of Model Variants

For the different IL variants presented in Fig. 5, overall initialization performed the best and alignment performed

the worst, while Word2Vec (W2V) and FastText (FT) have comparable performance to each other. $Word2Vec_{initialization}$ seems to be the most reliable high performer, though our highest score of 0.79 was achieved by $FastText_{initialization}$ on 0.1 alpha and 100 epochs. Looking at NusaX's reported SA performances focusing on the 10 local language datasets (excluding English and Indonesian), this score beats the baseline SVM model and even competes with heavy models like $IndoBERT_{LARGE}$ and $XLM\text{-}R_{LARGE}$.

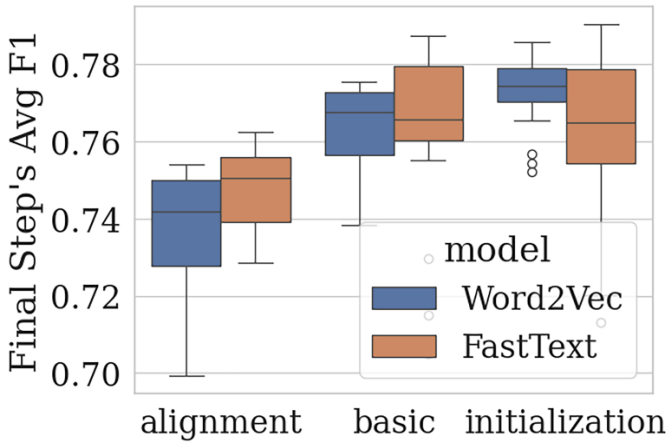

Figure 5. Boxplot of the performance of the model variants

The main challenge was maintaining a cohesive combined vector space with such little vocabulary overlap. The low data volume prevented the local language vector spaces from reaching a similar geometry/structure to Indonesian despite their theoretical linguistic similarity [5]. This was confirmed when a more aggressive vector replacement policy was allowed in the alignment models (which allowed it to overwrite the shared words' vectors) and it catastrophically deteriorated the CS score. In this case, following an anchor language seems to be a better choice than vector space alignment.

TABLE I displays some other aspects of their performance, showing that alignment models are the fastest and don't cause any vector movements in the Indonesian anchor language. Though, these training times are based on our specific implementations and may not be representative of more optimized implementations.

Table I. Training time and vs disruption of the models

Model	Training time (s)	Ending CS score
FT_{align}	3.92 ± 1.85	1.00 ± 0.00
FT_{basic}	88.3 ± 12.66	0.7 ± 0.33
$FT_{initialization}$	145.82 ± 13.13	0.7 ± 0.33
$W2V_{align}$	3.76 ± 1.68	1.00 ± 0.00
$W2V_{basic}$	11.47 ± 2.42	0.91 ± 0.09
$W2V_{initialization}$	35.41 ± 2.59	0.91 ± 0.09

5.4. IL Behavior

From the many F1 scores generated, some fine-grained behaviors of the model during IL can be extracted. Here, the performance "jump" is defined as the SA performance on dataset d upon learning dataset d, and the performance "drift" is defined as the SA performance on dataset d upon learning other datasets.

As TABLE II shows, the jump is expectedly positive. Word2Vec models seem to have higher jumps, whereas alignment models seem to have smaller jumps but more positive drifts. Speaking of drift, our expectation was for it to be clearly negative due to Catastrophic Forgetting (CF), however it is close to zero and relatively small compared

to its standard deviation. It can easily be a miniscule positive or negative number, with the unreasonable alphas like 0.25 contributing much of the negative drift results. This demonstrates some surprising resistance to CF.

Table II. Training time and vs disruption of the models

Model	Jump	Drift
FT_{align}	0.0025 ± 0.0106	0.0003 ± 0.0001
FT_{basic}	0.0114 ± 0.0076	0.0000 ± 0.0032
$FT_{initialization}$	0.0119 ± 0.0076	-0.0002 ± 0.0033
$W2V_{align}$	0.0280 ± 0.0180	-0.0001 ± 0.0004
$W2V_{basic}$	0.0496 ± 0.0134	-0.0003 ± 0.0016
$W2V_{initialization}$	0.0576 ± 0.0127	-0.0002 ± 0.0017

5.5. Generalization to Unseen Dataset

After our IL model and the baseline model have trained on the NusaX datasets, they were immediately tested against the unseen Minangkabau SA dataset [2] without training on it. The baseline model achieved a macro f1 score of 0.13 if it was trained on NusaX's Minangkabau dataset only, and 0.107 if it was trained on all 10 local language datasets. Meanwhile, the IL model's f1 score across all sweeps was 0.782 ± 0.013, reaching a maximum of 0.808. This comfortably beats mBERT's reported performance of 0.759 in [2].

6. DISCUSSION

The performance of the IL models was showing very promising performance. Even with basic models and rough hyperparameters, it delivered f1 scores that can beat other baseline models and compete with heavy Transformer models, despite our initial skepticism due to the reported importance of vector space specificity and noise in SWEs [16, 60, 64].

Part of its success was likely due to the Indonesian anchor language from pretraining. This statement is supported by the observation that the f1 scores, even in the early steps of IL training, almost never went below 0.7. As [50] has mentioned in their dataset details, there is some vocabulary overlap between the Indonesian and the 10 local language SA datasets, with an average vocabulary overlap of 28.94%. Thus, it was likely that our pretrained model was able to exploit the Indonesian words code-mixed in the local language datasets.

Another factor of success was our IL model's ability to resist the prominent IL challenge of CF, despite using no CF mitigation strategies and going through 10 IL steps. This is likely related to the SWE model architecture. It is known that connectionist networks (neural networks), especially those with high interconnectivity and unlocked weights, are susceptible to CF, where the old knowledge stored in weights gets catastrophically disrupted by new/later learned knowledge [65–67]. This is less of a problem for SWE models that represent knowledge in VSs instead of weights, where the semantic values of old vectors are more likely to drift rather than be catastrophically lost.

However, low data volumes could prevent the local language vectors from reaching a similar geometry/structure to Indonesian, which is not ideal for VS alignment attempts [4]. For further advancement, a high-resource anchor language over alignment is recommended. In addition, Word2Vec over FastText can be further explored as it's much faster for comparable performance. In general the presented experimentation's results affirm the promise of IL SWEs for low-resource NLP.

7. CONCLUSION

This project demonstrated the potential of IL SWE models for low-resource NLP scenarios. With reasonable

hyperparameters to balance stability-plasticity to incorporate new knowledge while retaining old knowledge, even the basic versions of this type of model can deliver competitive downstream performance. The combination with its other properties of maintaining a multilingual semantic space, relative computational efficiency, and versatility for other NLP tasks, affirms its prospect to be a holistic solution for the low-resource NLP effort.

The use of a high-resource anchor language was found to be beneficial for aligning the smaller chunks of new knowledge and handling common code-mixed text in Indonesia. The SWE models displayed high baseline resistance to CF, making them behave well in IL scenarios. The popular VS alignment method sufferred because the local language VSs couldn't reach similar structures to the Indonesian VS due to low data volume. Lastly, Word2Vec was generally preferable to FastText, as it delivered comparable performance with much faster running times.

There are some research limitations and suggestions for future works. First, the data volume of each local language dataset was very low. Different IL behaviors could emerge if the chunks of incoming data were larger. Second, this research only focused on the sentiment analysis task. More research could be done for finer-grained tasks such as POS tagging etc. Third, this research focuses on the behaviors of basic-level models. It would be interesting to see how more advanced ones hold up in this low-resource IL scenario.

Acknowledgements

I would like to express my gratitude to my thesis supervisor, Nur Afny C. Andryani, for her patient support and guidance in terms of the research scope, paper contents, and writing style. I would also like to sincerely thank my parents for their unwavering support throughout this project. Additionally, I extend my appreciation to the people in BINUS University and my internship company for their flexibility and understanding during this process.

REFERENCES

[1] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.12309

[2] F. Koto and I. Koto, "Towards computational linguistics in Minangkabau language: Studies on sentiment analysis and machine translation," *arXiv preprint arXiv:2009.09309*, 2020.

[3] S. Novitasari, A. Tjandra, S. Sakti, and S. Nakamura, "Cross-lingual machine speech chain for javanese, sundanese, balinese, and bataks speech recognition and synthesis," *arXiv preprint arXiv:2011.02128*, 2020.

[4] A. Magueresse, V. Carles, and E. Heetderks, "Low-resource languages: A review of past work and future challenges," *arXiv preprint arXiv:2006.07264*, 2020.

[5] A. F. Aji *et al.*, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia," 2022.

[6] M. Siregar, S. Bahri, D. Sanjaya, and others, "Code switching and code mixing in Indonesia: Study in sociolinguistics," *English Language and Literature Studies*, vol. 4, no. 1, pp. 77–92, 2014.

[7] European Language Resources Association, "BLT4All: Language Technologies for All," 2019.

[8] N. Mukhtar, M. A. Khan, and N. Chiragh, "Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains," *Telematics and Informatics*, vol. 35, no. 8, pp. 2173–2183, 2018, doi: https://doi.org/10.1016/j.tele.2018.08.003.

[9] F. Koto and I. Koto, "Towards Computational Linguistics in Minangkabau Language: Studies on Sentiment Analysis and Machine Translation," Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.09309

[10] J. Noh and R. Kavuluru, "Improved biomedical word embeddings in the transformer era," *J Biomed Inform*, vol. 120, p. 103867, 2021, doi: https://doi.org/10.1016/j.jbi.2021.103867.

[11] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A. Dahou, "Text Classification Based on Convolutional Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya," *Information*, vol. 12, no. 2, 2021, doi: 10.3390/info12020052.

[12] B. TaghiBeyglou and F. Rudzicz, "Context is not key: Detecting Alzheimer's disease with both classical and transformer-based neural language models," *Natural Language Processing Journal*, vol. 6, p. 100046, 2024, doi: https://doi.org/10.1016/j.nlp.2023.100046.

[13] H. Peng, J. Li, Y. Song, and Y. Liu, "Incrementally Learning the Hierarchical Softmax Function for Neural Language Models," 2017. [Online]. Available: www.aaai.org

[14] N. Kaji and H. Kobayashi, "Incremental Skip-gram Model with Negative Sampling," 2017. [Online]. Available: https://arxiv.org/abs/1704.03956

[15] N. Gozuacik, C. O. Sakar, and S. Ozcan, "Technological forecasting based on estimation of word embedding matrix using LSTM networks," *Technol Forecast Soc Change*, vol. 191, p. 122520, 2023, doi: https://doi.org/10.1016/j.techfore.2023.122520.

[16] A. Dridi, "Leveraging Temporal Word Embeddings for the Detection of Scientific Trends," 2021.

[17] A. Kutuzov, L. Øvrelid, T. Szymanski, and E. Velldal, "Diachronic word embeddings and semantic shifts: a survey," Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.03537

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[19] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," Association for Computational Linguistics, 2013. [Online]. Available: http://research.microsoft.com/en-

[20] O. Levy and Y. Goldberg, "Linguistic Regularities in Sparse and Explicit Word Representations," Association for Computational Linguistics, 2014.

[21] P. D. Turney and others, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, 2006.

[22] N. Tahmasebi, L. Borin, and A. Jatowt, "Survey of Computational Approaches to Lexical Semantic Change," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.06278

[23] Z. Yao, Y. Sun, W. Ding, N. Rao, and H. Xiong, "Dynamic Word Embeddings for Evolving Semantic Discovery," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, in WSDM '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 673–681. doi: 10.1145/3159652.3159703.

[24] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change," May 2016, [Online]. Available: http://arxiv.org/abs/1605.09096

[25] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change," 2016. [Online]. Available: http://nlp.stanford.edu/projects/histwords/.

[26] S. Eger and A. Mehler, "On the Linearity of Semantic Change: Investigating Meaning Variation via Dynamic Graph Models," 2017. [Online]. Available: https://arxiv.org/abs/1704.02497

[27] A. Kutuzov, "Distributional word embeddings in modeling diachronic semantic change," 2020.

[28] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *Journal of Artificial Intelligence Research*, vol. 63, pp. 743–788, 2018.

[29] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[30] N. Nakashole, T. Tylenda, and G. Weikum, "Fine-grained Semantic Typing of Emerging Entities," 2013.

[31] S. Ruder and A. Søgaard, “A Survey of Cross-lingual Word Embedding Models,” 2019. [Online]. Available: http://labs.theguardian.com/digital-language-divide/

[32] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[33] H. Schütze, “Automatic word sense discrimination,” *Computational linguistics*, vol. 24, no. 1, pp. 97–123, 1998.

[34] L. Wendlandt, J. K. Kummerfeld, and R. Mihalcea, “Factors Influencing the Surprising Instability of Word Embeddings,” Apr. 2018, doi: 10.18653/v1/N18-1190.

[35] J. Hellrich, B. Kampe, and U. Hahn, “The Influence of Down-Sampling Strategies on SVD Word Embedding Stability,” Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.06810

[36] L. Rettenmeier, “Word Embeddings Stability and Semantic Change,” 2020.

[37] Y. Wang *et al.*, “A comparison of word embeddings for the biomedical natural language processing,” *J Biomed Inform*, vol. 87, pp. 12–20, 2018, doi: https://doi.org/10.1016/j.jbi.2018.09.008.

[38] Y. Kim, Y.-I. Chiu, K. Hanaki, D. Hegde, and S. Petrov, “Temporal Analysis of Language through Neural Language Models,” 2014. [Online]. Available: https://arxiv.org/abs/1405.3515

[39] M. Jovanovic, P. Voss, A. Ai, M. Jovanović, and A. Tx, “Towards Incremental Learning in Large Language Models: A Critical Review,” 2024, doi: 10.48550/arXiv.2404.18311.

[40] X. Jin *et al.*, “Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora,” 2022. [Online]. Available: https://arxiv.org/abs/2110.08534

[41] A. Lazaridou *et al.*, “Mind the Gap: Assessing Temporal Generalization in Neural Language Models,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 29348–29363. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/f5bf0ba0a17ef18f9607774722f5698c-Paper.pdf

[42] V. V. Ramasesh, E. Dyer, and M. Raghu, “Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics,” Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.07400

[43] J. M. Coria and J. Manuel, “Continual Representation Learning in Written and Spoken Language,” 2023. [Online]. Available: https://theses.hal.science/tel-04069030v1

[44] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017, doi: 10.1073/pnas.1611835114.

[45] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural networks*, vol. 113, pp. 54–71, 2019.

[46] M. De Lange *et al.*, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 7, pp. 3366–3385, 2021.

[47] D. Lopez-Paz and M. A. Ranzato, “Gradient Episodic Memory for Continual Learning,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf

[48] L. Xiang Yang and C. Ying Xiu, “Characteristics and Techniques for Adaptive Models for Behavior Prediction in Dynamic Networks,” 2023.

[49] C. Cieri, M. Maxwell, S. Strassel, and J. Tracey, “Selection criteria for low resource language programs,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, 2016, pp. 4543–4549.

[50] G. I. Winata *et al.*, "NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages," *arXiv preprint arXiv:2205.15960*, 2022.

[51] David M. Eberhard, Gary F. Simons, and Charles D. Fennig, "Ethnologue: Languages of the World," 2021, *Dallas, Texas*.

[52] A. C. Cohn and M. Ravindranath, "Local languages in Indonesia: Language maintenance or language shift," *Linguistik Indonesia*, vol. 32, no. 2, pp. 131–148, 2014.

[53] B. Sutrisno and Y. Ariesta, "Beyond the use of code mixing by social media influencers in instagram," *Advances in Language and Literary Studies*, vol. 10, no. 6, pp. 143–151, 2019.

[54] A. M. Barik, R. Mahendra, and M. Adriani, "Normalization of Indonesian-English Code-Mixed Twitter Data," in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 417–424. doi: 10.18653/v1/D19-5554.

[55] R. and K. B. Johanes Ricky Chandra and Mahendra, "Structuring Code-Switched Product Titles in Indonesian e-Commerce Platform," in *Computational Data and Social Networks*, K.-K. R. and P. N. Chellappan Sriram and Choo, Ed., Cham: Springer International Publishing, 2020, pp. 217–227.

[56] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[57] S. Cahyawijaya, "Greenformers: Improving computation and memory efficiency in transformer models via low-rank approximation," *arXiv preprint arXiv:2108.10808*, 2021.

[58] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans Assoc Comput Linguist*, vol. 5, pp. 135–146, Jun. 2017, doi: 10.1162/tacl_a_00051.

[59] F. Carrión Salvador and Casacuberta, "Incremental Vocabularies in Machine Translation Through Aligned Embedding Projections," in *Pattern Recognition and Image Analysis*, P. and T. L. F. and S. J. A. Pinho Armando J. and Georgieva, Ed., Cham: Springer International Publishing, 2022, pp. 27–40.

[60] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing*, 2016, p. 595.

[61] G. Titan, "Indonesia Wikipedia Pages," Kaggle. Accessed: Oct. 25, 2024. [Online]. Available: https://www.kaggle.com/datasets/greegtitan/indonesia-wikipedia-pages/data

[62] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, Valletta, Malta: University of Malta, 2010, pp. 46–50. [Online]. Available: http://nlp.fi.muni.cz/projekty/gensim/

[63] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," 2013. [Online]. Available: https://arxiv.org/abs/1310.4546

[64] M. Zhao, A. J. Masino, and C. C. Yang, "A Framework for Developing and Evaluating Word Embeddings of Drug-named Entity," in *Proceedings of the BioNLP 2018 workshop*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 156–160. doi: 10.18653/v1/W18-2319.

[65] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn Sci*, vol. 3, no. 4, pp. 128–135, Apr. 1999, doi: 10.1016/S1364-6613(99)01294-2.

[66] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," in *Psychology of Learning and Motivation*, vol. 24, G. H. Bower, Ed., Academic Press, 1989, pp. 109–165. doi: https://doi.org/10.1016/S0079-7421(08)60536-8.

[67] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017, doi: 10.1073/pnas.1611835114.