RiverText: A Python Library for Training and Evaluating Incremental Word Embeddings from Text Data Streams

Gabriel Iturra-Bocaz, Felipe Bravo-Marquez

Department of Computer Science, University of Chile
National Center for Artificial Intelligence (CENIA)
Millennium Institute for Foundational Research on Data (IMFD)

Introduction

Word embeddings (WE) [1, 2] are crucial in information retrieval and natural language processing tasks, including ranking, document classification, and question answering. Despite their widespread use, traditional word embedding models have a limitation in their static nature, which hinders their ability to adapt to the ever-changing language patterns that arise in sources like social media and the web. To address this challenge, incremental word embedding algorithms have been introduced, employing a streaming learning paradigm. These algorithms have the remarkable capability to dynamically update word representations in response to new language patterns and seamlessly process continuous data streams, Incremental word embedding algorithms have been introduced to address this challenge

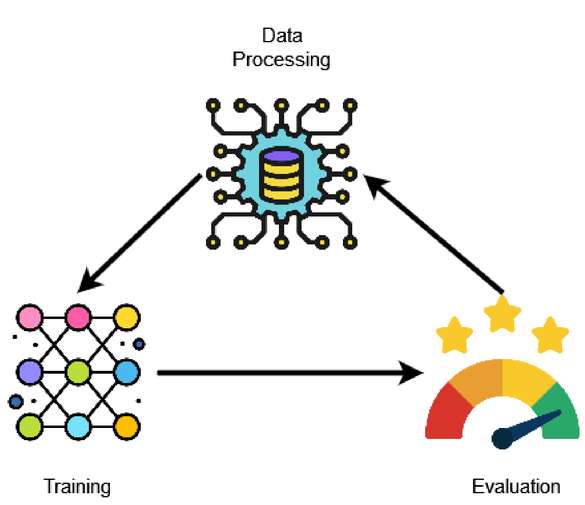

Figura 1: Streaming Learning Pipeline.

Although all the Incremental WE models have similar objectives and share common patterns, two major obstacles hinder their fair utilization:

- Firstly, the systems are stored in separate repositories, and uniform standardization processes are lacking.
- Secondly, there is no established approach for evaluating the performance of incremental WE in streaming scenarios.

RiverText Objectives

To address the aforementioned challenges, we have developed RiverText, a comprehensive framework designed for training word embeddings in a streaming context. RiverText aims to serve as a valuable resource by achieving the following objectives:

- Formalizing existing Incremental WE models into a unified framework, ensuring consistency and ease of use.
- Developing common interfaces that adhere to the principal incremental learning approaches, fostering compatibility and seamless integration.
- Proposing an evaluation scheme, called Periodic Evaluation, for Incremental WE models that utilize intrinsic NLP tasks, enabling robust and standardized performance assessment.
- Extending the functionality of River [3], a Python machine learning library explicitly designed for handling data streams, enhancing its capabilities for incremental word embedding tasks.

RiverText has three comprehensive Incremental WE models: Incremental Word Context Matrix (IWCM) [4], Incremental SkipGram (ISG), and Incremental Continuous Bag of Words (ICBOW). Notably, the ICBOW model is a novel addition proposed by our team, which boasts an accelerated training process through implementing the Adaptive Unigram Table algorithm. This innovative approach is an incremental version of the Negative Sampling algorithm initially introduced by Haji and Kobayashi [5]. All algorithms and procedures mentioned have been implemented in an open-source resource available at https://github.com/dccuchile/rivertext.

Periodic Evaluation

The proposed method for evaluating our incremental WE performance is called Periodic Evaluation. This method applies a series of evaluations to the entire model, using a test dataset associated with intrinsic NLP tasks after a fixed number, p, of instances, have been processed and trained. The algorithm takes as input the following arguments:

- The parameter p represents the number of instances between the evaluation series.
- The incremental WE model, referred to as M, is to be evaluated.
- The input text data stream, referred to as TS, used to train the incremental WE model.
- A test dataset, GR, associated with intrinsic NLP tasks.

Algorithm 1: Periodic Evaluation Algorithm. The evaluator function takes the words and their mapped vectors, and an intrinsic dataset.

```
Input: Stream ST, Incremental WE model, Intrinsic Dataset
       GR, int p
c = 0
while batch in ST do
    // train the model
    learn_many(model, batch)
    // evaluate the model during certain periods
    if c ≠ 0 ∧ c mod p then
        result = evaluator(model.wv, GR)
        // the result is stored in a JSON file
        save(result)
    c += length(batch)
```

Figura 2: Periodic Evaluation Algorithm.

Experiment and Results

To comprehensively assess the performance of Incremental WE models, we conducted the Periodic Evaluation using diverse datasets and a wide range of hyperparameter settings. The evaluation covered multiple architectural configurations, namely IWCM, ISG, and ICBOW, and incorporated intrinsic test datasets [6] for similarity (MEN [7] and Mturk [8]) and categorization (AP [9]) tasks. The hyperparameters under analysis included embedding size, window size, context size, and the number of negative samples. This extensive evaluation yielded valuable insights into the performance of various architectural configurations and hyperparameter settings, resulting in a comprehensive understanding of the subject matter.

The best configuration settings for each model and intrinsic tasks are summarized in the following graphs:

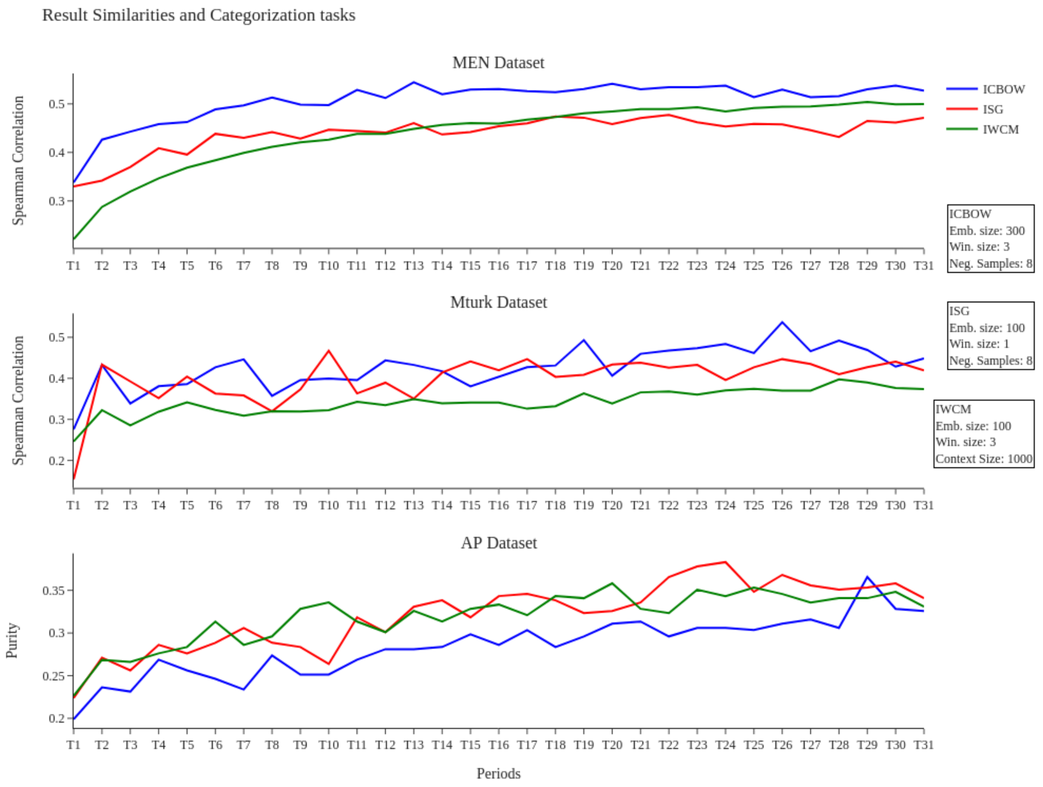

Figura 3: This figure showcases the performance dynamics across different periods for the MEN, Mturk, and AP datasets.

The figure highlights the crucial role of hyperparameter tuning in optimizing the performance of each model. The ICBOW model appears to perform more in similarity than categorization tasks. Conversely, the ISG and IWCM models perform better in the categorization task and outperform the ICBOW model. However, it is noteworthy that the results of a model for a specific intrinsic evaluation task and dataset can vary significantly and are not always related. Thus, a model may perform well in one task but poorly in another.

Conclusions

RiverText offers a systematic and unified approach for training and evaluating Incremental WE models using text data streams. RiverText provides a robust evaluation methodology based on intrinsic NLP tasks adapted to the streaming environment by standardizing the incremental WE methods. However, one limitation of the periodic evaluation approach is its inability to detect concept drift [10], which refers to changes in the data distribution over time that can significantly impact the models' performance. Although the proposed approach allows for visualizing the model's performance during training, it may not capture the effect of concept drift on the models. Therefore, caution is advised when interpreting the results in streaming scenarios.

In conclusion, RiverText addresses the challenges of training and evaluating Incremental WE models in streaming data. It achieves this through standardization and a robust evaluation approach based on intrinsic NLP tasks. Nevertheless, it is essential to know concept drift and its potential impact on model performance when applying the results in dynamic streaming environments.

Acknowledgements

This work was supported by ANID FONDECYT grant 11200290, the National Center for Artificial Intelligence CENIA FB210017, and ANID-Millennium Science Initiative Program - Code ICN17_002.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[2] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, pp. 141–188, 2010.

[3] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, *et al.*, "River: machine learning for streaming data in python," 2021.

[4] F. Bravo-Marquez, A. Khanchandani, and B. Pfahringer, "Incremental word vectors for time-evolving sentiment lexicon induction," *Cognitive Computation*, vol. 14, no. 1, pp. 425–441, 2022.

[5] N. Kaji and H. Kobayashi, "Incremental skip-gram model with negative sampling," *arXiv preprint arXiv:1704.03956*, 2017.

[6] S. Jastrzebski, D. Leśniak, and W. M. Czarnecki, "How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks," *arXiv preprint arXiv:1702.02170*, 2017.

[7] E. Bruni, N.-K. Tran, and M. Baroni, "Multimodal distributional semantics," *Journal of artificial intelligence research*, vol. 49, pp. 1–47, 2014.

[8] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th international conference on World wide web*, pp. 337–346, 2011.

[9] A. Almuhareb and M. Poesio, "Concept learning and categorization from the web," in *proceedings of the annual meeting of the Cognitive Science society*, vol. 27, 2005.

[10] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

gabrieliturrab@ug.uchile.cl