UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DESIGNING A FRAMEWORK FOR INCREMENTAL WORD VECTORS BENCHMARK

PROPUESTA PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN

SANTIAGO DE CHILE
JUNIO 2021

Designing a Framework for Incremental Word Vectors Benchmark

Introduction

The semantic change of words over time is the key to understanding how thoughts and culture evolve. Analyzing it is not an easy task because there is a massive amount of information available that makes it impossible to classify it manually. For this reason, Natural Language Processing (NLP) techniques are used to find patterns or laws in semantic evolution. [13].

The majority of NLP models need vectorial representations for processing the input values. Hence, one of its biggest problems is developing good representations of the natural language. Nowadays, the word vectors or word embedding [2] representations are the most used because they are dense vectors, and follow the distributional hypothesis [24] that postulates words occurring in the same contexts tend to have similar meanings.

Many researchers attempt to track semantic changes using word embedding representations [15, 23, 18], obtaining good results by training models capable of inferring the temporal evolution of the word. However, there are several problems with adopting these approaches; for example, these models must be trained using temporal corpus, known as diachronic corpus [15]. Often a vector of words is created by periods, which makes it challenging to compare vectors representing the same word in different periods, losing important information about the change of the word through time.

Another problem is that the models are trained in batch learning, which requires access to the data. However, this strategy has limitations in analyzing social media because it is incapable of capturing informal expressions like misspelled words, acronyms, and hashtags [7].

Some researchers have proposed incremental word representations [14, 19] based on modifications of static models, such as Skip-gram model [20], to face these problems. Thus, these models use online learning methods for their training on streaming text data. With that in

mind, when a new word or expression appears, the model can interpret it and get specific information from it.

Using incremental representations makes it possible to obtain better performance in tracking the semantic change of words. However, since these techniques work on data streams, it is not easy to compare which method or technique is the best for this task. To address this, we propose to build an open-source library that implements these techniques by performing some experiments that explore the semantic change of words in massive data streams, such as those that exist in social networks.

Technical Background

The following section does to explains and contextualizes a series of concepts used in this document.

Machine Learning (ML) [25] is the process of converting experience into expertise or knowledge. The input to a learning algorithm is training data, representing experience, and the output is some expertise, which usually takes the form of another computer program that can perform some task.

Streaming data analysis [4] in real time is the standard to obtain useful knowledge from what is happening right now, allowing organizations to react quickly when problems appear, or to detect new trends, helping them to improve their performance.

Natural language processing (NLP) [11] is a collective term referring to automatic computational processing of human languages. This includes both algorithms that take human-produced text as input, and algorithms that produce natural looking text as outputs. The need for such algorithms is ever increasing: we produce ever increasing amounts of text each year, and expect computer interfaces to communicate with them in their own language. Natural language processing is also very challenging, as human language is inherently ambiguous, ever changing, and not well defined.

Finding good representations of the words within a corpus is one of the most significant challenges of the NLP field. The commonly or most used representations by the community are **word embedding** [28]; word embedding can be defined as a mathematical function whose input can be a symbol, a word, or a phrase and whose output corresponds to a dense and low dimensional vector. Generally, these functions have been developed through neural network models using deep learning techniques.

Generally, word embedding are created from neural network-based models, which access

data sets (that are loaded into memory) and make several passes through them until the correct vectors are obtained; However, there are cases in which it is impossible to load the data set into memory because the data arrives continuously, with new terms or words appearing every second, as happens in social media such as Twitter. This type of continuously arriving data is called **data streams** [5]. In a formal way, a data stream is any ordered pair (s, Δ) where:

- s is a sequence of tuples and
- Δ is a sequence of positive real time intervals.

Since the data streams could be infinite sequences, conventional word vector models cannot define these vectors. Researchers in recent years have proposed extensions to conventional models by formulating new models [7, 14, 19, 17] that can be run on data streams, which we will call **incremental word vector** or **incremental word embedding**.

Over time words change their meaning; a specific example is the case of the word "ukrop" which changed its meaning from "drill" to "Ukrainian Patriot" during the Russian-Ukrainian crisis [26], acquiring a meaning with a negative connotation over time. This evolution of words is known as **semantic shift**.

Related work

It is well known that ML models are not able to process text directly [1], so one of the areas of study of NLP is to search for good vector representations of the words or terms that appear in a set of documents (corpus). Currently, the most popular word representations among the NLP community are word vectors or word embedding [28]. These representations infer the meaning of words according to the distribution of surrounding words [24].

There essentially are two main approaches to constructing word vectors. On the one hand, word-count-based approaches [27] create a word-context matrix that counts the co-occurrences of words close to a target word. On the other hand, distributed methods that rely internally on neural-network-based [20] architectures create dense and low-dimensional vectors of the words in a corpus. Theoretical studies have shown that these approaches are equivalent [16].

The previous methods [12, 22, 6] assign one vector per word, making it impossible to analyze specific problems, such as polysemy or semantic change, as words vary in meaning over time. To deal with this, researchers have proposed models capable of creating dynamic word embeddings. These models assign more than one vector to a word. In the context of semantic change, Bamler and Mandt [3] proposed a probabilistic language model that creates vectors with different timestamps to track their evolution over time. In similar work,

Rudolph and Blei [23] studied the evolution of language across different diachronic corpora [15] using dynamic word embeddings. They determined better performance in these methods than classical techniques in finding patterns in how language evolves. However, when extending this analysis to social networks that contain a large number of informal expressions [7] (e.g., misspelled words, acronyms, or hashtags) that arrive continuously, dynamic models will have to be retrained on the entire dataset to analyze the new information. Retraining the model each time new information arrives is very inefficient [19, 14]. These methods following a batch approach require many passes on the complete dataset until the correct parameter values are obtained.

Dynamic approaches are trained in batch configuration. Using these methods to create word embedding from a text coming from social networks (streaming text data) presents the following problems [9]:

- To obtain the appropriate parameter values requires making several passes to the dataset. The above is not possible in data streams because they are theoretically infinite sequences of information.
- As data streams are infinite sequences of data, it is impossible to load the entire data set into memory.
- They handle the time as a discrete variable.

Researchers have proposed migrating model training to online learning approaches to address these difficulties, so it is possible to develop incremental models suitable for representing vectors from the text stream. In [7], Bravo-Marquez et. al. proposes to build incremental word vectors using the count-based approach because it is simpler to adopt than the neural network approach. The incremental vectors are created as follows:

- The vectors' dimensions correspond to the contexts of the target word, which are represented by the words surrounding the target word, obtained from a window.
- The vector's numerical values represent the association between the target word and its contexts. These values are calculated using Positive Point-wise Mutual Information (PPMI) [27, 8].
- The resulting vectors are stored in memory.
- When new data arrives from the flow, the vectors are updated.

There are also two similar, but independently developed works that propose to create incremental word representations from neural networks. These models are based on the Skip-Gram model with Negative Sampling (SGNS) [20] proposed by Nikolov. The first is the Incremental SGNS [14] and was proposed by Kaji and Kobayashi. On the other hand, the second is named space-saving word2vec [19] and was proposed by May et. al. Both researchers develop incremental versions of the SGNS model that can be applied to a streaming environment. The main differences are listed below:

1. Dynamic vocabulary: to maintain a dynamic vocabulary, the incremental SGNS uses the Misra-Gries [21] algorithm and space-saving word2vec, which uses the Space-Saving algorithm developed by the same researchers.
2. Negative Sample Method: space-saving word2vec uses standard reservoir sampling to estimate an un-smoothed negative sampling distribution. The authors of incremental SGNS develop a modified reservoir sampling algorithm to estimate a smoothed negative sampling distribution.

Incremental techniques present innovative methods for creating word vectors in a streaming scenario. The researchers executed performance comparisons of incremental versus static methods (i.e., SGNS [20]), obtaining a higher performance in incremental techniques [19, 14]. It seems challenging to compare incremental techniques because of a lack of standardization of data sets, implementation of these techniques, and evaluation criteria. The aforementioned difficulties are due to the fact that these are recent techniques and because the evaluation criteria applied for the static scenario cannot be applied to the streaming case.

Problem Statement

In recent years, researchers have developed techniques for creating incremental word representations [7, 14, 19] applied to streaming text data (theoretically infinite text sequences). These techniques can represent word embedding of text from real-time data, which is an advantage over conventional techniques (which follow a batch learning approach) because it is possible to perform real-time analysis.

However, since these are recent techniques and operate on a streaming configuration, there is a lack of standardization of data sets, implementation of these techniques, and transparent evaluation criteria that measure the performance of incremental representation techniques, e.g. [7]. The conventional criteria operates on batch-type approaches [10] in which one has access to the complete data set. In our case, this is not possible because the data streams, as stated above, are possibly infinite text sequences.

We aim to produce an unified framework that adapts and implements the incremental word embedding techniques proposed by the state-of-art to standardize and define evaluation criteria that measure the performance of incremental representation methods.

Research questions

General question

Is it possible to create common-ground for existing incremental word vector models, that allows for a clean comparison and evaluation?

Specific questions

We hypothesize that implanting a framework will allow us to answer many research questions

1. Which method is more efficient (memory consumption)?
2. Which method performs better in each evaluation ask?
3. Which method adapts faster to semantic change?

Hypothesis

We hypothesize that it is possible to adapt and implement the incremental word embedding techniques found in state-of-art [7, 19, 14] to standardize the evaluation criteria, among these techniques, to operate a unified framework.

Main Goal

The general objective of this thesis is to design and build a framework capable of implementing different word vector representations based on incremental word embedding applied to streaming text data.

Specific goals

- Compare the latest incremental word embedding techniques proposed by state-of-art [7, 19, 14].
- Research and define the best evaluation metrics compare the performance using common metrics among the related works.
- Design and build a package that brings together the latest incremental word embedding techniques proposed applied to streaming text data.
- Track the semantic change of words through the streaming of text data.

Methodology

The methodology of the thesis work can be divided into the following sections:

Research

The first step of the research is to explore the state-of-the-art methods that develop incremental word embedding applied streaming text data. A significant literature review has been done, but further analysis and learning from these previous works must be achieved.

Incremental Word Representation Framework

The development of the framework considers three steps. First, a thorough review of the theory behind the techniques of incremental sparse word vectors [7] and incremental word embedding [14, 19, 17] methodologies proposed by the researchers. The second is the design and implementation of the above techniques considering that they are applied to text data streams and an online learning approach, using the open-source library River[1], specializing in Online Machine Learning techniques. The third is to define evaluation techniques for the incremental methods mentioned in this paper, and one idea is to create synthetic data streams to evaluate the performance of the models during the testing stage. Bravo-Marquez et. al. [7] explored this in evaluating the performance of the incremental classifier for time-evolving sentiment lexicon induction.

Experimentation

Throughout all framework steps, experiments will be performed on each of the described techniques of incremental word vector representations to perform semantic tracking of words or terms in streaming text data that can be found in social networks such as Twitter.

Expected results

- A short overview of the state-of-art methods (last 5 years) about incremental word embedding above data streams.
- An open-source library to facilitate the comparison between all different incremental word embedding techniques.

[1] https://github.com/online-ml/river

Bibliography

[1] Aggarwal, C. C. *Machine learning for text*. Springer, 2018.

[2] Almeida, F., and Xexéo, G. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069* (2019).

[3] Bamler, R., and Mandt, S. Dynamic word embeddings. In *International conference on Machine learning* (2017), PMLR, pp. 380–389.

[4] Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press, 2018. `https://moa.cms.waikato.ac.nz/book/`.

[5] Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. *Machine learning for data streams: with practical examples in MOA*. MIT press, 2018.

[6] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5* (2017), 135–146.

[7] Bravo-Marquez, F., Khanchandani, A., and Pfahringer, B. Incremental word vectors for time-evolving sentiment lexicon induction. *Cognitive Computation* (2021), 1–17.

[8] Church, K., and Hanks, P. Word association norms, mutual information, and lexicography. *Computational linguistics 16*, 1 (1990), 22–29.

[9] Fontenla-Romero, Ó., Guijarro-Berdiñas, B., Martinez-Rego, D., Pérez-Sánchez, B., and Peteiro-Barral, D. Online machine learning. In *Efficiency and Scalability Methods for Computational Intellect*. IGI Global, 2013, pp. 27–54.

[10] Ghannay, S., Favre, B., Esteve, Y., and Camelin, N. Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (2016), pp. 300–305.

[11] Goldberg, Y. Neural network methods for natural language processing. *Synthesis lectures on human language technologies 10*, 1 (2017), 1–309.

[12] Goldberg, Y., and Levy, O. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[13] Hamilton, W. L., Leskovec, J., and Jurafsky, D. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096* (2016).

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

[15] Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. Diachronic word embeddings and semantic shifts: a survey. *arXiv preprint arXiv:1806.03537* (2018).

[16] Levy, O., and Goldberg, Y. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems 27* (2014), 2177–2185.

[17] Luo, H., Liu, Z., Luan, H., and Sun, M. Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), pp. 1687–1692.

[18] Martinc, M., Novak, P. K., and Pollak, S. Leveraging contextual embeddings for detecting diachronic semantic shift. *arXiv preprint arXiv:1912.01072* (2019).

[19] May, C., Duh, K., Van Durme, B., and Lall, A. Streaming word embeddings with the space-saving algorithm. *arXiv preprint arXiv:1704.07463* (2017).

[20] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[21] Misra, J., and Gries, D. Finding repeated elements. *Science of computer programming 2*, 2 (1982), 143–152.

[22] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[23] Rudolph, M., and Blei, D. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052* (2017).

[24] Sahlgren, M. The distributional hypothesis. *Italian Journal of Disability Studies 20* (2008), 33–53.

[25] Shalev-Shwartz, S., and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[26] Stewart, I., Arendt, D., Bell, E., and Volkova, S. Measuring, predicting and visualizing short-term change in word representation and usage in vkontakte social network. In *Proceedings of the International AAAI Conference on Web and Social Media* (2017), vol. 11.

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

[28] Wang, Y., Hou, Y., Che, W., and Liu, T. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics* (2020), 1–20.