Decentralized Word2Vec Using Gossip Learning *

Abdul Aziz Alkathiri[†] **Lodovico Giaretta**[†] **Šarūnas Girdzijauskas**[†] **Magnus Sahlgren**[‡]
[†] KTH Royal Institute of Technology
[‡] RISE Research Institutes of Sweden
{aatba,lodovico,sarunasg}@kth.se sahlgren@ri.se

Abstract

Advanced NLP models require huge amounts of data from various domains to produce high-quality representations. It is useful then for a few large public and private organizations to join their corpora during training. However, factors such as legislation and user emphasis on data privacy may prevent centralized orchestration and data sharing among these organizations. Therefore, for this specific scenario, we investigate how gossip learning, a massively-parallel, data-private, decentralized protocol, compares to a shared-dataset solution. We find that the application of Word2Vec in a gossip learning framework is viable. Without any tuning, the results are comparable to a traditional centralized setting, with a reduction in ground-truth similarity scores as low as 4.3%. Furthermore, the results are up to 54.8% better than independent local training.

1 Introduction

Machine learning models, and especially deep learning models (LeCun, 2015) used to represent complex systems, require huge amounts of data. This is also the case with large-scale Natural Language Processing (NLP) models. Moreover, these models benefit from merging various sources of text from different domains to obtain a more complete representation of the language.

For this reason, a small number of separate organizations (for example, government agencies) may want to train a complex NLP model using the combined data of their corpora to overcome the limitations of each single corpus. However, the typical solution in which all data is moved to a centralized system to perform the training may not be viable, as that could potentially violate privacy laws or data collection agreements and would require all organization to trust the owner of the system with access to their data.

This problem can potentially be solved using massively-parallel, data-private, decentralized approaches – that is, distributed approaches where training is done directly on the machines that produce and hold the data, without having to share or transfer it and without any central coordination – such as gossip learning (Ormándi et al., 2013).

Therefore, we seek to investigate, in the scenario of a small group of large organizations, how models that are produced from the corpus of each node on a decentralized, fully-distributed, data-private configuration, i.e. gossip learning, compare to models trained using a traditional centralized approach where all the data are moved from local machines to a data center. Furthermore, we investigate how these models compare to models trained locally using local data only, without any cooperation.

Our results show that the Word2Vec (Mikolov et al., 2013b) models trained by our implementation of gossip learning are close to models produced by its centralized counterpart setting, in terms of quality of the generated embeddings, and vastly better than what simple local training can produce.

2 Background and related work

The main technique for massively-parallel, data-private training is federated learning (Yang et al., 2019), a centralized approach where each worker node calculates an update of the model based on local data. This gradient is then sent back to the central node which aggregates all these gradients to produce an updated global model which is sent

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813162. The content of this paper reflects the views only of their author (s). The European Commission/ Research Executive Agency are not responsible for any use that may be made of the information it contains.

back to the workers. This approach, however, suffers from issues such as the presence of a central node which may act as a privileged "gatekeeper", as well as reliability issue on the account of that central node.

Unlike centralized approaches, with decentralized machine learning all the nodes in the network execute the same protocols with the same level of privileges, mitigating chances of exploitation by malicious actors. Furthermore, with a peer-to-peer network protocol, decentralized machine learning can virtually scale up to unlimited sizes and be more fault-tolerant, as the network traffic is spread out across multiple links, and not all directed to a single central location. One such approach is the gossip learning protocol (Ormándi et al., 2013).

The gossip communication approach refers to a set of decentralized communication protocols inspired by the behaviour of the spread of gossip socially among people (Shah, 2009). First introduced for the purpose of efficiently synchronizing distributed servers (Demers et al., 1987), it has also applied to various problems, such as data aggregation (Kempe et al., 2003) and failure detection (Van Renesse et al., 1998).

3 Gossip Learning

Gossip learning is an asynchronous, data-parallel, decentralized averaging approach based on gossip communications. It has been shown to be effective when applied to various ML techniques, including binary classification with support vector machines (Ormándi et al., 2013), k-means clustering (Berta and Jelasity, 2017) and low-rank matrix decomposition (Hegedűs et al., 2016). However, these implementations of gossip learning are limited to simple scenarios, where each node holds a single data point and network communications are unrestricted. Giaretta and Girdzijauskas (2019) showed that the gossip protocol can be extended to a wider range of more realistic conditions. However, they identify issues with certain conditions that appear in some real-world scenarios, such as bias towards the data stored with faster communication speeds and the impact of network topologies on the convergence speed of models.

Algorithm 1 shows the general structure of gossip learning as introduced by Ormándi et al. (2013). Intuitively, models perform random walks over the network, merging with each other and training on local data at each node visited.

Algorithm 1: Generic Gossip Learning.

$m_{cur} \leftarrow$ INITMODEL()
$m_{last} \leftarrow m_{cur}$
loop
  WAIT (Δ)
  $p \leftarrow$ RANDOMPEER()
  SEND(p, m_{cur})
end loop
procedure ONMODELRECEIVED(m_{rec})
  $m_{cur} \leftarrow$ UPDATE(MERGE(m_{rec}, m_{last}))
  $m_{last} \leftarrow m_{rec}$
end procedure

Each node, upon receiving a model from a peer, executes ONMODELRECEIVED. The received model m_{rec} and the previous received model m_{last} are averaged weight-by-weight. The resulting model is trained on a single batch of local data and stored as m_{cur}. At regular intervals, m_{cur} is sent to a random peer.

We simulate gossip learning on a single machine, using synchronous iterations. This approximation works well under the assumption that all nodes have similar speeds. If that is not the case, additional measures must be taken to ensure correct model behaviour (Giaretta and Girdzijauskas, 2019).

4 Methodology

While gossip learning could be applied to most NLP algorithms, in this work we use Word2Vec (Mikolov et al., 2013a) because it is simple, small, and fast, thus allowing us to perform larger experiments on limited hardware resources. Additionally, it is a well-known, well-understood technique, allowing us to more easily interpret the results.

The dataset used is the Wikipedia articles dump (Wikimedia Foundation, 2020) of more than 16GB, which contains over 6 million articles and in wikitext format with embedded XML metadata. From this dump we extract the articles belonging to the following 5 Wikipedia categories of similar size: *science*, *politics*, *business*, *humanities* and *history*.

To measure the quality of the word embeddings produced by a specific model, we collect the $k = 8$ closest words to a target word w_t according to said model. We then assign to each of these words a score based on their *ground-truth* cosine similarity to w_t. We repeat this process for a set of (contextually ambiguous) target words W_t ($|W_t| = 23$) and use the total sum as the quality of the model. We estimate the ground-truth word similarities using a high-quality reference model, more specifically a state of the art Word2Vec model trained on the

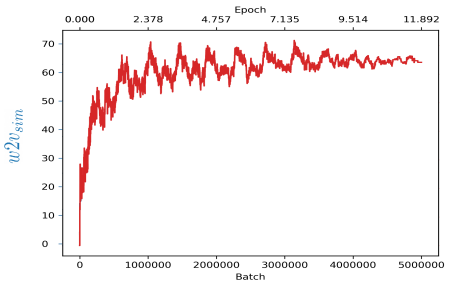


Figure 1. $w2v_{sim}$ evolution for centralized training.

Google News dataset, which uses a similar embedding size ($d = 300$) and contains a vocabulary of 3 million words (Google Code Archive, 2013).

This metric can be defined as

$$w2v_{sim}(M) = \sum_{w_t \in W_t} \sum_{w \in N_M^k(w_t)} sim_R(w, w_t)$$

where M is the model to be evaluated, $N_M^k(\cdot)$ is the top-k neighbourhood function over the embeddings of M and sim_R is the *ground-truth* cosine similarity measure defined based on the reference model.

5 Experimental results

To establish the baseline to compare to, the first experiment is in the traditional non-distributed, centralized configuration of Word2Vec. The baseline $w2v_{sim}$ value is 64.479, as shown in Figure 1.

We simulate gossip learning with 10 nodes, with three different data distributions. In the *r-balanced* distribution, the corpora of the nodes have similar sizes and are randomly drawn from the dataset. In the *r-imbalanced* distribution, the corpora are similarly drawn at random, but have skewed sizes (up to a 4:1 ratio). Finally, in the *topicwise* distribution, the dataset is divided between the nodes based on the 5 Wikipedia categories, with two nodes splitting each category.

The intuition behind dividing the texts by topic is that often times the corpora of organizations are limited to a specific domain. And setting imbalanced content sizes in one of the distributions can provide insights into how the learning is affected when some nodes have significantly bigger corpora than others. Both these configurations are very relevant to the practical applicability of this work, as they both reflect common real-world scenarios.

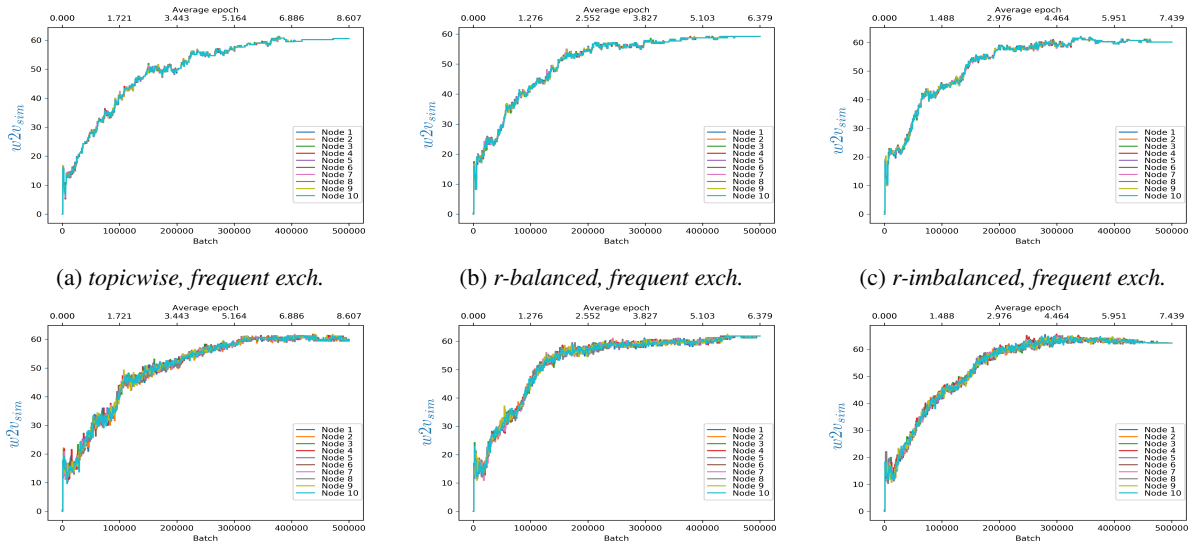
Exchange frequency	**Data distribution**	$w2v_{sim}$	$w2v_{sim}$ **reduction w.r.t. baseline**
Frequent	*topicwise*	60.606	6.390%
	r-balanced	59.936	7.580%
	r-imbalanced	60.122	7.247%
Infrequent	*topicwise*	61.840	4.267%
	r-balanced	60.910	5.859%
	r-imbalanced	60.968	5.759%

Table 1. Summary of $w2v_{sim}$ scores for all tested gossip learning configurations.

The formulation of gossip learning presented in Section 3 requires the nodes to exchange their models after every local batch update. As complex NLP models can require millions of training batches, the communication overheads can quickly add up. We thus investigate the effect of reducing the exchange frequency while still maintaining the same number of training batches. More precisely, we repeat the same tests but limit the nodes to exchange the models every 50 batch updates, thus reducing overall communication by a factor of 50.

Figure 2 shows the evolution of the trained models for all combinations of exchange frequency and data distribution. Table 1 summarizes the final scores and compares them to the baseline. In all combinations, the model quality is quite comparable to the traditional centralized configuration. In fact, for the gossip learning with infrequent exchange configuration, there is a slight improvement over the frequent exchange in terms of training time required and $w2v_{sim}$ value. This indicates that the original gossip learning formulation has significant margins of optimization in terms of communication overhead. Furthermore, the relatively unchanged values of $w2v_{sim}$ between the data distributions, in spite of the heterogeneity/homogeneity of the node contents and their sizes, show that gossip learning is robust to topicality and local dataset size. The results suggest that the quality of word embeddings produced using gossip learning is comparable to what can be achieved by training in a traditional centralized configuration using the same parameters, with a loss of quality as low as 4.6% and never higher than 7.7%.

We perform one more experiment, in which each node independently trains a model on its local data only, using the *topicwise* distribution. The $w2v_{sim}$ values do not converge as quickly and range from 41.657 to 56.570 (see Figure 3). This underscores the importance for different organizations to collab-



(a) *topicwise, frequent exch.* (b) *r-balanced, frequent exch.* (c) *r-imbalanced, frequent exch.*

(d) *topicwise, infrequent exch.* (e) *r-balanced, infrequent exch.* (f) *r-imbalanced, infrequent exch.*

Figure 2. Evolution of $w2v_{sim}$ similarity scores for all tested data distributions and exchange frequencies.

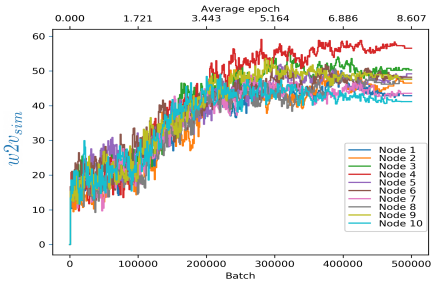


Figure 3. Local, independent training at each node: $w2v_{sim}$ similarity score evolution.

orate to overcome the specificity of local corpora, as this can increase model quality by as much as 54.8%.

6 Limitations and future work

Although the experimental setup of this research takes into account parameters and conditions which simulate real-world scenarios, it is still limited in scope. For instance, the network conditions were assumed to be perfect. Furthermore, security and privacy considerations in the area of networking were not taken into account. Although they were not the focus of this research, their significance cannot be overlooked. Investigating the behaviour of the proposed solution in more realistic network conditions is therefore a possible avenue of research.

A single, simple NLP algorithm (Word2Vec) was evaluated in this work. This is due to the purpose of this research, which was to test the viability of gossip learning and compare it to a centralized solution in a specific scenario. Evaluating more recent, contextualized NLP models, such as BERT (Devlin et al., 2019) would be an interesting research direction, as these can better capture the different meanings of the same words in multiple domains.

Finally, the experiments were run without extensive hyperparameter optimization. Given the satisfactory results obtained, it is likely that a proper tuning, based on state of the art distributed training research (Shallue et al., 2018), could lead to gossip learning matching or even surpassing the quality of traditional centralized training.

7 Conclusions

Motivated by the scenario where various organizations wish to jointly train a large, high-quality NLP model without disclosing their own sensitive data, the goal of this work was to test whether Word2Vec could be implemented on top of gossip learning, a massively-parallel, decentralized, data-private framework.

The quality of the word embeddings produced using gossip learning is close to what can be achieved in a traditional centralized configuration using the same parameters, with a loss of quality as low as 4.3%, a gap that might be closed with more advance tuning. The frequency of model exchange, which affects bandwidth requirements, has also been reduced 50 times without negative effects. Finally, gossip learning can achieve up to 54.8% better quality than local training alone, motivating

the need for joint training among organizations.

The results of this work therefore show that gossip learning is a viable solution for large-scale, data-private NLP training in real-world applications.

References

Árpád Berta and Márk Jelasity. 2017. Decentralized management of random walks over a mobile phone network. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 100–107. IEEE.

Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lodovico Giaretta and Šarūnas Girdzijauskas. 2019. Gossip learning: off the beaten path. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1117–1124. IEEE.

Google Code Archive. 2013. 3top/word2vec-api.

István Hegedűs, Árpád Berta, Levente Kocsis, András A Benczúr, and Márk Jelasity. 2016. Robust decentralized low-rank matrix decomposition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4):1–24.

David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE.

Yann LeCun. 2015. Yoshua bengio, and geoffrey hinton. *Deep learning. nature*, 521(7553):436–444.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Róbert Ormándi, István Hegedűs, and Márk Jelasity. 2013. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 25(4):556–571.

Devavrat Shah. 2009. Network gossip algorithms. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3673–3676. IEEE.

Christopher J. Shallue, Jaehoon Lee, Joseph M. Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. 2018. Measuring the effects of data parallelism on neural network training. *CoRR*, abs/1811.03600.

Robbert Van Renesse, Yaron Minsky, and Mark Hayden. 1998. A gossip-style failure detection service. In *Middleware'98*, pages 55–70. Springer.

Wikimedia Foundation. Wikipedia dump at https://dumps.wikimedia.org/backup-index.html [online]. 2020.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19.