

Privacy-Preserving Text Classification on BERT Embeddings with Homomorphic Encryption

Garam Lee^{*1} Minsoo Kim^{*2} Jai Hyun Park^{*2}

Seung-won Hwang^{†2} Jung Hee Cheon^{1,2}

¹CryptoLab ²Seoul National University

garamlee@cryptolab.co.kr

{minsoo9574, jhyunp, seungwonh, jhcheon}@snu.ac.kr

Abstract

Embeddings, which compress information in raw text into semantics-preserving low-dimensional vectors, have been widely adopted for their efficacy. However, recent research has shown that embeddings can potentially leak private information about sensitive attributes of the text, and in some cases, can be inverted to recover the original input text. To address these growing privacy challenges, we propose a privatization mechanism for embeddings based on homomorphic encryption, to prevent potential leakage of any piece of information in the process of text classification. In particular, our method performs text classification on the encryption of embeddings from state-of-the-art models like BERT, supported by an efficient GPU implementation of CKKS encryption scheme. We show that our method offers encrypted protection of BERT embeddings, while largely preserving their utility on downstream text classification tasks.

1 Introduction

In recent years, the increasingly wide adoption of vector-based representations of text such as BERT, eLMo, and GPT (Devlin et al., 2019; Peters et al., 2018; Radford et al., 2019), has called attention to the privacy ramifications of embedding models. For example, Coavoux et al. (2018); Li et al. (2018) show that sensitive information such as the authors' gender and age can be partially recovered from an embedded representation of text. Song and Raghunathan (2020) report that BERT-based sentence embeddings can be inverted to recover up to 50%–70% of the input words.

Previously proposed solutions such as differential privacy based on perturbation/noise (Qu et al., 2021), require manually controlling the noise injected into embeddings, to control the privacy-utility trade-off to a level suitable for each downstream task. In this work, we propose a privacy

solution based on *Approximate Homomorphic Encryption*, which is able to achieve little to no accuracy loss of BERT embeddings on text classification¹, while ensuring a desired level of encrypted protection, i.e. 128-bit security.

Homomorphic Encryption (HE) is a cryptographic primitive that serves computations over encrypted data without any decryption process. While previous works have focused on homomorphic computation where the inputs are numerical data, in applications such as privacy-preserving machine learning algorithms (Lauter, 2021), logistic regression (Kim et al., 2018), and neural network inference (Gilad-Bachrach et al., 2016), they have rarely been applied to *unstructured* data such as text. Recent works in this direction include Podschwadt and Takabi (2020), who conduct sentiment classification over encrypted word embeddings using RNN. However, they use a simple embedding layer which maps words in a dictionary to real-valued vectors, and model training is only supported on plaintext. The most closely related work to ours is PrivFT (Badawi et al., 2020), a homomorphic encryption based method for privacy preserving text classification built on fastText (Joulin et al., 2017).

We next describe our approach, focusing on our distinctions from PrivFT:

- **BERT Embedding-based Method:** The principle behind PrivFT is to perform all neural network computations in encrypted state. For this purpose, it adopts fastText (Joulin et al., 2017), which takes bag-of-words vectors as input, followed by a two-layer network and an embedding layer. However, PrivFT does not utilize pre-training; as a consequence, the embedding matrix and classifier of PrivFT must be updated from scratch, taking several days to train on a single dataset.

We introduce a new method for text classification on encrypted data. The crux is to operate a simple downstream classifier on encryptions

^{*}Equal contribution.

[†]Corresponding author.

¹Code and data are available at: <https://www.github.com/mnskim/hebert>

of semantically rich vector representations (i.e. BERT embeddings). By using rich input representations, our method significantly outperforms PrivFT, while the use of a simple downstream classifier on encrypted data makes our method much more practical. Importantly, by leveraging pre-trained embeddings from models such as BERT, a state-of-the-art in many NLP tasks, our method enables the training of a strong classifier in encrypted state within hours. As such, our method is well positioned to take full advantage of the recent trends in NLP, that rely on the language understanding capability of increasingly larger pre-trained language models (Brown et al., 2020; Kaplan et al., 2020).

- **Better GPU Implementation:** As BERT representations are real-valued vectors, we adopt CKKS scheme, which is well-suited for dealing with real numbers compared to other HE schemes. We develop an efficient GPU implementation of CKKS which greatly improves computation speed. While PrivFT also provides a GPU implementation of CKKS, their implementation lacks the *bootstrapping* operation of CKKS. Inevitably, this limits the multiplicative depth of PrivFT, and it makes the method less scalable. It also results in the use of less secure CKKS parameters which have roughly 80-bit security level. In contrast, our GPU implementation includes the bootstrapping operation, which allows unlimited number of multiplications. This enables us to use a higher degree polynomial approximation (which is key to achieving a high downstream accuracy), and more secure CKKS parameters (128-bit security level²). Moreover, with practicality in mind, we improved the implementation in terms of communication cost. More precisely, we introduce a practical implementation of CKKS to significantly reduce the size of ciphertexts by more than $7.4\times$ compared to the rudimentary implementation.

We experimentally validate our approach on text classification datasets, showing that it offers encrypted protection of embedding vectors, while maintaining utility competitive to plaintext on downstream classification tasks. Additionally, we compare our method with PrivFT on homomorphic training on encrypted data, showing it outperforms PrivFT, with much improved training efficiency.

²An attacker needs $> 2^{128}$ operations to recover the plaintext from a ciphertext with the current best algorithm.

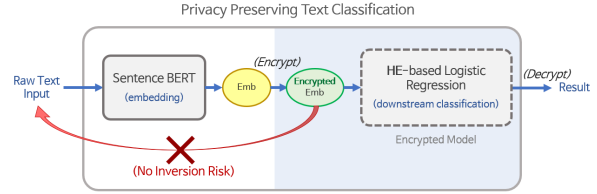


Figure 1: The Architecture of Text Classification. The region shaded in light blue represents encrypted state. The privatization inference takes the following steps: 1) User generates sentence embedding. 2) User encrypts embedding. 3) Logistic regression in encrypted state is performed using encrypted embedding.

2 Method

We focus on the scenario in which the user directly applies the privacy mechanism to the output embeddings from a neural text encoder, before passing it on to a service provider for usage in a downstream task. This is also referred to as a local privacy setting (Qu et al., 2021).

The privatization procedure M_{priv} in the local privacy setting can be defined as follows:

$$M_{priv}(x) = P(F_{emb}(x)) \quad (1)$$

where x is the raw text input, F_{emb} is Sentence-BERT (Reimers and Gurevych, 2019)³, a popular pre-trained model for obtaining sentence embeddings, and P denotes a privacy mechanism. Next, we securely classify the text datum, x , by feeding its privatized embedding, $M_{priv}(x)$, to the downstream classification model. In this work, we adopt a logistic regression model (in encrypted state) as the downstream classifier. Figure 1 demonstrates the entire privatization inference procedure, starting with user’s embedding of raw text and encryption of embedding, to the operation of the classifier in encrypted state and finally, the output of encrypted classification results. We note that the training process also can be performed in encrypted state as we describe in Section 2.2.

2.1 Baseline: Local Differential Privacy

As a baseline for the local privacy setting, we implement noise-based local differential privacy (LDP). Qu et al. (2021) introduced noise-based LDP for single-token embeddings as privatization mechanism P . In the case of single-token embeddings, local differential privacy can be achieved with respect to a chosen Euclidean distance by adding randomly

³<https://www.sbert.net/>

Table 1: Size of Ciphertext/Training Time of Encrypted Logistic Regression. We report the ciphertext size and training time for the ciphertext model. Note that plaintext and LDP classifiers have negligible training and inference times.

	Twitter (2 classes)			SNIPS (7 classes; OvR)		
	ciphertext	plaintext	training time	ciphertext	plaintext	training time
train	1.4 GB	183.7 MB	143.2 sec/epoch	11.4 GB	206.5 MB	1111.4 sec/epoch

sampled noise N drawn from an n -dimensional distribution with density $p(N) \propto \exp(-\eta\|N\|)$. That is, the privacy mechanism is $P(y) = y + N$, where y denotes an embedding vector. In our work, we adopt the same mechanism to sentence embeddings. Following Qu et al. (2021), the noise $N \in \mathbb{R}^n$ is sampled as a pair (r, p) , where r is the distance from the origin and p is a point in \mathbb{B}^n (the unit hypersphere in \mathbb{R}^n). r is sampled from Gamma distribution $\Gamma(n, \frac{1}{\eta})$ and p is sampled uniformly over \mathbb{B}^n , and N is computed as $N = rp$.

2.2 HE Based Logistic Regression

We now describe our proposed privatization mechanism in detail. We adopt Eq.1, with the privacy mechanism $P(y) = H(y)$, where H is the homomorphic encryption. For downstream tasks, we feed the privatized embedding $M_{priv}(x)$ to an encrypted logistic regression classifier. By utilizing HE, an encrypted model and labels will be obtained after the training and inference process. Only the user who knows the secret key of HE can decrypt the results and get either the classifier or labels for the classification.

For the homomorphic encryption H , we adopt the CKKS scheme (Cheon et al., 2017, 2018, 2019). While the majority of HE schemes are being optimized for computations over finite fields, CKKS supports efficient computations over *real numbers*, so it is advantageous in application to real world data. We refer the readers to the paper (Cheon et al., 2017) for full details of CKKS.

CKKS is a levelled homomorphic encryption scheme, where the *level* of each ciphertext indicates the remaining number of times we can operate.⁴ When we multiply two ciphertexts of level l , the output ciphertext has a level of $l - 1$. Once the level of a ciphertext becomes too low, we can refresh its level higher by using the *bootstrapping*

⁴We remark that this *level* is not related to the security level of CKKS. For example, in our implementation, all ciphertexts have 128-bit security level regardless of the remaining number of operations.

technique so that the number of possible operation times increases. For ciphertexts ct_1 and ct_2 of complex vector messages m_1 and m_2 , we summarize the operations of CKKS as follows:

- **Add**(ct_1, ct_2): output a ciphertext of $m_1 + m_2$.
- **Mult**(ct_1, ct_2): output a ciphertext of $m_1 \odot m_2$, where \odot is the entry-wise multiplication.
- **Bootstrap**(ct_1): output a ciphertext of m_1 at refreshed level.

While it is prevalent to encrypt data into the top level, L , in this work, we encrypt the data into a lower level, 3, to decrease the initial size of ciphertexts.⁵ Note that the ciphertexts are the privatized embeddings, so their size determines the communication cost. As shown in Table 1, by using the lower level ciphertexts, we reduce the initial size of ciphertext by more than $7.4\times$ in both Twitter training dataset (10.8GB to 1.4GB), and SNIPS training dataset (85.3GB to 11.4GB).

Finally, we feed the output of our privatization mechanism to the next step, the training and inference of an encrypted logistic regression classifier. However, since CKKS supports only addition and multiplication while the logistic function $(1/(1 + \exp(-x)))$ is a non-polynomial function, we evaluate the logistic function via its polynomial approximation. We use the minimax approximate polynomial (Pachon and Trefethen, 2009) of degree 15 on $[-12, 12]$ that approximates the logistic function within the error of 0.00614 on $[-12, 12]$.

2.3 Datasets

To validate our approach in real-world scenarios, we conduct experiments on tasks with realistic privacy concerns and utility needs. We select text classification tasks on data in three settings:

- **Tweets Hate Speech Detection** (Sharma, 2018)⁶: Is a crowd-sourced dataset of Tweets for binary classification, where labels denote a Tweet as containing hate speech, if it has a racist or sexist sentiment associated with it. We created random data splits for train/validation/test, with 11,634/3,197/4,795 examples, respectively.
- **SNIPS** (Coucke et al., 2018): Is a dataset of crowd-sourced queries collected from the

⁵We use $L = 29$ in our implementation.

⁶https://huggingface.co/datasets/tweets_hate_speech_detection

Table 2: Results of Logistic Regression Experiments. For SNIPS, macro average of F1 over classes is reported and AUC denotes the average of each class AUC. Bold and underline denote the ciphertext model and the noising model most comparable to it (measured by absolute difference of metric), respectively.

Model	Twitter					SNIPS			
	(Thresh)	Dev F1	Test F1	Dev AUC	Test AUC	Dev F1	Test F1	Dev AUC	Test AUC
Noising									
$\eta = 50$	0.5149	0.3809	0.3337	0.7975	0.7991	0.7291	0.6944	0.9190	0.9106
$\eta = 75$	0.5300	0.5226	0.4680	0.8847	0.8819	0.8818	0.8524	0.9689	0.9621
$\eta = 100$	0.4997	0.5744	0.5323	0.9098	0.9105	0.9279	0.8990	0.9826	0.9776
$\eta = 125$	0.4555	0.6107	0.5760	0.9226	0.9245	0.9422	0.9190	0.9931	0.9844
$\eta = 150$	0.4843	0.6224	0.5939	0.9234	0.9332	0.9547	0.9303	0.9953	0.9879
$\eta = 175$	0.5128	<u>0.6404</u>	<u>0.6065</u>	<u>0.9300</u>	<u>0.9390</u>	<u>0.9616</u>	<u>0.9345</u>	<u>0.9955</u>	<u>0.9900</u>
Ciphertext	0.8635	0.6596	0.6361	0.9481	0.9535	0.9729	0.9402	0.9974	0.9948
Plaintext	0.4987	0.6625	0.6439	0.9536	0.9575	0.9787	0.9520	0.9977	0.9959

Snips Voice Platform, distributed among 7 user intents. It has been widely adopted in evaluating spoken language understanding (SLU) systems. We use the same data splits as [Goo et al. \(2018\)](#); [Qin et al. \(2019\)](#), with 13,084/700/700 examples, respectively.

- **Youtube Spam Collection** ([Badawi et al., 2020](#))⁷: Is a public data set collected for spam research from UCI Machine Learning Repository, where five datasets are composed by 1,956 real messages extracted from five videos. As train/validation/test splits are not provided, we created our own random splits, with 1,564/196/196 examples, respectively.

3 Experiments

3.1 Encrypted Sentence Classification

Once sentence embeddings are extracted from Sentence-BERT for each input text, the vectors consist of 768 numerical values of 32-bit floating point from -1 to 1. Then, a logistic regression model is trained for binary classification on the Twitter dataset, and multiclass classification on the SNIPS dataset, respectively.

To perform a fair comparison of the results of each approach, we keep the same implementation of logistic regression for plaintext, as that of the ciphertext model. Multiclass classification is performed as multiple separate binary logistic regression models for each class, and we take the argmax from the combined results; One-vs-Rest (OvR). Experiments for noise-based local differential privacy on plaintext are conducted in the same way, using the privacy mechanism described in Section 2.1.

⁷<https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection>

Logistic regression parameters are optimized by SGD with Nesterov momentum. For all models, the best performing model and optimal threshold for F1 was identified by validation set performance.

For plaintext experiments, the following hyperparameters were used for training: Learning rate 3.0, gamma 0.9, batch size 256 for Twitter dataset, and learning rate 3.0, gamma 0.1, batch size 128 for SNIPS dataset. Both models were trained for 10 epochs. For the parameters of the CKKS scheme, we selected the dimension $N = 2^{17}$ and set the size of the maximum modulus q_L to be 1540 bits. We note that our CKKS parameters satisfy 128-bit security level ([Albrecht, 2017](#)). For ciphertext experiments, the following hyperparameters were used for training: Learning rate 3.0, gamma 0.9, batch size 512 for Twitter dataset, and learning rate 2.0, gamma 0.1, batch size 512 for SNIPS dataset. Both models were trained for 10 epochs. Additionally, we developed an efficient parallelized CKKS implementation for bootstrapping with GPU acceleration for the encrypted logistic regression model. For implementation, we use a dual-NVLink Nvidia Quadro RTX6000 GPU with 24 GiBs of memory, on a server with a Intel Xeon Gold 6242R CPU (80 core) and 125 GiBs of RAM.

3.2 Embedding Inversion

As a quantitative evaluation of inversion risk, we adopt sentence embedding inversion. Introduced in [Song and Raghunathan \(2020\)](#), embedding inversion is an adversarial attack whose goal is to recover the original text (its tokens) from its embedding. In this work, we focus on *black-box* inversion, where the adversary can only interact with the model by querying it to obtain embeddings, and is therefore more pertinent to real-world privacy considerations.

Table 3: Sentence Embedding Inversion. Black-box inversion of sentence embeddings on SNIPS. We report F1 for the task of recovering the input words from the sentence embedding. Ciphertext denoted in bold.

Model	(Thresh)	Dev F1	Test F1
Noising			
$\eta = 50$	0.8	0.2082	0.1905
$\eta = 75$	0.8	0.3078	0.2955
$\eta = 100$	0.9	0.3587	0.3276
$\eta = 125$	0.9	0.4164	0.3899
$\eta = 150$	0.9	0.4572	0.4337
$\eta = 175$	0.85	0.4919	0.4803
Ciphertext	-	-	-
Plaintext	0.85	0.6705	0.6759

4 Results

We report the results of our logistic regression experiments in Table 2. We compare our approach, denoted as Ciphertext, with the Plaintext baseline, as well as the noise-based LDP from Qu et al. (2021), at different levels of the noise parameter η (smaller indicates larger noise). Measured by F1/AUC metric, our HE classifier achieves roughly 98.79%/99.58% and 98.76%/99.89% of the plaintext baseline classifier’s performance on Twitter and SNIPS test sets, respectively, indicating that our HE of embeddings is able to preserve their downstream utility to a significant degree. We find that our model performs better at all noise levels considered in Qu et al. (2021) (up to $\eta = 175$), nearly matching the plaintext model. On the other hand, for noise-based LDP, we observe a clear trade-off between increasing (via decreasing η) privacy protection and classification performance. As can be seen with $\eta = 50, 75$ on both datasets, the decrease in performance becomes greater as η becomes smaller and privacy protection is prioritized. Moreover, at any reasonable level of η , noise-based LDP cannot necessarily guarantee the complete elimination of inversion risk.

We next perform sentence embedding inversion experiments on SNIPS. In Table 3, we report the results at varying levels of η , using black-box inversion with a multi-label classification model as in Qu et al. (2021). For plaintext, the degree of inversion risk is consistent with black-box inversion results from Song and Raghunathan (2020), who report F1 of 59.76 for inverting BERT-based sentence embeddings⁸, indicating a high degree of invertibility. Our results show that, in order to significantly

⁸Trained using Sentence-BERT objective on BookCorpus and Wikipedia data.

Table 4: Comparison to PrivFT. We report wallclock training time and test accuracy of binary spam classification. Ciphertext model results are denoted in bold.

Model	PrivFT	Ciphertext	Plaintext
(Num. GPUs)	8	1	-
Training time	60.48 hrs/epoch	23.04 sec/epoch	-
(Thresh)	-	0.53	0.51
Test accuracy	0.863	0.908	0.913

reduce inversion risk, noise-based LDP requires low η settings, sacrificing downstream utility. In contrast, our method eliminates conventional risk of black-box inversion: Because all results of HE inference remain encrypted, and cannot be revealed without decryption with the user’s secret key, black-box inversion cannot be applied. Therefore, 128-bit security level of homomorphic encryption guarantees practically complete protection from inversion, while offering significantly improved performance.

Finally, to directly compare our model with PrivFT, we conduct an experiment on the YTSC dataset, following the methodology in Section 3. We report the results in Table 4, along with PrivFT results on the same dataset from Badawi et al. (2020). We measure the test accuracy of the classifier, as well as the wallclock time required to perform encrypted training. We find that our method requires only 460.81 seconds with a single GPU to achieve 90.8% test accuracy, whereas PrivFT needs 5.04 days with 8 GPUs to obtain 86.3% test accuracy. This amounts to roughly $\times 9,450$ faster training per epoch, while achieving higher accuracy and utilizing $1/8^{\text{th}}$ the number of GPUs. These results experimentally validate our expectation that homomorphic encryption of pretrained embeddings significantly improves performance and efficiency.

5 Conclusion

We propose a privatization mechanism based on homomorphic encryption which, by leveraging BERT pre-trained embeddings, enables efficient training of an HE logistic regression classifier with little to no loss of downstream utility. While our method compares favorably to noise-based LDP and PrivFT, we also note that there are some limitations. Since HE based models require higher computation costs compared to plaintext models, the challenge remains to adopt more complex models, such as neural networks, as downstream classifiers. Nevertheless, the privacy benefits and efficiency of our method makes it a suitable candidate for scenarios with real-world privacy concerns.

Acknowledgements

We would like to show our gratitude to Younggi Lee and Seewoo Lee from CryptoLab for assistance with the experiments. We also greatly thank Jeonghwan Kim and Sungwoo Oh from KB Kookmin Bank who provided the initial research topic and suggestions for feasibility in the real-world applications.

Cheon’s team was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2020-0-00840, Development and Library Implementation of Fully Homomorphic Machine Learning Algorithms supporting Neural Network Learning over Encrypted Data, 50%]. Hwang’s team was supported by Microsoft Research Asia and IITP [(2022-00155958, High Potential Individuals Global Training Program) and (NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), 50%].

References

- Martin R. Albrecht. 2017. A Sage Module for estimating the concrete security of Learning with Errors instances. <https://bitbucket.org/malb/lwe-estimator>.
- Ahmad Al Badawi, Louie Hoang, Chan Fook Mun, Kim Laine, and Khin Mi Mi Aung. 2020. **Privft: Private and fast text classification with homomorphic encryption**. *IEEE Access*, 8:226544–226556.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. **Language models are few-shot learners**. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2018. Bootstrapping for approximate homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2018*, pages 360–384, Cham. Springer International Publishing.
- Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2019. A full rns variant of approximate homomorphic encryption. In *Selected Areas in Cryptography – SAC 2018*, pages 347–368, Cham. Springer International Publishing.
- Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham. Springer International Publishing.
- Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. 2018. **Privacy-preserving neural representations of text**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. **Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces**. *CoRR*, abs/1805.10190.
- 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.
- Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. **Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy**. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 201–210, New York, New York, USA. PMLR.
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. **Slot-gated modeling for joint slot filling and intent prediction**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. **Scaling laws for neural language models**. *CoRR*, abs/2001.08361.

- Andrey Kim, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Cheon. 2018. [Logistic regression model training based on the approximate homomorphic encryption](#). *BMC Medical Genomics*, 11.
- Kristin E. Lauter. 2021. Private ai: Machine learning on encrypted data. Cryptology ePrint Archive, Report 2021/324. <https://ia.cr/2021/324>.
- Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. [Towards robust and privacy-preserving text representations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, Melbourne, Australia. Association for Computational Linguistics.
- Ricardo Pachon and Lloyd Trefethen. 2009. [Barycentric-remez algorithms for best polynomial approximation in the chebfun system](#). *BIT*, 49:721–741.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Robert Podschwadt and Daniel Takabi. 2020. Classification of encrypted word embeddings using recurrent neural networks. In *PrivateNLP@ WSDM*, pages 27–31.
- Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. [A stack-propagation framework with token-level intent detection for spoken language understanding](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.
- Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. [Natural language understanding with privacy-preserving bert](#). In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM '21*, page 1488–1497, New York, NY, USA. Association for Computing Machinery.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Roshan Sharma. 2018. Sentimental analysis of tweets for detecting hate/racist speeches.
- Congzheng Song and Ananth Raghunathan. 2020. [Information leakage in embedding models](#). In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20*, page 377–390, New York, NY, USA. Association for Computing Machinery.