Unsupervised Contextualized Document Representation

Ankur Gupta
Indian Institute of Technology, Kanpur
ankugupt@iitk.ac.in

Vivek Gupta
School of Computing, University of Utah
vgupta@cs.utah.edu

Abstract

Several NLP tasks need the effective representation of text documents. Arora et al., 2017 demonstrate that simple weighted averaging of word vectors frequently outperforms neural models. SCDV (Mekala et al., 2017) further extends this from sentences to documents by employing soft and sparse clustering over pre-computed word vectors. However, both techniques ignore the polysemy and contextual character of words. In this paper, we address this issue by proposing SCDV+BERT(ctxd), a simple and effective unsupervised representation that combines contextualized BERT (Devlin et al., 2019) based word embedding for word sense disambiguation with SCDV soft clustering approach. We show that our embeddings outperform original SCDV, pre-train BERT, and several other baselines on many classification datasets. We also demonstrate our embeddings effectiveness on other tasks, such as concept matching and sentence similarity. In addition, we show that SCDV+BERT(ctxd) outperforms fine-tune BERT and different embedding approaches in scenarios with limited data and only few shots examples.

1 Introduction

The semantics of a document is highly dependent on the constituent words, and words can have different meaning in different contexts. Approaches such as Socher et al., 2013; Liu et al., 2015a; Le and Mikolov, 2014; Ling et al., 2015 go beyond words to capture the semantic meaning of sentences but are restricted to perceiving the meaning of a single sentence, thus reducing its expressive power.

A simple weighted average of the individual word embeddings doesn't account for word ordering and long-distance semantic relationships. Gupta et al., 2016; Mekala et al., 2017 proposed clustering-based technique with tf-idf weighting to form sparse composite document vector, thus extending the simple averaging approach beyond a single sentence. Recently, Gupta et al., 2020a introduced SCDV-MS, which highlights how multi-sense word embedding resolves cluster disambiguation, which improves embedding performance, further enhancing SCDV. Gupta et al., 2020b (PSIF) additionally demonstrates that a sparsity constraint in clustering can be advantageous.

Modern contextualized representations such as BERT (Devlin et al., 2019) can capture the exact meaning of a word based on the surrounding context, which can automatically disambiguate the meaning of words in a corpus-based on its interpretations. Previous approaches for document representation, as discussed above, ignore these contextualized representation benefits. Therefore, in this work, we propose a new approach (SCDV+BERT(ctxd), which leverages a combination of clustering techniques for word-sense disambiguation and combines it with the expressibility of SCDV partition averaging method for creating better document representation. We contextualized the corpus by using the word sense disambiguation power contextualized pre-train BERT embedding. Contextual embedding from pre-train BERT disambiguate the occurrence of the same word with different context (meaning). We then use the SCDV partition averaging algorithm to convert this contextualized corpus into document embeddings using contextual pre-train BERT word embeddings as static word vectors.

We show that our unsupervised embeddings SCDV+BERT(ctxd) outperform existing techniques on several classification datasets in supervised, semi-supervised, and few-shot settings. We also demonstrate performance improvement in non-classification tasks such as concept matching and sentence similarity using our representation. The datasets along with the associated scripts, can be located at https://github.com/vgupta123/contextualize_scdv.

Proceedings of the 2nd Workshop on Simple and Efficient Natural Language Processing, pages 166–173
November 10, 2021. ©201 Association for Computational Linguistics

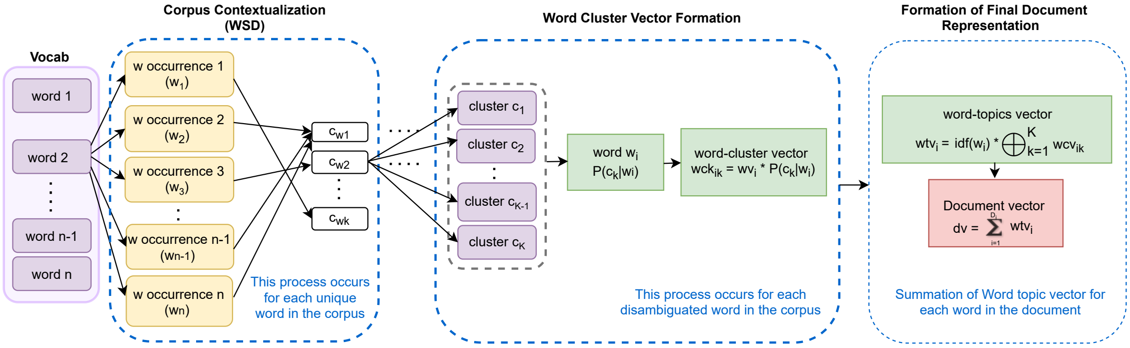
Corpus Contextualization (WSD)
Word Cluster Vector Formation
Formation of Final Document Representation
Vocab
word 1
word 2
word n-1
word n
w occurrence 1 (w_1)
w occurrence 2 (w_2)
w occurrence 3 (w_3)
w occurrence n-1 (w_{n-1})
w occurrence n (w_n)
c_{w1}
c_{w2}
c_{wk}
This process occurs for each unique word in the corpus
cluster c_1
cluster c_2
cluster c_{K-1}
cluster c_K
word w_i $P(c_k|w_i)$
word-cluster vector $wck_{ik} = wv_i * P(c_k|w_i)$
This process occurs for each disambiguated word in the corpus
word-topics vector
$wtv_i = idf(w_i) * \bigoplus_{k=1}^{K} wcv_{ik}$
Document vector
$dv = \sum_{i=1}^{D_i} wtv_i$
Summation of Word topic vector for each word in the document

Figure 1: High level flowchart of our document representation approach a.k.a SCDV + BERT(cxtd)

2 Proposed Algorithm

Our algorithm is similar to SCDV algorithm, but uses word sense disambiguated contextualize word vectors obtained from pre-trained BERT Embedding as static word vectors. Below are details of the primary steps involved.

Corpus Contextualization (WSD): Our objective here is to disambiguate different occurrences of a word in a corpus document with its separate interpretation. For example, the word '*bank*' in *"He went to 'bank' for withdrawing money"* and *"I went to a river 'bank' during summer holiday"* has different meanings based on it's used context. Given a word w and all its occurrence in the corpus documents as $w_1, w_2, \ldots, w_n$, for each w_i , we find its contextualized embedding representation b_{w_i} using transformer based pre-trained language model such as BERT (Devlin et al., 2019).

Taking inspiration from Mekala et al., 2020, we treat the word disambiguation problem as a local clustering problem of contextualizing word vectors. For our case, we also cluster the contextualize word embedding b_{w_i}, obtained via the pre-train BERT model from the last step. We use the efficient K-means algorithm to cluster all b_{w_i} into k clusters, where k represents the total possible interpretations of word w in all the documents of the corpus.[1] We use the cosine distance between the contextualize word representation as our clustering metric. [2] The value of k i.e., cluster numbers, is decided using a similarity threshold (τ), which is a hyperparameter and a dataset property, usually set using the heuristic described in (Mekala et al., 2020).

Let $c_{w_1}, c_{w_2}, \ldots, c_{w_k}$ be the k cluster centroids obtained after the K-means clustering for the word w. We treat these k centroid representation as our polysemous word representations highlighting the k sense of the words w. After clustering for each occurrence of word w in a corpus, we perform contextualised word sense disambiguation using cosine similarity between it's BERT representation and our centroid embedding i.e. $c_{w_1}, c_{w_2}, \ldots, c_{w_k}$ to discover the closest cluster centroid j a.k.a nearest neighbour (j), the word sense for that occurrence.

Finally, we assign this nearest neighbor cluster centroids embedding c_{w_j} as the contextualized disambiguated word embedding for that word w occurrence. We repeat this procedure for all the occurrences of word w to obtain final sense disambiguated contextualized word embedding. Each of these contextualized embeddings of word w now act as our distant sense-disambiguated word vectors.

Document Representation (SCDV): Similar to the SCDV, we use Gaussian Mixture Model to cluster our final sense-disambiguated word vectors (obtain from earlier step) into K partition of the words in the corpus.[3] For each contextualized word vector w_i of word $w \in V$, we created K different word-cluster vectors of d dimensions ($wc\vec{v}_{io}$) by weighting word's embedding with sparse probability distribution for the o^{th} cluster, i.e., P($c_o|w_i$). Then, we concatenate the K word-cluster vectors ($wc\vec{v}_{io}$) and weight them with their inverse document frequency (idf(w_i)) to form a contextualized word-topic vector ($w\vec{tv}_i$). The dimension of word-topic vector ($w\vec{tv}_i$) is $K \times d$. To obtain the final document embedding $d\vec{v}_{D_n}$, we computed the average of the contextualized word-topic vectors $w\vec{tv}_i$ from the words and it's context as appearing in that document D_n. For more details on SCDV, refer to Algorithm 1 in Mekala et al., 2017.

Figure 1 shows flowchart for our document em-

[1] K-means is computationally more efficient than other clustering approaches. Any other suitable clustering algorithm also works.

[2] L2 on normalized vectors is same as the cosine distance.

[3] Note this capital K is very different from the small k use in the word sense disambiguation. The value of K depend of number of distinct high-level concepts covered in the corpus.

Embedding	**Amazon**	**BBCSport**	**Twitter**	**Classic**	**Recipe-L**	**20NG**
SIF(Glove)	$94.1_{(0.2)}$	$97.3_{(1.2)}$	$57.8_{(2.5)}$	$92.7_{(0.9)}$	$71.1_{(0.5)}$	$72.3_{(0.11)}$
PV-DM	$88.6_{(0.4)}$	$97.9_{(1.3)}$	$67.3_{(0.3)}$	$96.5_{(0.7)}$	$71.1_{(0.4)}$	$74.0_{(0.11)}$
Doc2VecC	$91.2_{(0.5)}$	$90.5_{(1.7)}$	$71.0_{(0.4)}$	$96.6_{(0.4)}$	$76.1_{(0.4)}$	$78.2_{(0.11)}$
Word2Vec (idf-weighted)	$94.00_{(0.45)}$	$97.30_{(0.67)}$	$72.00_{(0.36)}$	$95.20_{(0.44)}$	$74.90_{(0.89)}$	$81.70_{(0.22)}$
BERT(pr)	$91.04_{(0.27)}$	$99.12_{(0.66)}$	$66.63_{(0.22)}$	$95.63_{(0.36)}$	$68.44_{(0.07)}$	$64.81_{(0.17)}$
SCDV + Word2Vec	$93.90_{(0.40)}$	$98.81_{(0.60)}$	$74.20_{(0.40)}$	$96.90_{(0.10)}$	$78.50_{(0.50)}$	$84.90_{(0.13)}$
SCDV + BERT(weight-avg)	$94.62_{(0.21)}$	$97.29_{(0.56)}$	$72.98_{(0.24)}$	$96.54_{(0.61)}$	$78.13_{(0.15)}$	$84.90_{(0.13)}$
SCDV + BERT(ctxd)	$94.16_{(0.31)}$	$99.58_{(0.41)}$	$75.98_{(0.36)}$	$97.84_{(0.40)}$	$80.71_{(0.19)}$	$86.12_{(0.11)}$
SCDV + BERT(ctxd) + Anisotropy	$\mathbf{95.88}_{(0.34)}$	$\mathbf{99.60}_{(0.59)}$	$\mathbf{77.03}_{(0.27)}$	$\mathbf{99.01}_{(0.41)}$	$80.74_{(0.15)}$	$\mathbf{86.94}_{(0.11)}$
BERT (fine-tune)	$94.60_{(0.19)}$	$\mathbf{99.67}_{(0.51)}$	$73.13_{(0.31)}$	$\mathbf{98.67}_{(0.56)}$	$\mathbf{81.13}_{(0.21)}$	$\mathbf{86.91}_{(0.28)}$

Table 1: Embedding performance with complete training i.e. full data setting. Bold represents best performance. Reported number are means performance and subscript brackets number x $_{(x)}$ represent the standard deviation over 5 random runs. Baselines are taken from Gupta et al., 2020b. We use $k = 6$ for the ainsotropic adjustment.

bedding approach (SCDV+BERT(cxtd)).

3 Experimental Results

We perform experiments with multi-class classification with data restriction settings, concept matching problem and sentence similarity task.

Datasets and Baselines: We experimented on 6 classification dataset whose statistics are shown in 4. We also validated our algorithm through the Concept Matching experiment on Concept-Project (Gong et al., 2018) dataset, where the task was to generate concepts from a given document corpus. We also perform experiments on the SemEval dataset (Y12 - Y16) involving 27 semantic textual similarity (STS) tasks from 2012 - 2016 (Agirre et al., 2012). We represent our model as SCDV + BERT(ctxd), which is SCDV using multi sense-disambiguated contextual BERT word vector for document representation. We consider the following baselines for comparison: SCDV with single sense Word2Vec (Mikolov et al., 2013), BERT(pr) (Devlin et al., 2019) i.e. pre-trained BERT vectors averaging, BERT (fine-tune)(Sun et al., 2019) i.e. fine-tune BERT model, and Word2Vec (tfidf-weighted) i.e. a tf-idf weighted Word2Vec average. We also compare SCDV + BERT(ctxd) with an ablation representation obtain without corpus contextualization a.k.a, sense-disambiguation i.e. the value of WSD clustering parameter $k = 1$ for all the words in the corpus. We referred this ablation representation as SCDV + BERT(weight-avg) in the paper. [4] Furthermore, we also adopted the work of Ethayarajh, 2019 which adjust Anisotropy (more uniformity) with our embeddings. We refer this representation as SCDV + BERT(ctxd) + Anisotropy. We followed Ethayarajh, 2019 recommendation and used BERT last layer for pre-train word embeddings.

Full Data Setting: Table 1 shows a comparison of accuracy across all the datasets. We observe that SCDV + BERT(ctxd) model outperforms the original SCDV+Word2Vec across all the datasets. To ablate the contribution of sense-disambiguation using BERT contextualization, we also compare SCDV + BERT(ctxd) result with SCDV + BERT(weight-avg). We also analyze the effect of reducing Anisotropy on SCDV + BERT(ctxd).[5]

Analysis: Contextualized BERT (SCDV + BERT(ctxd)) performs better than the average BERT (SCDV + BERT(weight-avg)), which in turn performs competitively with original SCDV + Word2Vec. This indicates that the sense-disambiguated BERT based word vectors captures multiple meaning of word (better corpus contextualization) effectively. [6]. Furthermore, reducing Anisotropy i.e. SCDV + BERT(ctxd) + Anisotropy, further boosts the performance of SCDV + BERT(ctxd). As expected, the BERT (fine-tune) performs the best on most datasets (except for Twitter and Amazon), where all versions of SCDV perform much better than pre-train BERT averaging i.e. BERT(pr). The good performance of BERT (fine-tune) is expected as fine-tuning modifies the model parameter (layer weights), producing task and domain-specific embedding, often impressed in the *[CLS]* token representation. Note that SCDV + BERT(ctxd)+ Anisotropy are unsupervised embedding but had accuracy competitive (sometimes even better) to the supervised fine-tuned BERT models i.e. BERT (fine-tune). Thus, our approach could be

[4] Model hyperparameters are provided in appendix §A.

[5] For top 1000 words, the cosine similarity reduces from 0.5468 to 0.3752 after anisotropic adjustment.

[6] Similar observation made by Gupta et al. 2020a (SCDV-MS), an extension for SCDV with multi-sense word2Vec.

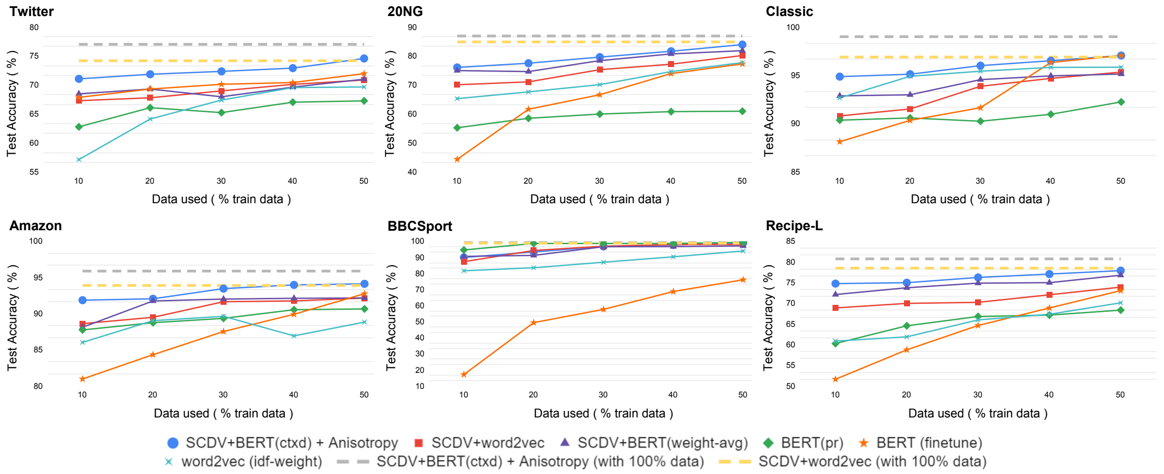

Figure 2: Embedding performance with limited training data i.e. semi-supervised setting. Standard deviation avg: 0.28 with range (0.03,0.81).

used as an effective alternative to fine-tuning BERT for selected classification tasks.

Limited Data Setting: BERT has produced state-of-the-art results in various NLP domains, but its use is restricted to the availability of labeled data, whereas with the help of pre-trained BERT and SCDV, our approach can also work with limited data i.e. semi-supervised setting, requiring only sufficient enough labeled data to learn the downstream classifier. To test the effectiveness of our approach in low-data conditions, we ran the same multi-class classification experiment with 10%, 20%, 30%, 40%, and 50% of the training data. See Figure 2 for the results.

Analysis: Contrast to earlier fully supervised settings, we observe that BERT (fine-tune) performed worst (except for Twitter data) due to limited training data. BERT (fine-tune) have a significant mean performance drop of greater than 33.5% across datasets with limited data training of 10% data compare to full training with 100% data. Whereas, the performance of SCDV + BERT(ctxd) (and it's Anisotropic version i.e., SCDV + BERT(ctxd) + Anisotropic) remained comparable with the full data (i.e., 100% data) setting, with the mean performance drop of just ≈ 7.2% across datasets with the 10% data. [7] It also outperformed original SCDV + Word2Vec and SCDV+BERT(weighted-avg) on all datasets. [8] Under low data conditions, SCDV+BERT(weight-avg) also outperformed the original SCDV + Word2Vec due to added contextualization benefits of the pre-trained BERT word vector representation. Moreover, under significantly less data (10% or 20% data), even BERT(pr) and Word2Vec (tdf-idf weighted) performed significantly better than the BERT(fine-tune). Furthermore, we find that our method outperforms SCDV with complete training on a range of datasets while only employing limited 40% to 50% data. Overall, our unsupervised method significantly outperforms fine-tune BERT in the low data domain. We predict that fine-tuning BERT with little supervision is exceedingly difficult due to the needed learning of large-parameter space.

Few-Shot Setting: We also experimented under few-shot conditions where the available data is too low for even training a classification (and obviously for BERT fine-tuning i.e. BERT(fine-tune) too). We implemented a K-shot N-way prototypical few-shot classifier where K is the number of samples used from each class and N is the number of classes in the dataset. We set the K values from 5, 10, 15, and 20 for our setting, and N is equal to numbers of class labels (see Appendix §A). New examples are assigned the label using the nearest neighbor approach (KNN algorithm with K = 1), i.e., the label of closest averaged prototypical class point.

Analysis: We see in Figure 3 that SCDV + BERT(ctxd) and SCDV+BERT(weight-avg) outperform all other methods by a significant margin (> 11% mean across datasets). SCDV+BERT(ctxd) for most of the time marginally leads SCDV+BERT(weighted-avg), showing that contextualization also helps in the few shots settings. In contrast to earlier results adjusting anisotropy over SCDV + BERT(ctxd) only improve performance marginally.

[7] For some dataset such as BBCSport the performance drop was also < 1%.

[8] Except the BBCSport, where pre-trained BERT embeddings, i.e., BERT(pr), produced comparable results due to fewer polysemous words as evident from Appendix §A Table 1.

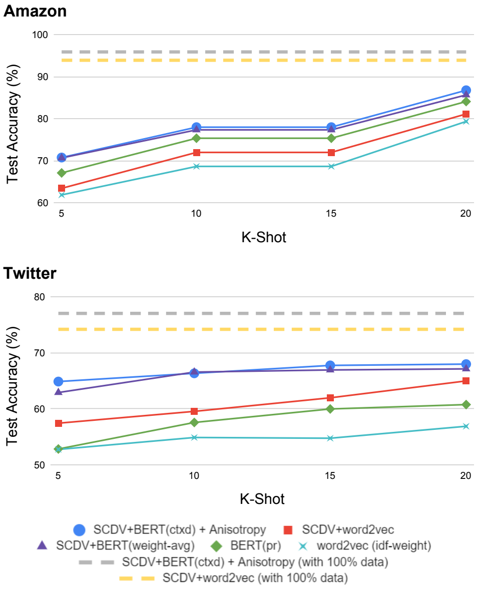

Figure 3: Results for the few-shot experiment. Reported number are mean of 5 random runs with having standard deviation average: 0.14 with range of (0.09, 0.43)

Concept Matching: The task is to link the concept with the relevant projects. Concept Matching dataset includes 537 pairs of projects and concepts involving 53 unique concepts from the Next Generation Science Standards3 (NGSS) and 230 unique projects from Science Buddies. We compare cosine similarity between our method with the TF-IDF-weighted vectors, SCDV + Word2Vec, InferSent (Conneau et al., 2017) and pre-trained BERT(pr) baselines. From table 2, we observed that our algorithm (SCDV + BERT(ctxd) + Anisotropy) outperformed pre-trained BERT and SCDV + Word2Vec by 5.2% and 4.6% on F1 and accuracy respectively on the Concept-Project (Gong et al., 2018) dataset.

Embedding	Accuracy	F1
TF-IDF	53.8	70.0
InferSent	54.0	70.1
BERT(pr)	$54.8_{(0.2)}$	$70.6_{(0.3)}$
SCDV + Word2Vec	$53.7_{(0.1)}$	$70.0_{(0.1)}$
SCDV + BERT(ctxd)	$57.1_{(0.2)}$	$73.8_{(0.2)}$
SCDV + BERT(ctxd) + Anisotropy	$\mathbf{58.9}_{(0.1)}$	$\mathbf{74.6}_{(0.1)}$

Table 2: Embedding performance on Concept-Marching dataset. Bold represents best performance. Baselines are taken from Zhang and Danescu-Niculescu-Mizil, 2020.

Sentence Similarity Task: The objectives of these tasks are to predict the similarity between two sentences. Performance is assessed by computing the Pearson correlation (Freedman et al., 2007) between machine-assigned semantic similarity scores and ground truth. SCDV + BERT(ctxd) + Anisotropy substantially outperform several other baselines as demonstrated in the table 3.

Embedding	Y12	Y13	Y14	Y15	Y16	Avg.
ELMO orig+all	55	51	63	69	64	60.4
ELMO orig+top	54	49	62	67	63	59
BERT(pr) Avg.	53	67	62	73	67	64.4
USE	65	**68**	64	77	73	69.4
p-mean	54	52	63	66	67	60.4
fastText	58	58	65	68	64	62.6
Skip Thoughts	41	29	40	46	52	41.6
InferSent	61	56	68	71	77	66.6
PSIF + PSL	65.7	64.0	74.8	77.3	73.7	71.1
u-SIF + PSL	65.8	65.2	75.9	77.6	72.3	71.4
SCDV + WordVec	64.1	63.9	73.0	76.9	**77.3**	71.0
SCDV + BERT(ctxd)	64.7	**64.0**	75.4	77.1	73.3	70.9
SCDV + BERT(ctxd) + Anisotropy	**66.8**	64.1	**77.3**	**78.0**	74.6	**72.2**

Table 3: Embedding performance on Semantic Textual Similarity task (STS) with several embeddings for each year with overall average (avg.). Bold represent best performance. Baselines are taken from Gupta et al., 2020b.

4 Comparison with Related Works

The closest work to our paper is SCDV by Mekala et al., 2017 that extend BoWV by Gupta et al., 2016 using an overlapping clustering technique and direct idf weighting of word vectors. Recently Gupta et al., 2020a extended SCDV to SCDV-MS via utilising the multi-sense embeddings obtained via using AdaGram (Bartunov et al., 2016) over Word-Vectors (Mikolov et al., 2013). Our idea is similar to SCDV-MS; however, it utilises pre-train BERT contextual embedding as word embedding, a.k.a a more robust sense disambiguated aware embedding (Mekala et al., 2020). Thus, our approach (SCDV + BERT(ctxd)) uses contextual word vectors as the foundational block for document representation.

5 Conclusion and Future Work

In this paper, we enhance sparse document representation (SCDV) with pre-trained BERT contextualization and propose SCDV+BERT(ctxd). We showed that one could effectively utilize the BERT contextualization for word-sense disambiguation. Our approach outperforms other unsupervised approaches in the full data regime. Our approach is also very successful for low data regime, outperforming the standard model with roughly half the training data required and few shot settings, where fine-tuning of model fails.

6 Acknowledgement

We appreciate the Utah NLP group members' valuable suggestions at various phases of the project, as well as the reviewers' helpful remarks.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations, ICLR 2017*.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 130–138, Cadiz, Spain. PMLR.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Zhiyuan Chen and Bing Liu. 2017. *Topic Models for NLP Applications*, pages 1276–1280. Springer US, Boston, MA.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Felix Gers, Nicol Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3:115–143.

Hongyu Gong, Tarek Sakakini, Suma Bhat, and JinJun Xiong. 2018. Document similarity for texts of varying lengths via hidden topics. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2341–2351, Melbourne, Australia. Association for Computational Linguistics.

Vivek Gupta, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. 2016. Product classification in E-commerce using distributional semantics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 536–546, Osaka, Japan. The COLING 2016 Organizing Committee.

Vivek Gupta, Ankit Saw, Pegah Nokhiz, Harshit Gupta, and Partha Talukdar. 2020a. Improving document classification with multi-sense embeddings. In *Proceedings of the European Conference on Artificial Intelligence*.

Vivek Gupta, Ankit Saw, Pegah Nokhiz, Praneeth Netrapalli, Piyush Rai, and Partha Talukdar. 2020b. P-sif: Document embeddings using partition averaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Han Kyul Kim, Hyun joong Kim, and S. Cho. 2017. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352.

Stefan Kombrink, Tomas Mikolov, Martin Karafiát, and Lukas Burget. 2011. Recurrent neural network based language modeling in meeting recognition. pages 2877–2880.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of Word2Vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2015a. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1284–1290.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015b. Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2418–2424. AAAI Press.

Dheeraj Mekala, Vivek Gupta, Bhargavi Paranjape, and Harish Karnick. 2017. SCDV : Sparse composite document vectors using soft clustering over distributional representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 659–669, Copenhagen, Denmark. Association for Computational Linguistics.

Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. META: Metadata-empowered weak supervision for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8351–8361, Online. Association for Computational Linguistics.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Dat Quoc Nguyen, Kairit Sirts, and Mark Johnson. 2015. Improving topic coherence with latent feature word representations in MAP estimation for topic modeling. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 116–121, Parramatta, Australia.

Matthew 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.

Pranjal Singh and Amitabha Mukerjee. 2015. Words are not equal: Graded weighting model for building composite document vectors. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 11–19, Trivandrum, India. NLP Association of India.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2078–2088.

Justine Zhang and Cristian Danescu-Niculescu-Mizil. 2020. Balancing objectives in counseling conversations: Advancing forwards or looking backwards. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5276–5289, Online. Association for Computational Linguistics.

A Hyper parameter Details

We obtain the word embeddings using BERT-base-uncased pre-trained model and use k-means for contextual clustering for given word. For simplicity we used similarity threshold (τ) of 0.8 for all words in all the datasets (see datasets details in Table 4) which lead to multiple polysemous word representation for each word, the distribution for the same is shown in Table 5. For SCDV, we set the dimension of word embeddings to 200 and the number of mixture components in GMM between $30-90$ (dataset dependent) as shown in Table 6. For the GMM we ensure that all mixture components share the same *tied co-variance* matrix. Sparsifying the document vectors further as propose in SCDV leads to only marginal performance gain (statistically insignificant), so we skip that step for our experiments. We used LinearSVM for multi-class classification during fully supervised and semi-supervised settings and prototypical networks for few-shot setting. The choice of the classifier was the same in all baselines and the proposed model to maintain uniformity. We used 5-fold cross-validation on the F1 score to tune parameter C of LinearSVM. In semi-supervised setting the example are added in incremental setting for fair comparison. We also repeated the experiment 10 times (varying random set selection seed) and consider mean as our final performance.

Dataset	Train Size	Test Size	#Label
20NG	11314	7358	20
Amazon	5600	2400	4
Twitter	2180	935	3
BBCSport	516	221	5
Classic	4966	2129	4
Recipe-l	27842	11932	20

Table 4: Dataset Statistics.

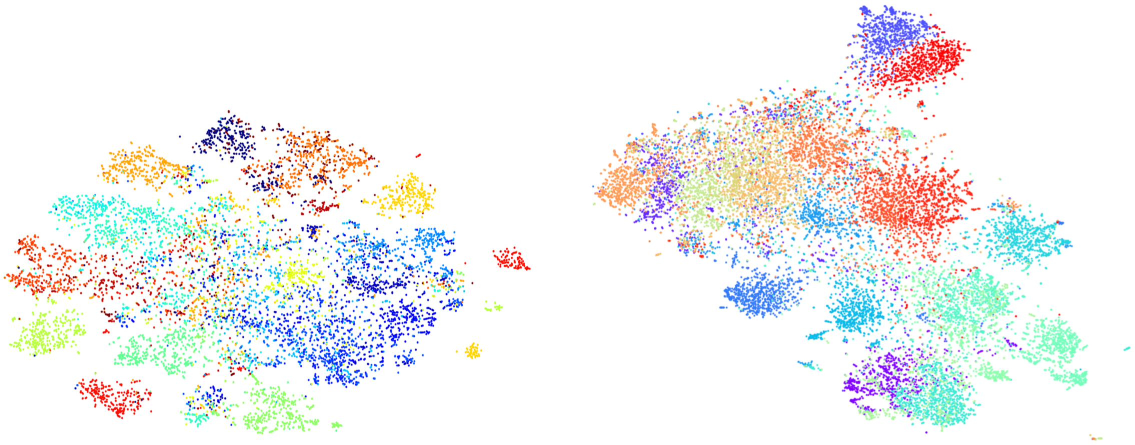

Figure 4: t-SNE plots: SCDV+WordVec(left) and SCDV+BERT(ctxd) + Anisotropy(right). Clearly, better class separation for SCDV+BERT(ctxd) + Anisotropy.

Dataset	k=1	k=2	k≥3
20NG	80.29	13.58	6.23
Amazon	76.12	17.68	6.20
Twitter	80.79	15.60	3.61
BBCSport	87.29	11.56	1.15
Classic	73.63	17.01	9.36
Recipe-l	67.11	13.98	18.91

Table 5: Distribution (in %) of vocabulary as it's disambiguated into k = 1, 2 and ≥ 3 polysemous words.

Dataset (%)	10	20	30	40	50	100
20NG	45	45	60	60	60	60
Amazon	30	30	30	30	30	30
Twitter	30	45	45	45	45	45
BBCSport	60	60	75	75	75	90
Classic	30	30	30	30	30	30
Recipe-l	30	30	30	30	30	30

Table 6: Number of mixture components in GMM used in various experiments. [9]

STS Task Details: For the STS task, the gold score is a continuous valued similarity score on a scale from 0 to 5, with 0 indicating that the semantics of the sentences are completely independent and 5 signifying semantic equivalence is computed

B Word Sense Disambiguation Examples

Table 7 shows word sense disambiguation for few polysemous words from 20NewsGroup dataset along with the cosine similarity between BERT embedding with different context usage. We use threshold of 0.8 for sense cluster disambiguation. Figure 4 shows tha t-sne plots for SCDV+WordVec(left) and SCDV+BERT(ctxd) + Anisotropy(right) embeddings. Clearly, we see much better class separation for SCDV+BERT(ctxd) + Anisotropy than SCDV+WordVec. We can alse the anisotropic reduction conical effect.

Word	Sentence	Score
Subject	The math **subject1** is difficult He sent the mail without **subject2**	0.71
Apple	The stocks of **Apple1** have increased I eat an **apple2** everyday	0.67
Unit	Metre is **unit1** of Distance He is in 1st **unit2**	0.78

Table 7: Word Sense Disambiguation. Here, score represent the cosine similarity between BERT embedding of the **bold** subject word.

C Other Related Work

Levy and Goldberg, 2014 used unweighted averaging of word vectors, Singh and Mukerjee, 2015 proposed tfidf-weighted averaging of word vectors, Socher et al., 2013 proposed a recursive neural network defined over a parse tree with supervised training. Le and Mikolov, 2014 proposed PV-DM and PV-DBoW models which treat each sentence as a shared global latent vector. Other approaches use seq2seq models such as RNN (Kombrink et al., 2011) and LSTM (Gers et al., 2002) which can handle long term dependency. Wieting and Gimpel, 2017 proposed a neural network model which optimizes the word embeddings based on the cosine similarity. Recent deep contextual word embeddings such as ELMo (Peters et al., 2018), USE (Cer et al., 2018) and BERT (Devlin et al., 2019), which capture the word context, outperform all earlier approaches. There are also other topic based modeling approaches such as LDA (Chen and Liu, 2017), weight-BoC (Kim et al., 2017), TWE (Liu et al., 2015b) , NTSG (Liu et al., 2015a), w2v-LDA (Nguyen et al., 2015), etc, as explored in SCDV (Mekala et al., 2017).