Generation-Distillation for Efficient Natural Language Understanding in Low-Data Settings

Luke Melas-Kyriazi
lmelaskyriazi@college.harvard.edu

George Han
hanz@college.harvard.edu

Celine Liang
cliang@college.harvard.edu

Abstract

Over the past year, the emergence of transfer learning with large-scale language models (LM) has led to dramatic performance improvements across a broad range of natural language understanding tasks. However, the size and memory footprint of these large LMs makes them difficult to deploy in many scenarios (e.g. on mobile phones). Recent research points to knowledge distillation as a potential solution, showing that when training data for a given task is abundant, it is possible to distill a large (teacher) LM into a small task-specific (student) network with minimal loss of performance. However, when such data is scarce, there remains a significant performance gap between large pretrained LMs and smaller task-specific models, even when training via distillation. In this paper, we bridge this gap with a novel training approach, called *generation-distillation*, that leverages large finetuned LMs in two ways: (1) to generate new (unlabeled) training examples, and (2) to distill their knowledge into a small network using these examples. Across three low-resource text classification datsets, we achieve comparable performance to BERT while using $300\times$ fewer parameters, and we outperform prior approaches to distillation for text classification while using $3\times$ fewer parameters.

1 Introduction

Over the past year, rapid progress in unsupervised language representation learning has led to the development of increasingly powerful and generalizable language models (Radford et al., 2019; Devlin et al., 2018). Widely considered to be NLP's "ImageNet moment" (Ruder, 2018), this progress has led to dramatic improvements in a wide range of natural language understanding (NLU) tasks, including text classification, sentiment analysis, and question answering (Wang et al., 2018; Rajpurkar et al., 2016). The now-common approach for employing these systems using transfer learning is to (1) pretrain a large language model (LM), (2) replace the top layer of the LM with a task-specific layer, and (3) finetune the entire model on a (usually relatively small) labeled dataset. Following this pattern, Peters et al. (2018), Howard and Ruder (2018), Radford et al. (2019), and Devlin et al. (2018) broadly outperform standard task-specific NLU models (i.e. CNNs/LSTMs), which are initialized from scratch (or only from word embeddings) and trained on the available labeled data.

Notably, transfer learning with LMs vastly outperforms training task-specific from scratch in low data regimes. For example, GPT-2 is capable of generating coherent text in a particular style (i.e. poetry, Java code, questions and answers) when conditioned on only a handful of sentences of that style (Radford et al., 2019). Similarly, on discriminative tasks such as question answering, BERT reaches accuracies comparable to previous task-specific models with orders of magnitude less labeled data (Devlin et al., 2018).

At the same time however, these large language models are extremely unwieldy. The largest versions of GPT-2 and BERT have over 1.5B and 340M parameters, respectively; it is challenging to use either of these models on a modern GPU (with 12GB of VRAM) and nearly impossible to deploy them on mobile or embedded devices. Thus, there is a strong need for efficient task-specific models that can leverage the knowledge from large pretrained models, while remaining highly compressed.

In this project, we attempt to bridge this gap for the task of low-resource text classification. We propose a new approach, called *generation-distillation*, to improve the training of small, task-specific text classification models by utilizing

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo), pages 124–131
Hong Kong, China, November 3, 2019.
https://doi.org/10.18653/v1/P17

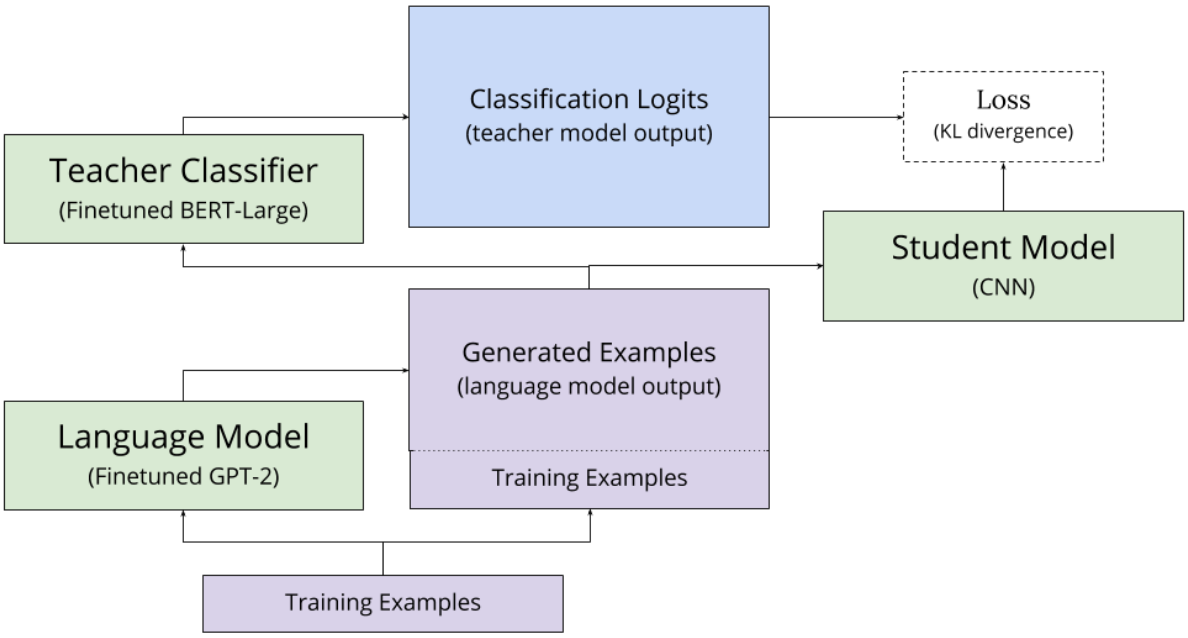

Figure 1: Our proposed generation-distillation training procedure. First, we use a large language model to augment our set of training examples, and second we train our student via distillation with a large language model-based classifier. In the diagram above, green blocks indicate models and purple blocks indicate text data.

multiple large pretrained language models. First, we use a large LM (GPT-2) to generate text in the style of our training examples, augmenting our data with unlabeled synthetic examples. Second, we use the synthetic examples to distill a second large LM (BERT), which has already been fine-tuned for classification, into a small task-specific model (CNN).

In our experiments, we show that this procedure delivers significant gains over a standard distillation approach in low-data regimes. Specifically, on low-data versions of three widely-adopted text classification datasets (AG News, DBPedia, Yahoo Answers), we obtain 98% of BERT's performance with 300× fewer parameters. Moreover, compared to prior work on distilling BERT (Chia et al., 2018) on these datasets, we outperform past approaches while using 3× fewer parameters.

2 Related Work

Designed to produce contextual word embeddings, large language models (LMs) build upon the now-classic idea of using pretrained word embeddings to initialize the first layer of deep natural language processing models (Collobert et al., 2011). Early proponents of contextual word vectors, including CoVe, ULMFit, and ELMo (McCann et al., 2017; Howard and Ruder, 2018; Peters et al., 2018), extracted word representations from the activations of LSTMs, which were pretrained for either machine translation (CoVe) or for language modeling (ULMFit, ELMo).

Recent work has adopted the transformer architecture for large-scale language representation. BERT (Devlin et al., 2018) trains a transformer using masked language modeling and next sentence prediction objectives, giving state-of-the-art performance across NLU tasks. GPT/GPT-2 (Radford et al., 2019) trains a unidirectional objective, showing the ability to generate impressively coherent text.

Due to the unwieldy size of these models, a line of recent research has investigated how to best compress these models (Tang et al., 2019). In the most popular of these approaches, knowledge distillation (Hinton et al., 2015), the outputs of a larger "teacher" model are used to train a smaller "student" model. These outputs may contain more information than is available in the true label, helping bring the performance of the student closer to that of the teacher. On the task of text classification, (Tang et al., 2019) and (Chia et al., 2018) both recently showed that it is possible to compress transformer-based LMs into

Model	*Params (1000s)*	*AG News*	*DBPedia*	*Yahoo Answers*
Baseline - TFIDF + SVM (Ramos et al., 2003)	18.1	81.9	94.1	54.5
Baseline - FastText (Joulin et al., 2016)	N/A	75.2	91.0	44.9
BERT-Large	340,000	*89.9*	**97.1**	**67.0**
Chia et al. (2018) - BlendCNN*	3617	87.6	94.6	58.3
Chia et al. (2018) - BlendCNN + *Dist**	3617	**89.9**	96.0	63.4
Ours (Kim-style)	1124	85.7	94.3	62.4
Ours (Res-style)	1091	86.2	94.7	60.9
Ours + *Dist* (Kim-style)	1124	86.9	95.0	62.9
Ours + *Dist* (Res-style)	1091	87.3	95.4	62.2
Ours + *Gen-Dist* (Kim-style)	1124	*89.9*	*96.3*	64.2
Ours + *Gen-Dist* (Res-style)	1091	89.8	96.0	*65.0*

Table 1: *(Results)* A comparison of model size and accuracy on 3 text classification datasets. Bold font indicates best accuracy and italics+underline indicates second-best accuracy. Generation-distillation broadly improves small model performance over distillation, which in turn broadly improves performance over training from scratch. * results from other papers.

CNNs/LSTMs with fewer parameters, at the cost of a small (but nontrivial) drop in accuracy.

Our project builds on prior work in multiple ways. When performing generation-distillation, we employ a finetuned GPT-2 (Radford et al., 2019) as our generator and a finetuned BERT (Devlin et al., 2018) as our teacher classifier. Additionally, the distillation component of our generation-distillation approach is similar to the method used in (Chia et al., 2018), but with a different loss function (KL divergence in place of mean absolute error).

3 Methodology

As shown in Figure 1, our *generation-distillation* approach is divided into three steps: finetuning, generation and distillation.

3.1 Finetuning

The first step in our approach involves finetuning two different large LMs on our small task-specific dataset. First, we finetune a generative model (in our case, GPT-2) using only the text of the dataset. This model is used to generate new synthetic examples in the *generation* step. Second, we finetune a large LM-based classifier (in our case, BERT with an added classification head) using both the text and the labels of the dataset. This model is used as the teacher in the *distillation* step.

3.2 Generation

In the generation step, we used a large generative LM, finetuned in the first step, to augment our training dataset with synthetic examples. Specifically, we use GPT-2 to generate new sentences in the style of our training dataset and add these to our training dataset. We do not have labels for these generated sentences, but labels are not necessary because we train with distillation; our goal in generating synthetic examples is not to improve the large LM-based classifier, but rather to improve our ability to distill a large LM-based classifier into a small task-specific classifier.

3.3 Distillation

We combine both the real training examples and our synthetic examples into one large training set for distillation. We distill a large LM-based teacher classifier, finetuned in the first step, into our smaller student model via standard distillation as in Hinton et al. (2015). For our loss function, like Hinton et al. (2015), we use the KL divergence between the teacher logits and the student logits; this differs from Chia et al. (2018), who use the mean absolute error between the logits.

4 Experiments

4.1 Data

We perform text classification on three widely-used datasets: *AG News*, *DBPedia*, and *Yahoo Answers* (Gulli; Auer et al., 2007; Labrou and Finin, 1999). For purposes of comparison, we select our training set using the same procedure as Chia et al. (2018), such that the training set contains 100 examples from each class. For the generation-distillation experiments, we use GPT-2 to generate 13600 synthetic training examples on AG News and 25000 synthetic training examples on DBPedia and Yahoo Answers. Combining these with the $400, 1400$, and 1000 original (labeled) examples yields a total of $14000, 26400$, and 26000 examples on AG News, DBPedia, and Yahoo Answers, respectively.

4.2 Finetuning Details and Examples

We finetune GPT-2 345M using Neil Shepperd's fork of GPT-2: https://github.com/nshepperd/gpt-2/blob/finetuning/train.py

Finetuning is performed for a single epoch with a learning rate of $2e-5$ with the Adam optimizer. We use batch size 1 and gradient checkpointing in order to train on a single GPU with 12GB of VRAM. We choose to train for only 1 epoch after examining samples produced by models with different amounts of finetuning; due to the small size of the dataset relative to the number of parameters in GPT-2, finetuning for more than 1 epoch results in significant dataset memorization.

For sampling, we perform standard sampling (i.e. sampling from the full output distribution, not top-p or top-k sampling) with temperature parameter $T = 1$. Although we do not use top-k or top-p sampling, we believe it would be interesting to compare the downstream effect of different types of sampling in the future.

In Supplementary Table 3, we show examples of synthetic training texts generated by sampling from the finetuned GPT-2 model, for both DBPedia and Yahoo Answers.

In Supplementary Table 4, we show two synthetic training texts along with their nearest neighbors in the training set. Nearest neighbors were calculated by ranking all examples from the training dataset (1400 examples) according to cosine similarity of TF-IDF vectors. As can be seen in the example in the right column, the GPT-2 language model has memorized some of the entities in the training dataset (i.e. the exact words "Ain Dara Syria"), but provides a novel description of the entity. This novel description is factually incorrect, but it may still be helpful in training a text classification model in a low-resource setting, because the words the model generates (i.e. "Syria", "Turkey", "Karzahayel") are broadly related to the original topic/label. For example, they may help the model learn the concept of the class "village", which is the label of Nearest Neighbor 1.

4.3 Student Models & Optimization

We experiment with two main CNN architectures. The first is a standard CNN architecture from Kim (2014). The second is a new CNN based on ResNet (He et al., 2016). This "Res-style" model has 3 hidden layers, each with hidden size 100, and dropout probability $p = 0.5$. We use multiple models to demonstrate that our performance improvements over previous approaches are not attributable to architectural changes, and to show that our approach generalizes across architectures.

We train the CNNs using Adam (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) with learning rate 10^{-3}. Additionally, the CNNs both use 100-dimensional pretrained subword embeddings (Heinzerling and Strube, 2018), which are finetuned during training.

4.4 Results

We report the performance of our trained models in Table 1.

When trained with standard distillation, our KimCNN and ResCNN models perform as would be expected given the strong results in Chia et al. (2018). Our models perform slightly worse than the 8-layer BlendCNN from Chia et al. (2018) on AG News and DBPedia, while performing slightly better on Yahoo Answers. Standard distillation improves their performance, but there remains a significant gap between the CNNs and the BERT-Large based classifier. Training with the proposed generation-distillation approach significantly reduces the gap between the CNNs and BERT-Large; across all datasets, the model trained with generation-distillation matches or exceeds both the model the model trained with standard distillation and the BlendCNN.

4.5 Ablation

In Figure 2, we show how the accuracy of the final distilled model varies with the number of syn-

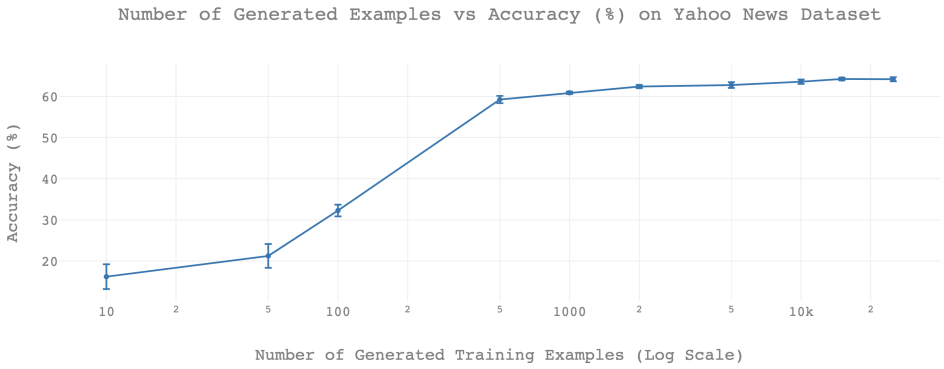

Figure 2: Above, we show how the accuracy of the final distilled model varies with the number of synthetic training examples generated by GPT-2. Error bars show the standard deviation of accuracies on five separate runs. The same GPT-2 model (trained on 100 examples per class, or a total of 1000 examples) was used to generate all synthetic texts.

Hard Labeling vs. Distillation on Generated Examples (Yahoo Answers)

	Hard Labeling with BERT	Distillation with BERT
Accuracy	62.9 $\pm$0.22	64.2 $\pm$0.13

Table 2: Above, we show a comparison of hard labeling and distillation for labeling the synthetic examples produced by our generator network. We report the the mean and standard error of the student (Kim) model accuracy across 5 random restarts on the Yahoo Answers dataset. Generation and distillation significantly outperforms generation and hard labeling.

thetic training examples generated by GPT-2. The distilled model is trained entirely on synthetic examples, without ever seeing the original data. The model shows strong performance (60% accuracy) with as few as 500 generated training examples, or 50 per class. Moreover, model performance continues to increase with more generated training examples, up to $25,000$.

In Table 2, we compare two different methods of labeling the synthetic examples produced by our generator network (GPT-2): hard labeling and distillation. Hard labeling refers to taking the maximum-probability class according to our finetuned BERT model as the label for each generated example and using a standard cross entropy loss function. Distillation refers to using the probability distribution outputted by BERT as the label for each generated examtple and using a KL divergence loss function. Put differently, in the former we use BERT to generate labels, whereas in the latter we use BERT to generate perform distillation. We find that generation and distillation outperforms generation and hard labeling by a significant margin, consistent with previous work on knowledge distillation (Hinton et al., 2015).

5 Conclusion

In this work, we present a new approach to compressing natural language understanding models in low-data regimes. Our approach leverages large finetuned language models in two ways: (1) to generate new (unlabeled) training examples, and (2) to distill their knowledge into a small network using these examples. Across three low-resource text classification datsets, we achieve comparable performance to BERT while using $300\times$ fewer parameters, and we outperform prior approaches to distillation for text classification while using $3\times$ fewer parameters. Although we focus on text classification in this paper, our proposed method may be extended to a host of other natural language understanding tasks in low-data settings, such as question answering or extractive summarization.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Yew Ken Chia, Sam Witteveen, and Martin Andrews.

2018. Transformer to cnn: Label-scarce distillation for efficient text classification.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Antonio Gulli. [link].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *ACL 2018*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yannis Labrou and Tim Finin. 1999. Yahoo! as an ontology: using yahoo! categories to describe documents. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 180–187. ACM.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL 2018*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries.

Sebastian Ruder. 2018. Nlp's imagenet moment has arrived.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Examples of Generated Training Texts

DBPedia
Landmine: Landmine[1] (also known as LNG mine) is a landmine created by the Chernobyl nuclear powerplant. It is a slurry subterranean mine typically laid in shallow pools of water. The mines are connected by run-off points and can be faced off against one another.
Naukembe Consolidated School: Naukembe is a boarder boarding and lodging school based in the township of Naushere East Sussex England. The school is a member of the N30 co-education network. The school holds around 750 students from grade six to eleven.
Peter Moldegrd: Peter Moldegrd (born 6 July 1940) is a Swedish film director known for his 1972 Melancholia. He later worked in Zurich and Hong Kong.
Ain Dara Syria: Ain Dara (Arabic: Andin Qasim Qasim; also Romanized as Andin Qs Qasim and Madd Drqt) is a small village in Doubs Governorate southwestern Syria close to the Turkey-Syria border. Nearby localities include Afrin to the north Karzahayel to the east and Siloamfara to the northwest. Ain Dara is settled by around 30 families.
Yahoo Answers
Why is America the most geographically illiterate first world country?
Where I can get program that erases voice from music track?: Where I can get program that erases voice from music track? nowhere
does anyone know the name of the song that's used in the ADIDAS commercial Jos +10? (That's adidas, by the way)?: This commercial was recently in a recent adidas commercial, and they apparently used the credits for the commercial, so I saw it and thought it was pretty cool.
What would be a good way to express how you feel about another person?: say something nice, thoughtful, creative, professional... whatever . just let it go and move on, someone else will take care of the rest

Table 3: Examples of captions generated by GPT-2 for the DBPedia and Yahoo Answers datasets. The GPT-2 model that generated these texts was trained on 100 examples per class, or a total of 1000 examples for Yahoo Answers and 1400 for DBPedia. These examples were picked randomly from all generated sentences.

Generated Training Examples and their Nearest Neighbors in the Real Training Data (DBPedia)

Generated Example	Naukembe Consolidated School: Naukembe is a boarder boarding and lodging school based in the township of Naushere East Sussex England. The school is a member of the N30 co-education network. The school holds around 750 students from grade six to eleven.	Ain Dara Syria: Ain Dara (Arabic: Andin Qasim Qasim; also Romanized as Andin Qs Qasim and Madd Drqt) is a small village in Doubs Governorate southwestern Syria close to the Turkey-Syria border. Nearby localities include Afrin to the north Karzahayel to the east and Siloamfara to the northwest. Ain Dara is settled by around 30 families.
Nearest Neighbor 1	East High School (Erie Pennsylvania): East High School part of the Erie City School District is a public high school located in Erie Pennsylvania United States. The school colors are scarlet and gray. The school mascot is a Native American Warrior. People associated with East High may be referred to as East High School Warriors East High Warriors or Warriors.	Ain Dara Syria: Ain Dara (Arabic: Ł \u200e also spelled Ayn Darah) is a small village in northern Syria administratively part of the Afrin District of the Aleppo Governorate located northwest of Aleppo. Nearby localities include Afrin to the north Karzahayel to the east and Bassouta to the south. According to the Syria Central Bureau of Statistics (CBS) Ain Dara had a population of 248 in the 2004 census.The modern-day settlement of Ain Dara is situated just east of the ancient Ain Dara temple.
Nearest Neighbor 2	Calvert School: Calvert School is a lower and middle co-educational private school with a day school operation in Baltimore Maryland and an associated homeschooling division that administers a curriculum shipped to families around the United States and the world. Developed in 1906 the home school curriculum grew by being advertised in the National Geographic magazine as a kindergarten program for those wishing to offer a better education to their children.	Carabus hemprichi:,Carabus hemprichi is a species of black-coloured ground beetle in the Carabinae subfamily that can be found in Israel Lebanon Syria and Turkey
Nearest Neighbor 3	South Elgin High School: South Elgin High School (SEHS) opened 2004 is a four-year high school located in South Elgin Illinois a northwest suburb of Chicago Illinois in the United States. It is part of Elgin Area School District U46 which also includes Elgin High School Larkin High School Bartlett High School and Streamwood High School. The class of 2008 was the first to graduate at the high school. The class of 2009 was the first four year graduating class from the high school.	Retowy:,Retowy [rtv] (German: Rettauen) is a village in the administrative district of Gmina Spopol within Bartoszyce County Warmian-Masurian Voivodeship in northern Poland close to the border with the Kaliningrad Oblast of Russia. It lies approximately 10 kilometres (6 mi) north-west of Spopol 14 km (9 mi) north-east of Bartoszyce and 68 km (42 mi) north-east of the regional capital Olsztyn.Before 1945 the area was part of Germany (East Prussia).

Table 4: Above, we show two example sentences from DPedia along with their nearest neighbors from the training dataset (DBPedia). Nearest neighbors were calculated by selecting the three examples from the training dataset (1400 examples) with the greatest TF-IDF vector cosine distance to the generated example.