Journal of Physics: Conference Series

PURPOSE-LED PUBLISHING™

PAPER • OPEN ACCESS

A Geometric Approach to Textual Augmented Data Filtering

To cite this article: Sherry J.H Feng *et al* 2024 *J. Phys.: Conf. Ser.* **2833** 012007

View the article online for updates and enhancements.

You may also like

- Cross-media Age Regression with Textual Adaptation
Jing Chen, Long Cheng, Bo Deng et al.
- A multilayer network diffusion-based model for reviewer recommendation
Yiwei Huang, , Shuqi Xu et al.
- Authorship recognition via fluctuation analysis of network topology and word intermittency
Diego R Amancio

This content was downloaded from IP address 125.239.200.174 on 30/09/2024 at 03:59

Journal of Physics: Conference Series **2833** (2024) 012007 doi:10.1088/1742-6596/2833/1/012007

A Geometric Approach to Textual Augmented Data Filtering

Sherry J.H Feng, Edmund M-K Lai and Weihua Li

School of Engineering, Computer, and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand

E-mail: jiahui.feng@autuni.ac.nz, edmund.lai@aut.ac.nz, weihua.li@aut.ac.nz

Abstract. Data augmentation is necessary if the amount of training data is insufficient for supervised learning. For natural language processing tasks, obtaining good quality augmented data is not easy. This paper introduces GATFilter, a novel method for filtering out inappropriate augmented textual data for text classification (TC). Utilizing geometric concepts, more specifically the principle component and convex hull analyses, this method adeptly preserves the semantic integrity of words within augmented texts. GATFilter is versatile and applicable across various types of textual augmentation methods. Experiments using several datasets and augmentation strategies showed that classifiers trained with GATFilter-filtered augmented data sets showed improvements in key performance metrics, including accuracy, precision, recall, and F1 score. The method's efficacy is notably influenced by the quality of the underlying augmentation techniques, indicating its potential to complement and refine various text augmentation strategies. Furthermore, our analysis showed that GATFilter is particularly able to amplify the effectiveness of methods that generate good quality augmented data. GATFilter is openly available online on Github[1], and as a Python package[2].

1. Introduction

Supervised machine learning requires labelled datasets which are expensive and time-consuming to obtain. In practice, the amount of labelled data available is often insufficient to train a model to a satisfactory level of robustness and accuracy. One way to overcome this problem is to artifically expand the training datasets through data augmentation (DA). DA techniques make use of label-preserving transformations of existing training data to create additional data for particular labels. They were first used in image recognition, where augmented data are generated by translating and rotating the object of interest in an image [1]. DA for natural language processing (NLP) tasks is not as straightforward [2]. For TC, a number of different DA methodologies have been developed over the years. They include rule/structure-based approaches [3, 4], which manipulate text based on predefined linguistic rules, language model-based methods [5, 6], which leverage advanced neural models for context-aware text generation, and generative techniques [7], which create new data instances via model-driven synthesis. More recently, large language models (LLMs) such as Bidirectional Encoder Representations from Transformers (BERT) [8] and Generative Pre-trained Transformer (GPT) [9] offer unprecedented capabilities in language understanding and generation. These models have significantly

[1] https://github.com/SherryFeng123/gatfilter
[2] https://pypi.org/project/gatfilter/

Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

influenced the evolution of DA techniques, providing more nuanced and contextually rich augmentations.

The augmented texts generated by the above methods are not without problems. The primary concern lies in the introduction of noise and errors, which can negatively affect the quality and reliability of the augmented data [10]. The main issue stems from the difficulty in synthesizing texts that preserve the original meaning while introducing sufficient variability. There is inherent subjectivity in interpreting a text, where different annotators might categorize the same piece of text differently [11]. This necessitates the need for approaches that can effectively discern and maintain the semantic integrity of the augmented texts.

1.1. Related Works

Synthesized augmented data often need to be filtered to maintain the quality of training data for NLP tasks. This is particularly important for synthesis methods that generate non-label-preserving instances [2]. Common DA filtering mechanisms involve removing instances based on unigrams [12], and utilizing various similarity metrics to assess the relevance and coherence of augmented data [13, 14]. These techniques, while effective in some specific contexts, often lack the sophistication needed to handle the complexities of most NLP tasks [2]. Recently, generative DA approaches have been proposed [10, 15]. These methods integrate filtering directly into the augmentation process, typically employing a class-trained classifier to filter instances. This strategy, however, has its drawbacks. Most notably, the diversity of training samples could be reduced, leading to overfitting issues. Recognizing these challenges, some generative DA methods [16–18] incorporate filtering mechanisms that aim to select a diverse range of training samples. While this represents a step forward, current filtering mechanisms are augmentation method-specific, and could not be applied to alternative augmentation strategies or NLP tasks [2]. In addition to this constraint, many filtering mechanisms for DA only show minimal improvements [2].

Meanwhile with traditional anomaly detection, performance significantly depends on predefined parameters, such as the predetermined number of outliers [19]. This requirement can be problematic for TC tasks that involve filtering out words, as it assumes prior knowledge of the data's outlier characteristics, potentially limiting the model's ability to adapt to diverse and unforeseen anomalies within textual data.

1.2. Algorithm Advantages

We propose a Geometric Augmented Text Filter (GATFilter) that solves the limitations in current filtering mechanisms for DA. GATFilter is a standalone algorithm that focuses on preserving the semantic meaning of text by analyzing the words and their relationships in the training data using geometric concepts. The following are the advantages of our algorithm:

(i) **Standalone Filtering Mechanism**
To the best of our knowledge, GATFilter is the first filtering mechanism that is not confined to a specific augmentation method or generator.

(ii) **Improvement Across Various Classification Needs**
Based on our experiments, GATFilter shows improvement across different classification applications, including sentiment, topic, translation and question-answering.

(iii) **Flexibility in DA stages**
GATFilter can be used at any stage in the DA pipeline. This adaptability is particularly important in the ever-evolving field of NLP.

Our approach is based on the understanding that the geometric properties of word embeddings can offer insightful clues into the semantic relationships and contextual appropriateness of words

in augmented texts. The proposed method first performs a dimensionality reduction of the word embeddings to 2D. Words that are outside of the convex hull of these 2D points are then filtered out. Section 2 provides details of the filtering method. The effectiveness of the method is verified through experiments using different word embeddings and datasets. Description of the experimental design is given in Section 3. The results and their analyses are presented in Section 4.

2. GATFilter

Word embeddings, such as Word2Vec [20], GloVe [21], and FastText [22] represent words as dense vectors that encapsulate their semantic meaning [23]. This is how words are converted to numerical values that could be used as inputs to neural networks. Each vector can be viewed as a point in the geometric space of words. For a classification task, their relative positions within the semantic landscape of each category or label could be leveraged to determine if an augmented text is semantically similar enough to be considered as useful additional samples for training.

The GATFilter algorithm constructs a geometric boundary based on the point set of word embeddings of the training data of each label. It is then used as a decision boundary to determine if the augmented data is suitable for this label. Among a selection of geometric boundaries, the convex hull is a well-established concept that can be computed efficiently. A convex hull is defined as the smallest convex set that fully encloses a set of points [24], forming the tightest possible boundary around these points without concave indentations. By analyzing the convex hull of a point set, we can obtain valuable insights into its geometric structure. This concept has seen applied to various fields, including image classification [25].

Previous attempts to apply convex hull analysis to text data encountered challenges due to the computational complexity [26]. This is because the word embeddings are very high-dimensional vectors, with typical dimensions in the range of 250 to over 300. One way to overcome this problem is to perform dimensionality reduction. For GATFilter, Principal Component Analysis (PCA) is used to map each embedding vector onto a two-dimensional space. This reduction is not only computationally efficient but studies have shown that PCA preserves the essential semantic relationships between words [27], especially for TC [28].

With the word embeddings mapped to a 2D space, a convex hull of those points can be computed for the training data of each separate label in the dataset. The convex hull hence acts as a boundary encapsulating the baseline texts associated with each label. Given some augmented data for a label, we can examine whether these augmented data points fall within the semantic boundaries of that label. If a data point (and hence the associated word) falls inside the convex hull, it is considered relevant and is retained. Words outside the convex hull are deemed irrelevant or out of context and are filtered out. This process ensures that the augmented dataset maintains integrity and relevance to the specific labels.

Details of the algorithm are described in Algorithm 1.

The GATFilter algorithm can be integrated into existing workflows with ease, and is available as a Python package. For details on installation and usage, please refer to the GATFilter Python package[3].

3. Experimental Design

A set of computational experiments is designed to assess the effectiveness of the proposed GATFilter algorithm. In the rest of this paper, we shall use the terms – Baseline, Augmentation, and Filtered, to refer to each of the following conditions.

- Baseline: The original datasets are used without any augmentation.

[3] `https://pypi.org/project/gatfilter/`

Algorithm 1 GATFilter Algorithm

Input: *baseline_text*, *augmented_text*, *word_embedding*
Output: *text_within_convex_hull*

```
input_text ← Group(input)
vector_map ← {}
text_within_convex_hull ← {}
for each word in input_text do
    Tokenize(word)
    reduced_2D ← PCA(word)
    vector_map[word: reduced_2D]
end for
for each label do
    label_CH ← ConvexHull(baseline_text)
    Plot augmented_data on label_CH
    for each data_point in label_CH do
        if data_point is inside label_CH then
            word ← vector_map[(data_point)]
            text_within_convex_hull ← {label : word}
        end if
    end for
end for
return text_within_convex_hull
```

- Augmentation: Datasets are augmented with the methods described in Table 2.
- Filter: Datasets where one of the augmentation methods in Table 2 is applied, followed by the filtering using the GATFilter algorithm.

In each case, a Convolutional Neural Network (CNN) is trained to assess performance variations.

3.1. Datasets

Four commonly used datasets involving three different types of TC tasks are chosen for our experiments. These datasets and their characteristics, including the number of training and test samples, and the number of labels, are summarized in Table 1.

Table 1: Dataset Information

Dataset	**Train**	**Test**	**Label**	**Task**
SST-2 [29]	9613	1821	2	Sentiment Analysis
TREC [30]	5500	500	6	Question Classification
SNIPS [31]	13084	700	7	Intent Detection
Question Topic [32]	5452	500	6	Question Classification

3.2. Augmentation Methods

Although GATFilter is independent of the text augmentation method, it is still of interest to know how well it works for a diversity of such methods. Three text augmentation methods have been selected for our experiments. They are listed in Table 2. Each of them represents a distinct approach to text augmentation.

For each training sample, the following augmentation strategies are employed:

Table 2: Text Augmentation Methods

Method	**Characteristics**
EDA [4]	Simple rule-based transformations
Backtranslation (FR) [17]	Neural network augmented strategies and round trip translations
Contextual TinyBERT [5, 6]	Transformed-based Large language models (LLMs)

- **EDA**: We change 0.05% of words in each sentence using simple rule-based transformation.
- **Backtranslation (BT):**Each sentence is translated to French and then back to English to introduce linguistic variations.
- **Contextual Augmentation using TinyBERT:** For each potential replacement word, the top 15 alternatives are generated using a BERT-based Masked Language Model. A replacement occurs with a 40% probability. However due to the nature of LLMs, the actual number of unique augmentations may vary.

So the training data sizes are doubled for EDA and BT. But for contextual augmentation, the resulting size of the training dataset varies.

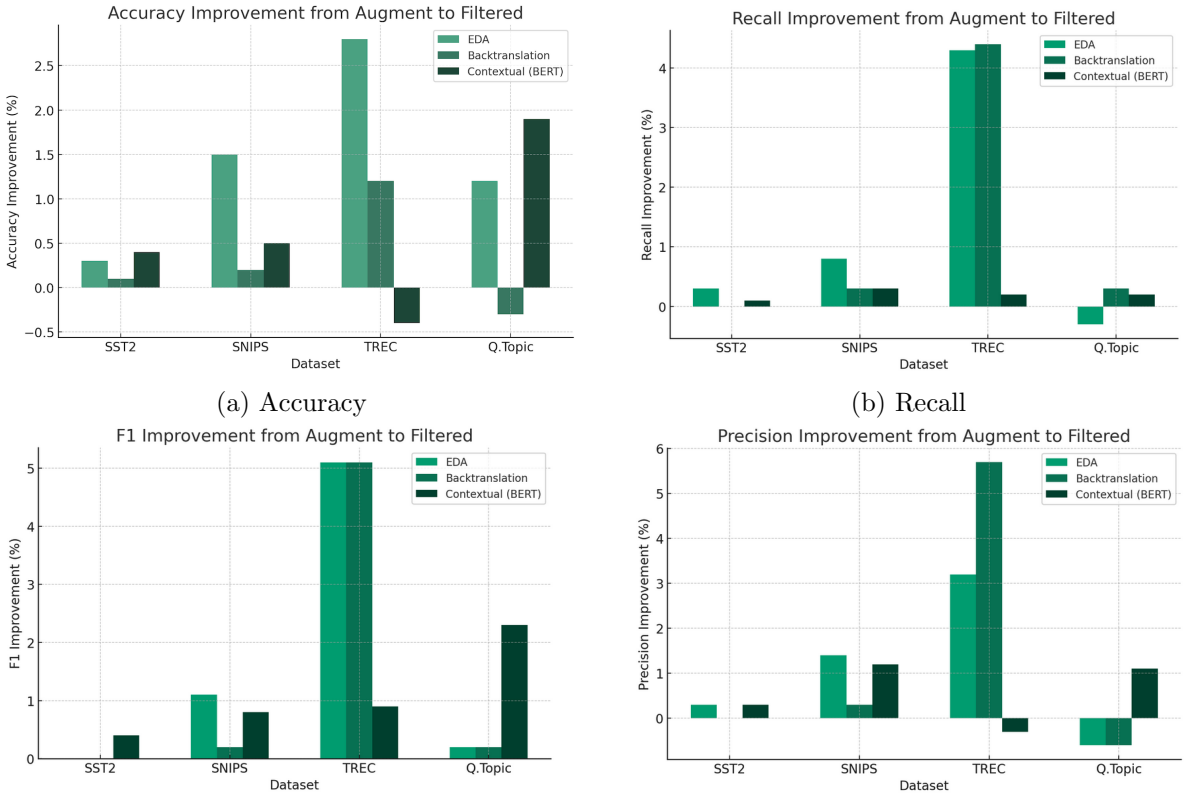
3.3. Text Classification Model

We utilize a Convolutional Neural Network (CNN), a prevalent model in supervised learning for TC [2]. Key aspects of the architecture are:

- **Embedding Dimension:** 300-dimensional embedding that is compatible with FastText embeddings.
- **Filters:** A 1D Convolutional layer with 128 filters for feature extraction.
- **Activation Function:** Softmax for the final (output) layer; Rectified Linear Unit (ReLU) otherwise.
- **Pooling:** A Global MaxPooling layer to reduce dimensionality and highlight significant features.
- **Architecture:** Comprises an embedding layer, a convolutional layer, a global max pooling layer, and two fully connected layers.

Table 3: Baseline, Augment and Filtering Results

Aug.	Corpus	test_acc. (%)			test_precision (%)			test_recall (%)			test_f1 (%)		
Method		Baseline	Aug.	GATFilter	Baseline	Aug.	GATFilter	Baseline	Aug.	GATFilter	Baseline	Aug.	GATFilter
EDA	SST2	77.6	80.5	**80.8**	77.9	80.6	**80.9**	77.7	80.5	**80.8**	77.6	80.4	**80.8**
	SNIPS	87.1	88.7	**90.4**	87.5	90.3	90.0	87.1	88.7	**90.1**	87.1	88.9	**90.0**
	TREC	84.4	87.6	**92.0**	84.1	84.8	**91.2**	77.9	82.4	**86.0**	79.0	83.2	**88.1**
	Q.Topic	94.6	95.9	**98.0**	95.6	96.3	**98.0**	94.5	95.4	**98.0**	94.8	95.8	**98.0**
backtranslation	SST2	80.1	80.5	**80.6**	80.3	80.7	80.6	80.1	80.5	**80.6**	80.0	80.4	**80.6**
	SNIPS	90.6	89.9	**91.1**	91.4	91.1	91.4	90.6	89.9	**91.1**	90.7	90.0	**91.1**
	TREC	86.0	89.2	**90.4**	83.6	86.3	**92.0**	79.4	82.1	**86.7**	80.8	83.6	**88.7**
	Q.Topic	94.8	96.6	96.5	94.7	96.8	96.2	94.3	96.3	**96.5**	94.5	96.6	96.3
contextual (BERT)	SST2	76.3	76.2	**76.6**	76.5	76.2	76.5	76.3	76.2	**76.5**	76.3	76.2	**76.6**
	SNIPS	89.4	89.7	**91.9**	90.2	91.0	**92.2**	89.4	89.7	**91.9**	89.5	89.9	**91.9**
	TREC	82.8	81.4	81.0	81.7	82.3	82.0	80.7	76.0	**76.2**	80.6	78.1	77.0
	Q.Topic	73.3	75.5	**77.9**	75.8	76.4	**77.5**	71.9	74.0	**75.7**	71.6	73.7	**76.0**



(a) Accuracy (b) Recall

(c) F1 (d) Precision

Figure 1: Filter Method Testing Performance

4. Results and Analysis

The experimental results are shown in Table 3. Improved classification performance across all four metrics: accuracy, precision, recall, and F1 score is achieved using GATFilter. for all datasets and most augmentation strategies.

4.1. Effects on Augmentation Strategies

The improvements provided by GATFilter to each of the three augmentation methods are visually depicted in Figure 1. Among the augmentation strategies, EDA emerged as the method that benefitted the most, with enhancements in test performance up to 8% across all datasets and metrics. On the other hand, Backtranslation showed mixed results. For instance, in the TREC dataset, it showed notable improvements, especially in precision and F1 scores. However, its impact varied across other datasets, indicating its effectiveness is highly dependent on the characteristics of the dataset.

Contextual Augmentation using BERT displayed variable performance as well. While it significantly enhanced the F1 score for the Question Topic dataset, it demonstrated less or even negative impacts with TREC. This variability suggests that the effectiveness of filtering contextually augmented data is dataset dependent.

4.2. Performance Trends

On average, GATFilter does provide improved classification accuracies across all three augmentation methods and the four datasets as shown in Figure 2 and 3. The effectiveness of GATFilter tends to be amplified when the augmentation method itself contributes positively

to the training dataset. On the other hand, when the augmentation method is less effective, as observed with the contextual method in the TREC dataset for example, the ability of GATFilter to enhance performance is relatively limited. Further investigation is warranted to validate this hypothesis.

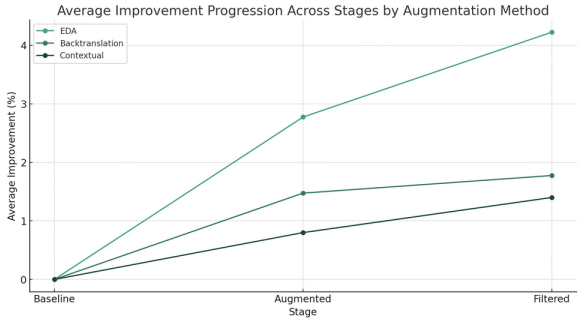


Figure 2: Method

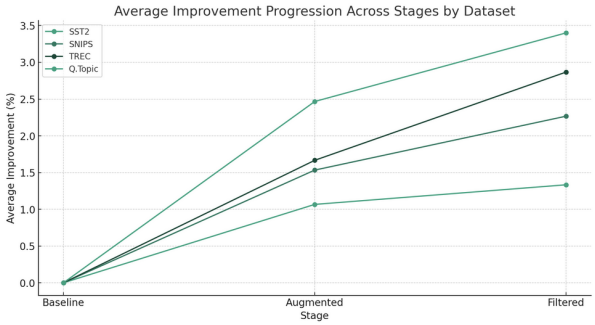


Figure 3: Dataset

4.3. Geometric Analysis

Figure 4 shows the convex hull of label 0 of the baseline TREC dataset. The data points shown in this figure include those of the augmented data. We can see that many augmented data points are located outside the convex hull. Those data points would be removed by GATFilter. Hence, it is not surprising that the classification accuracy for this label showed the most improvements after filtering.

We compare a TREC CH with contextual augmented data points (which benefited the most) against SNIPS CH with back-translated augmented data points (which had only minimal improvement).

Figure 5 shows the convex hull of the label 0 of the baseline SNIPS dataset together with the augmented data points. In contrast to Figure 4, there are much less data points outside of the convex hull. This means that the GATFilter would have removed less augmented data in this case. As a result, the improvement in classification accuracy is minimal after applying the GATFilter.

These observations provided some justification to the use of convex hull for filtering.

4.4. Sentences Produced

Table 4: Samples of original, augmented, and filtered text from TREC and SNIPS datasets.

Notes	**Original**	**Augmented**	**Filtered**
TREC, CA	What is the correct way to abbreviate cc. at the bottom of a business letter?	That is the correct way to aback ci. at the bottom of a financial letter?	that is the correct way to . at the bottom of a financial letter?
SNIPS, BT	What's the weather at my current location?	What is the weather forecast for my current location?	What is the weather forecast for my current location?

It is interesting to see the sentences produced by the augmentation processes based on the original and the resulting sentences after GATFilter. Table 4 shows a sample from label 0 of the TREC dataset. The contextually augmented version of the sentence has transformed the words in such a way that the original meaning was lost. The GATFilter then removed out words

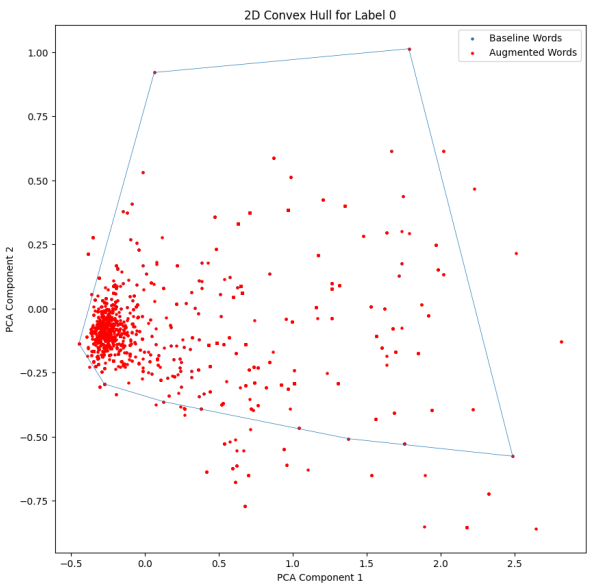


Figure 4: TREC Label 0 Convex Hull

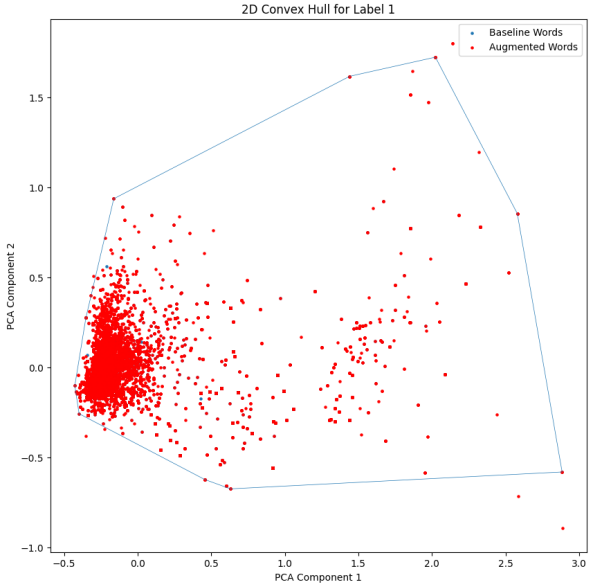


Figure 5: SNIPS Label 0 Convex Hull

"aback" and "ci" which are outside of the convex hull, resulting in an incomplete sentence. The word "financial" is retained since it is semantically close enough to "business" for its data point to be inside the convex hull.

Another sample taken from label 0 of the SNIPS dataset is shown in Table 4. In this case, the augmented sentence is generated by the backtranslation method. Also, the augmented sentence retains the original meaning. When passed to the GATFilter, it presents an augmentation that does not need any filtering. It is also worth noting that there was a significant performance improvement from the baseline to the backtranslation augmented dataset in this case. Translating between English and French benefitted from positive cross-linguistic influence, as suggested by the language transfer theory [33].

Overall, our findings indicate a correlation between the number of words in the augmented sentences that are outside the convex hull and the improvement in accuracy after GATFiltering. This shows that GATFilter effectively removes words that do not contribute meaningfully to the sentence, enhancing the quality of augmented datasets.

5. Conclusions

In this paper, we proposed a method, called GATFilter, for augmentation data filtering that is tailored for textual data. Central to our method is the innovative use of geometric concepts, more specifically convex hull, that is constructed from the dimensionally reduced word embeddings of the original training datasets. An advantage of GATFilter is that it is independent of the augmentation method. Hence it is universally applicable across various types of textual augmentation and at any point within the data augmentation pipeline.

Experimental results demonstrate that GATFilter enhances the performance of augmented datasets. This improvement is evident across several metrics including accuracy, precision, recall, and F1 score. It is also evident that there is a correlation between the effectiveness of GATFilter and the inherent quality of the underlying augmentation method. This relationship underscores GATFilter's potential to serve as a complementary tool to amplify the benefits of good augmentation methods, thereby improving the performance of TC models.

We believe that adopting a geometric view in analyzing textual data holds promise. Looking

ahead, we envisage numerous avenues for future research. For instance, extending the application of GATFilter to a broader range of NLP tasks, delving into its adaptability and effectiveness in diverse linguistic scenarios. It can also be used as a filtering mechanism for non-augmented TC pipelines. In addition, the effects of using different word embeddings could be explored.

Acknowledgement
This work is partially supported by Project 111 (No. D23006).

References

[1] Krizhevsky A, Sutskever I and Hinton G E 2017 *Communications of the ACM* **60** 84–90
[2] Bayer M, Kaufhold M A and Reuter C 2022 *ACM Computing Surveys* **55** 146:1–146:39
[3] Yu A W, Dohan D, Luong M T, Zhao R, Chen K, Norouzi M and Le Q V 2018 QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension *6th International Conference on Learning Representations*
[4] Wei J and Zou K 2019 EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* ed Inui K, Jiang J, Ng V and Wan X (Hong Kong, China: Association for Computational Linguistics) pp 6382–6388
[5] Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, Wang F and Liu Q 2020 TinyBERT: Distilling BERT for Natural Language Understanding *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online: Association for Computational Linguistics) pp 4163–4174
[6] Kobayashi S 2018 Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* ed Walker M, Ji H and Stent A (New Orleans, Louisiana: Association for Computational Linguistics) pp 452–457
[7] Hou Y, Liu Y, Che W and Liu T 2018 Sequence-to-Sequence Data Augmentation for Dialogue Language Understanding *Proceedings of the 27th International Conference on Computational Linguistics* ed Bender E M, Derczynski L and Isabelle P (Santa Fe, New Mexico, USA: Association for Computational Linguistics) pp 1234–1245
[8] Devlin J, Chang M W, Lee K and Toutanova K 2019 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding *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)* ed Burstein J, Doran C and Solorio T (Minneapolis, Minnesota: Association for Computational Linguistics) pp 4171–4186
[9] Radford A, Narasimhan K, Salimans T and Sutskever I 2018 Improving language understanding by generative pre-training Tech. rep. Technical Report, Open AI publisher: OpenAI
[10] Anaby-Tavor A, Carmeli B, Goldbraich E, Kantor A, Kour G, Shlomov S, Tepper N and Zwerdling N 2020 Do Not Have Enough Data? Deep Learning to the Rescue! *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (AAAI Press) pp 7383–7390
[11] Hameed A, Sleeman D and Preece A 2002 Detecting Mismatches among Experts' Ontologies acquired through Knowledge Elicitation *Research and Development in Intelligent Systems XVIII* ed Bramer M, Coenen F and Preece A (London: Springer) pp 9–22
[12] Paradis F and Nie J Y 2007 *Information Processing & Management* **43** 344–352
[13] Wan Z, Wan X and Wang W 2020 Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation *Proceedings of the 28th International Conference on Computational Linguistics* ed Scott D, Bel N and Zong C (Barcelona, Spain (Online): International Committee on Computational Linguistics) pp 2202–2212
[14] Parikh A, Täckström O, Das D and Uszkoreit J 2016 A Decomposable Attention Model for Natural Language Inference *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* ed Su J, Duh K and Carreras X (Austin, Texas: Association for Computational Linguistics) pp 2249–2255
[15] Queiroz Abonizio H and Barbon Junior S 2020 Pre-trained Data Augmentation for Text Classification *Intelligent Systems* Lecture Notes in Computer Science ed Cerri R and Prati R C (Cham: Springer International Publishing) pp 551–565 ISBN 978-3-030-61377-8
[16] Labani M, Moradi P, Ahmadizar F and Jalili M 2018 *Engineering Applications of Artificial Intelligence* 25–37

[17] Yang Y, Malaviya C, Fernandez J, Swayamdipta S, Le Bras R, Wang J P, Bhagavatula C, Choi Y and Downey D 2020 Generative Data Augmentation for Commonsense Reasoning *Findings of the Association for Computational Linguistics: EMNLP 2020* ed Cohn T, He Y and Liu Y (Online: Association for Computational Linguistics) pp 1008–1025
[18] Xie Z, Genthial G, Xie S, Ng A and Jurafsky D 2018 Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* ed Walker M, Ji H and Stent A (New Orleans, Louisiana: Association for Computational Linguistics) pp 619–628
[19] Taha A and Hadi A S 2019 *ACM Comput. Surv.* **52**
[20] Mikolov T, Chen K, Corrado G and Dean J 2013 Efficient Estimation of Word Representations in Vector Space *Workshop Proceedings of the 2013 International Conference on Learning Representations*
[21] Pennington J, Socher R and Manning C 2014 Glove: Global Vectors for Word Representation *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar: Association for Computational Linguistics) pp 1532–1543
[22] Joulin A, Grave E, Bojanowski P and Mikolov T 2017 Bag of Tricks for Efficient Text Classification *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* ed Lapata M, Blunsom P and Koller A (Valencia, Spain: Association for Computational Linguistics) pp 427–431
[23] Raunak V, Gupta V and Metze F 2019 Effective Dimensionality Reduction for Word Embeddings *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)* (Florence, Italy: Association for Computational Linguistics) pp 235–243
[24] de Berg M, Cheong O, van Kreveld M and Overmars M 2008 Convex Hulls *Computational Geometry: Algorithms and Applications* (Berlin, Heidelberg: Springer) pp 243–258 ISBN 978-3-540-77974-2
[25] Yousefzadeh R 2020 Deep Learning Generalization and the Convex Hull of Training Sets *NeurIPS 2020 Workshop: Deep Learning through Information Geometry*
[26] Casadio M, Komendantskaya E, Rieser V, Daggitt M L, Kienitz D, Arnaboldi L and Kokke W 2022 *arXiv preprint arXiv:2206.14575*
[27] Ning-min S and Jing L 2015 *International Journal of Database Theory and Application* **8** 57–74
[28] Taloba A I, Eisa D A and Ismail S S I 2018 *International Journal of Computer Applications* **180** 1–6 arXiv:1807.03283 [cs]
[29] Socher R, Perelygin A, Wu J, Chuang J, Manning C D, Ng A and Potts C 2013 Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* ed Yarowsky D, Baldwin T, Korhonen A, Livescu K and Bethard S (Seattle, Washington, USA: Association for Computational Linguistics) pp 1631–1642
[30] Li X and Roth D 2002 Learning question classifiers *Proceedings of the 19th international conference on Computational linguistics* vol 1 (Taipei, Taiwan: Association for Computational Linguistics) pp 1–7
[31] Coucke A, Saade A, Ball A, Bluche T, Caulier A, Leroy D, Doumouro C, Gisselbrecht T, Caltagirone F, Lavril T, Primet M and Dureau J 2018 Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces arXiv:1805.10190 [cs]
[32] Hovy E, Gerber L, Hermjakob U, Lin C Y and Ravichandran D 2001 Toward Semantics-Based Answer Pinpointing *Proceedings of the First International Conference on Human Language Technology Research*
[33] Odlin T 1989 *Language Transfer* (Cambridge University Press) ISBN 978-0-521-37168-1