Beyond Bag-of-Concepts: Vectors of Locally Aggregated Concepts

Maarten Grootendorst[1] (✉) and Joaquin Vanschoren[2]

[1] Jheronimus Academy of Data Science. 5211 DA 's-Hertogenbosch, The Netherlands
maartengrootendorst@gmail.com
[2] Eindhoven University of Technology. 5612 AZ Eindhoven, The Netherlands
j.vanschoren@tue.nl

Abstract. Bag-of-Concepts, a model that counts the frequency of clustered word embeddings (i.e., concepts) in a document, has demonstrated the feasibility of leveraging clustered word embeddings to create features for document representation. However, information is lost as the word embeddings themselves are not used in the resulting feature vector. This paper presents a novel text representation method, Vectors of Locally Aggregated Concepts (VLAC). Like Bag-of-Concepts, it clusters word embeddings for its feature generation. However, instead of counting the frequency of clustered word embeddings, VLAC takes each cluster's sum of residuals with respect to its centroid and concatenates those to create a feature vector. The resulting feature vectors contain more discriminative information than Bag-of-Concepts due to the additional inclusion of these first order statistics. The proposed method is tested on four different data sets for single-label classification and compared with several baselines, including TF-IDF and Bag-of-Concepts. Results indicate that when combining features of VLAC with TF-IDF significant improvements in performance were found regardless of which word embeddings were used.

Keywords: Bag of Concepts · Vector of Locally Aggregated Descriptors · Vectors of Locally Aggregated Concepts.

1 Introduction

Methods for creating structure out of unstructured data have many applications, ranging from classifying images to creating spam-filters. As a typical form of unstructured data, textual documents benefit greatly from these methods as words can have multiple meanings, grammatical errors may occur and the way text is constructed differs from language to language. Arguably, one of the most popular methods for representing documents is Bag-of-Words, which scores the frequency of words in a document based on its corpus [28]. This results in a structured document representation despite the inherently messy nature of textual data. However, as corpora grow bigger and exceed tens of thousands of words, Bag-of-Words representations lose their interpretability.

Bag-of-Concepts was proposed as a solution to this problem [14]. Based on the corpus of a collection of documents, Bag-of-Concepts generates word clusters (i.e., concepts) from vector representations of words (i.e., word embeddings) and, similar to Bag-of-Words, counts the number of words in a document associated with each concept, hence the name Bag-of-Concepts.

Interestingly, Bag-of-Concepts shares many similarities with Bag-of-Visual-Words, a feature generation method used for image classification [27]. Much like Bag-of-Concepts, Bag-of-Visual-Words represents images by the occurrence count of its clustered features (i.e., descriptors). The main difference between these methods is that Bag-of-Concepts leverages word clusters whereas Bag-of-Visual-Words leverages image feature clusters.

Although Bag-of-Visual-Words shows promising results in image classification, it typically generates sparse features with high dimensionality [21]. Vector of Locally Aggregated Descriptors (VLAD) extends upon Bag-of-Visual-Words by including first order statistics into its feature vectors [7]. Compared to Bag-of-Visual-Words, VLAD allows for compact visual representations with high discriminative ability due to the inclusion of descriptors' locations in each cluster.

As the main difference between Bag-of-Visual-Words and Bag-of-Concepts is the type of clustered features that are used, it follows that VLAD could be generalized to the generation of textual features by leveraging word embeddings instead of image descriptors. This would result in a document representation with more discriminative ability than Bag-of-Concepts as it contains additional first order statistics in its feature vectors. The resulting method was named Vectors of Locally Aggregated Concepts (VLAC) after both VLAD and Bag-of-Concepts.

To the best of my knowledge, no research seems to exist concerning the application of VLAD for representing textual documents. Although creating structure out of unstructured has many applications, document classification, due to its popularity, was chosen as a proxy for measuring the quality of document representation. This study shows that VLAD offers a novel way to create features for document representation, resulting in better predictions for document classification.

2 Related Work

2.1 Bag-of-Words

Bag-of-Words counts the occurrences of words within a document in which each word count is considered a feature. A disadvantage of this method is that highly frequent words may dominate the feature space while rarer and more specific words may contain more information. In order to lessen the impact of those words and evaluate the importance of words in a document, one can use a weighting scheme named TF-IDF. It combines two statistics, namely term frequency (TF) multiplied by its inverse document frequency (IDF). Term frequency is the count of word t in a document d. Then, for each term t, inverse document frequency calculates how common t is across all documents D by taking the logarithm of the number of documents in a corpus N divided by the number of documents that contain t.

$$IDF(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{1}$$

Together, it takes the frequency of words in a document and calculates the inverse proportion of those words to the corpus [11, 24].

2.2 Word Embeddings

Although TF-IDF succeeds in representing the occurrence and importance of words in a document, the context of these words is lost. Instead, in order to retain semantic similarity among words, one can map words to vectors of real numbers, named word embeddings [15].

Word2Vec is a popular tool for mapping words in a document to a vector representation. It combines multiple two-layer neural networks to construct embeddings, namely the Continuous Bag-of-Words (CBOW) and Skip-gram architectures [17]. In the CBOW architecture, the model predicts a target word given a set of surrounding context words. In contrast, the Skip-gram architecture tries to predict a set of context words given a target word. The hidden layer then represents the word vectors as the relationships between words and context are learned. See Figure 1 for an overview of the architecture of Word2Vec.

The disadvantage of Word2Vec is that word embeddings are created locally within documents while disregarding the global representation of words across all documents. Models such as GloVe (Global Vectors for Word Representation), construct large co-occurrence counts (word × context) in order to learn the global representation of a word [20].

Typically, 300-dimensional word vectors are created as they have been shown to balance representational ability and the density of the resulting vectors [17, 20].

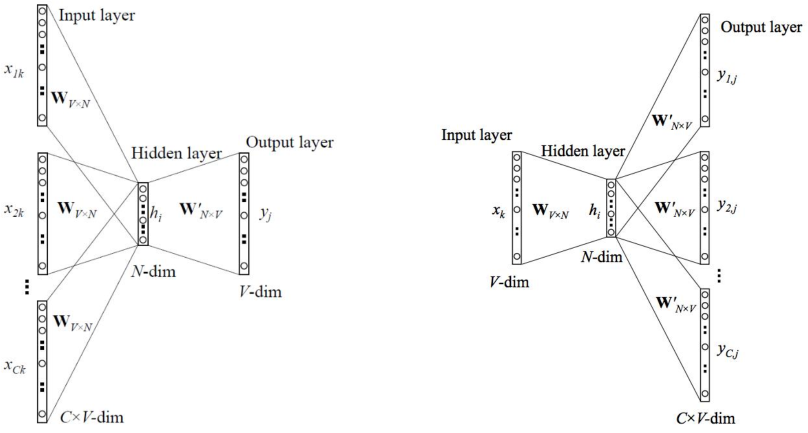

Fig. 1. CBOW architecture (left) versus Skip-gram architecture (right).

2.3 Bag-of-Concepts

When we want to represent documents instead of individual words, word embeddings can be averaged across all words in a document [3]. However, the resulting document vectors are difficult to interpret intuitively as they merely represent a point in a 300-dimensional space. In order to deal with this problem, Kim et al. (2017) [14] developed a model named Bag-of-Concepts. Based on a collection of documents, Bag-of-Concepts generates word clusters by applying spherical k-means to word embeddings. The resulting clusters typically contain words with similar meaning and are therefore referred to as concepts. Then, similar to Bag-of-Words, a document is represented as a bag of its concepts by counting the number of words in a document associated with each concept [14].

In order to lessen the impact of concepts that appear in most documents, a TF-IDF-like weighting scheme is applied in which all terms t are replaced by concept c, which is appropriately named CF-IDF. This allows the model to create document vectors that are interpretable, as each feature of a document represents the importance of a concept.

Bag-of-Concepts was found to be largely dependent on the number of concepts that were generated [14]. The authors showed that the classification accuracy of Bag-of-Concepts consistently increases with the number of concepts, but that this increase stabilizes around 200 concepts at which near-maximum performance is reached.

This method has shown to provide better document representation than Bag-of-Words and TF-IDF in a classification task to find the two most similar documents among triplets of documents [14]. However, in a classification task to predict the correct label for each document Bag-of-Concepts failed to outperform TF-IDF on two out of three data sets.

2.4 Vector of Locally Aggregated Descriptors (VLAD)

Before deep learning achieved state-of-the-art results in image classification, an approach called Bag-of-Visual-Words was often used for image classification [18]. This method is similar to Bag-of-Concepts as both cluster a collection of features of which the occurrence of these clusters is counted in each sample, thereby creating a vector for each sample containing the prevalence of clustered features. Specifically, Bag-of-Visual-Words clusters image features which are typically generated using feature extractor algorithms like SIFT or KAZE [18]. Then, it counts the occurrence of the clusters resulting in a vector of occurrence counts of local image features.

To further increase the representative ability of Bag-of-Visual-Words, first order statistics were additionally included in the resulting vectors thereby providing more information about the images. This method was named Vector of Locally Aggregated Descriptors (VLAD) and was shown to have superior performance compared to Bag-of-Visual-Words [9, 2].

As illustrated in Figure 2, VLAD extends Bag-of-Visual-Words by taking the residual of each image feature with respect to its assigned cluster center. Using

k-means each image feature x_i is assigned to a cluster with cluster center c_j, both having the same dimensionality D. N_j is equal to the number of image features in j and j ranges from 1 to k. Then, the sum of residuals of each image feature in a cluster is accumulated, resulting in k vectors for each image:

$$v_j = \sum_{i=1}^{N_j} x_i - c_j \tag{2}$$

k vectors are created containing the sum of residuals of each cluster and are then concatenated to create a single vector for each image:

$$v = \begin{bmatrix} \vdots \\ v_j \\ \vdots \end{bmatrix} \tag{3}$$

The resulting image vector is of size $k \times D$. Next, the concatenated vectors are typically first power normalized and then l2 normalized to reduce bursty visual elements [10]:

$$v = sign(v)\sqrt{|v|} \tag{4}$$

$$v = \frac{v}{\|v\|} \tag{5}$$

Several extensions to this model have been proposed to further improve its representative ability and classification performance. For example, intra-normalization has been suggested as a way to further reduce bursty image features. Instead of applying l2 normalization to the concatenated vector of the sum of residuals, it is suggested to l2 normalize the sum of residuals within each VLAD block, followed by l2 normalization of the entire vector. The effect of bursty features would then be localized to each cluster [2]. Other improvements have been suggested such as directly l2 normalizing each feature's residuals [7], adding aggregations of tensor products of the descriptors [23], and using VLAD as a layer in a convolutional neural network [1].

3 Vectors of Locally Aggregated Concepts (VLAC)

Interestingly, VLAD and Bag-of-Concepts both use clustered feature vectors as their basis for the generation of summarized features in the task of classification. This similarity suggests that VLAD could be extended to be used in the domain of natural language processing as words could be clustered instead of image features. Thus, instead of clustering descriptors, one can cluster word embeddings into concepts for the generation of features. The result is a feature generation model for textual documents inspired by VLAD and Bag-of-Concepts, namely Vectors of Locally Aggregated Concepts (VLAC).

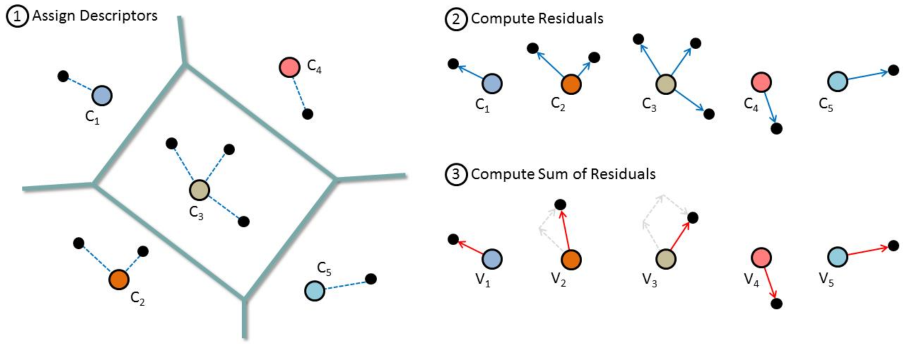

Fig. 2. Procedure of VLAD

As illustrated in Figure 3, VLAC clusters word embeddings to create k concepts. Due to the typically high dimensionality of word embeddings (i.e., 300) spherical k-means is used to perform the clustering as applying euclidean distance will result in little difference in the distances between samples. Similar to the original VLAD approach, let w_i be a word embedding of size D assigned to cluster center c_k. Then, for each word in a document, VLAC computes the element-wise sum of residuals of each word embedding to its assigned cluster center.

This results in k feature vectors, one for each concept, and all of size D. All feature vectors are then concatenated, power normalized, and finally, l2 normalization is applied as with the original VLAD approach. If 10 concepts were to be created out of word embeddings of size 300 then the resulting document vector would contain 10 × 300 values.

The resulting feature vectors contain more discriminative information than Bag-of-Concepts since the sum of residuals gives information with regard to the relative location of the word embeddings in the clusters. Therefore, it is expected that VLAC will outperform Bag-of-Concepts (with CF-IDF).

4 Experiments

In order to test the quality of the generated features by VLAC, two single-label classification experiments were performed using several baselines. VLAC is dependent on the quality of word embeddings and the number of concepts generated. Therefore, in the first experiment, several implementations of VLAC were tested against each other at different numbers of concepts. This experiment served as a way to explore how VLAC is affected by the number of concepts generated and the word embeddings that were used.

Then, to validate VLAC across different discriminative thresholds a second experiment was executed in which VLAC was compared against several baselines using Receiver Operating Characteristic (ROC) curves and compared on their

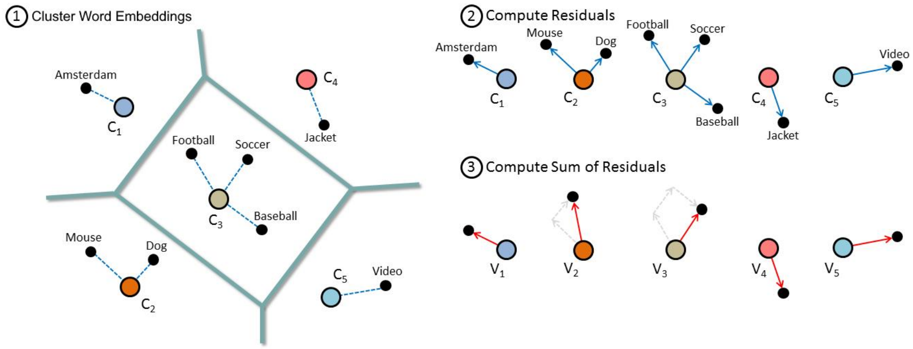

Fig. 3. Procedure of VLAC

Area Under the Curve (AUC) scores. ROC curves were not used in experiment 1 since they cannot show the effect of generated concepts on performance.

Finally, in all experiments, features generated by VLAC were added to TF-IDF features. Since TF-IDF cannot generate more features, adding VLAC features to TF-IDF will help in understanding if VLAC adds additional information not already contained within features of TF-IDF. Although more interaction is possible by creating a larger feature matrix, any improvement in performance could only be attributed to this higher dimensionality containing information not previously seen in TF-IDF.

4.1 Experimental Setup

Data Four data sets were chosen on which the effectiveness of the proposed method was tested. Three of these (20 Newsgroups, Reuters R8, and WebKB) were included because they are typically used in document classification research and therefore allow for comparisons to be made with prior work (e.g. [26, 6, 16]). As these three data sets are all written in English, an additional data set containing Portuguese documents was included as a way to further generalize the evaluation. Stemming and stop word removal were applied to all data sets. All data sets were retrieved from [5]. See Tables 1 and 2 for more information.

	Reuters R8	**20 Newsgroups**	**WebKB**	**Cade12**
Number of documents	7674	18821	4199	40983
Number of classes	8	20	4	12
Average number of words per document	64.5	141.1	133.4	117.4
Vocabulary size	17387	70213	7770	193997
Total number of words	495226	2654770	560015	4813116

Table 1. An overview of the data sets used in this study.

Reuters R8		**20 Newsgroups**		**WebKB**		**Cade12**	
Classes	Samples	Classes	Samples	Classes	Samples	Classes	Samples
earn	3923	rec.sport.hockey	999	student	1641	01–servicos	8473
acq	2292	soc.religion.christian	996	faculty	1124	02–sociedade	7363
crude	374	rec.motorcycles	996	course	930	03–lazer	5590
trade	326	rec.sport.baseball	994	project	504	04–informatica	4519
money-fx	293	sci.crypt	991			05–saude	3171
interest	271	sci.med	990			06–educacao	2856
ship	144	rec.autos	989			07–internet	2381
grain	51	sci.space	987			08–cultura	2137
		comp.windows.x	985			09–esportes	1907
		sci.electronics	984			10–noticias	1082
		comp.sys.ibm.pc.hardware	982			11–ciencias	879
		misc.forsale	975			12–compras-online	625
		comp.graphics	973				
		comp.os.ms-windows.misc	966				
		comp.sys.mac.hardware	963				
		talk.politics.mideast	940				
		talk.politics.guns	909				
		alt.atheism	799				
		talk.politics.misc	775				
		talk.religion.misc	628				

Table 2. Number of samples in each class per data set.

Balanced Accuracy Although the quality of classification is typically measured by the accuracy of the prediction model, it suffers from over representing the performance on larger classes [25]. Due to the imbalance of the data sets (see Table 2) a different measure for validation was used, namely balanced accuracy [4]:

$$BalancedAccuracy = \frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i+fp_i}}{n} \tag{6}$$

With n classes, where tp_i is the true positive for class i in n, and fp_i is the false positive for class i in n. For multi-class classification, balanced accuracy can be interpreted as the macro-average of recall scores per class [19, 13] which has the property of allowing the performance of all classes to be weighted equally.

Baselines Bag-of-Words, TF-IDF, Bag-of-Concepts (with CF-IDF), and averaged word embeddings (with Word2Vec embeddings) served as baselines in this study. Bag-of-Words, TF-IDF, and averaged word embeddings are typically used to test novel techniques against, whereas Bag-of-Concepts was chosen due to the methodological similarities it shares with VLAC. For the implementation of Bag-of-Concepts, initial experiments were performed to find a balance between the number of concepts and computational efficiency. At 500 concepts the performance of Bag-of-Concepts typically stabilizes. Moreover, previous research has found the classification accuracy of Bag-of-Concepts to stabilize around 200 concepts and that the classification accuracy consistently increases with the number of concepts generated [14]. Ultimately, Bag-of-Concepts was set at 500 concepts in order to maximize its performance.

4.2 Experiment 1

The performance of VLAC, based on balanced accuracy, was analyzed for each data set with the number of concepts systematically increasing from 1 to 30. The maximum number of concepts was set at 30 as computing more concepts would be computationally too demanding for this experiment.

Bag-of-Words, TF-IDF and averaged word embeddings were used as baselines. Kim et al. [14] demonstrated that Bag-of-Concepts, compared to TF-IDF, would need at least 100 concepts for it to reach a competitive performance. Therefore, Bag-of-Concepts was excluded from this experiment as it would not be fair to compare Bag-of-Concepts to VLAC at merely 30 concepts.

Four different types of word embeddings were used for VLAC on each data set. Word2Vec and GloVe embeddings were generated by training the model on the data sets themselves, henceforth referred to as self-trained embeddings. Moreover, pre-trained embeddings for Word2Vec and GloVe were additionally used as they had been trained on larger data sets and therefore might have better representative ability. Word2Vec pre-trained embeddings were trained on the Google News data set and contain vectors for 3 million English words.[1] GloVe pre-trained embeddings were trained on the Common Crawl data set and contain vectors for 1.9 million English words.[2] Pre-trained embeddings for Cade12 were trained on 17 different Portuguese corpora.[3] To make a comparison across VLAC implementations possible, all word embeddings were of size 300.

Linear Support Vector Machines (Linear SVM) have been shown to do well on single-label text classification tasks [12] and are used in this experiment as classifiers on top of the feature generation methods. Moreover, 10-fold cross-validation was applied in each prediction instance in order to decrease the chance of overfitting on the data and creating biased results.

Results Several one-sided, one-sample Wilcoxon signed rank tests were applied to observe which VLAC versions, on average across all 30 concepts, may outperform TF-IDF. The results are shown in Table 3 and indicate that VLAC typically does not outperform TF-IDF. However, looking at the best scores of each model in Table 3 and the accuracy curves in Figure 4, the results suggest that, around 30 concepts, VLAC can outperform TF-IDF on Reuters R8, WebKB and Cade12 depending on the word embeddings that were used.

In contrast, the combined features of TF-IDF and VLAC generally outperformed TF-IDF on Reuters R8, WebKB and Cade12 (see Table 3). This suggests

[1] Retrieved from `https://code.google.com/archive/p/word2vec/`

[2] Retrieved from `https://nlp.stanford.edu/projects/glove/`

[3] Retrieved from `http://nilc.icmc.usp.br/embeddings`

that VLAC features contain information not seen in TF-IDF features. Interestingly, Figure 5 shows that the number of concepts seem to have little influence on performance, thereby indicating that a few concepts would be sufficient in generating additional features when combining VLAC with TF-IDF.

From Figure 4 one can observe that VLAC's balanced accuracy scores are highest at 30 concepts and are likely to improve at a higher number of concepts. To evaluate VLAC at its highest performance (i.e., 30 concepts), several one-sided, two-samples Wilcoxon signed rank tests were used to compare VLAC against TF-IDF. For this, 20-fold cross-validation was applied to TF-IDF and all VLAC versions to generate 20 accuracy scores for each model. The same folds for each algorithm were created. Folds were then paired across algorithms to test for the possible difference in performance.

Although 10 folds are typically used in cross-validation, a choice was made for 20 folds in order to create a sufficiently sized sample size to increase the statistical power. The results of each fold were averaged across all data sets for each method. Thus, 20 averaged balanced accuracy scores were created for each model and allowed for the one-sided, two-samples Wilcoxon signed rank tests.

VLAC did not significantly outperform TF-IDF across all four data sets ($V = 6$, $p = .821$). However, combining features of VLAC with TF-IDF led to a significant improvement over TF-IDF ($V = 207$, $p < .001$), which was found for both self-trained embeddings ($V = 198$, $p < .001$) and for pre-trained embeddings ($V = 199$, $p < .001$). This confirms the idea that VLAC features contain information not seen in TF-IDF features.

Finally, the results indicate that pre-trained word embeddings used for VLAC performed significantly better than self-trained word embeddings ($V = 159$,

	Reuters R8		**20 Newsgroups**		**WebKB**		**Cade12**	
	Average	Best	Average	Best	Average	Best	Average	Best
VLAC (Self: W2V)	91.48	92.57	81.95	84.20	89.92**	90.83	41.69	44.85
VLAC (Pre: W2V)	92.01*	92.95	82.42	84.44	88.13	89.39	46.67	48.13
VLAC (Self: GloVe)	91.74	92.53	83.08	87.43	87.99	89.58	42.19	45.65
VLAC (Pre: GloVe)	92.10	93.11	85.27	87.50	88.40	89.97	45.10	46.36
Averaged W2V	-	89.81	-	75.21	-	87.23	-	36.02
Bag-of-Words	-	90.96	-	88.11	-	85.41	-	43.22
TF-IDF	-	91.97	-	90.44	-	89.37	-	47.02
TF-IDF + VLAC (Self: W2V)	93.22**	93.80	90.00	90.20	**90.41****	**90.74**	48.36**	48.65
TF-IDF + VLAC (Pre: W2V)	**93.71****	**94.39**	89.57	89.81	89.96**	90.43	**49.08****	**49.37**
TF-IDF + VLAC (Self: GloVe)	92.96**	93.33	89.60	89.88	89.71**	90.24	46.56	47.07
TF-IDF + VLAC (Pre: GloVe)	93.30**	93.83	**90.04**	**90.38**	90.01**	90.47	48.11**	48.80

Table 3. Average and best performance in experiment 1 across different implementations of VLAC, where *self* relates to embeddings trained on the data itself and *pre* relates to pre-trained embeddings. Underlined values are the highest results in each block, whereas bold values are the best results compared to all other methods for a single data set. One-sided, one-sample Wilcoxon signed rank tests were executed to compare all VLAC versions against TF-IDF based on their average scores. ** $p < 0.001$; * $p < 0.05$

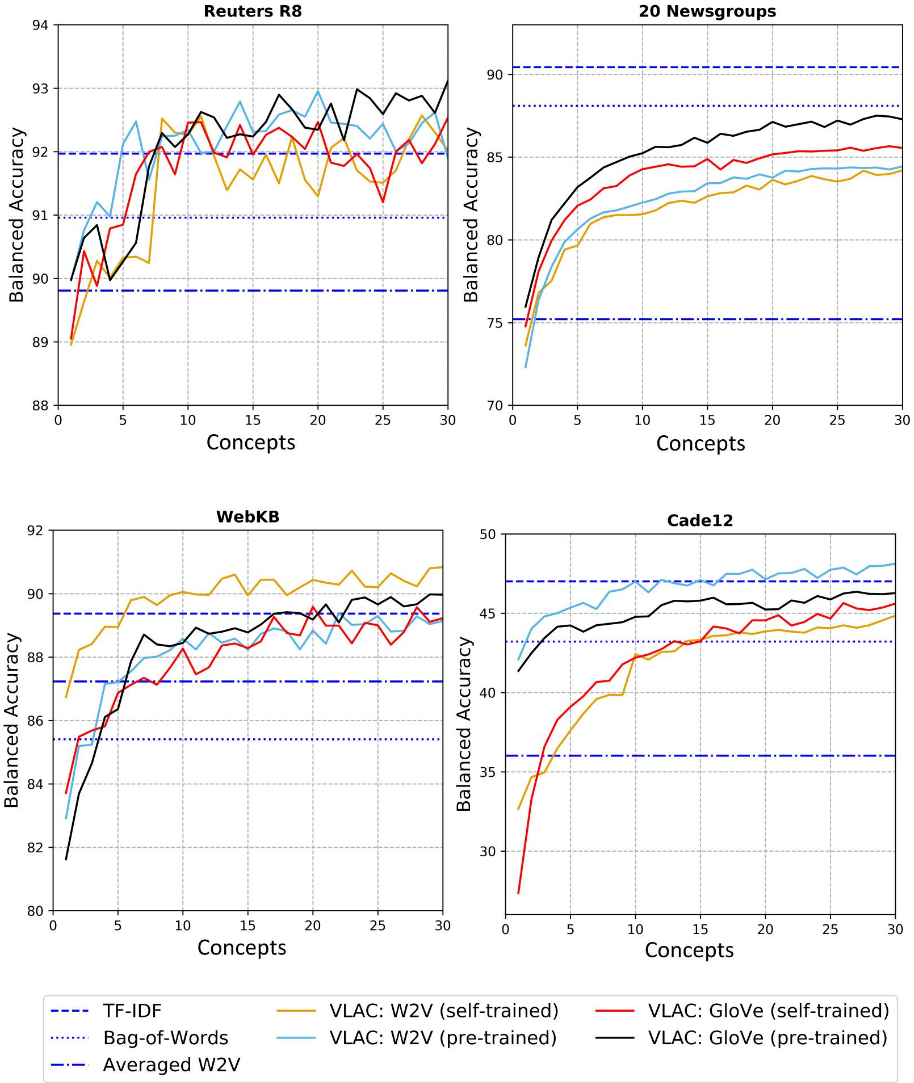

Fig. 4. Results of different word embeddings used with VLAC compared to TF-IDF, Bag-of-Words, and averaged Word2Vec embeddings.

$p = .022$). However, no differences were found between the performance of pre-trained and self-trained word embeddings when combining features of VLAC with TF-IDF ($V = 113$, $p = .392$).

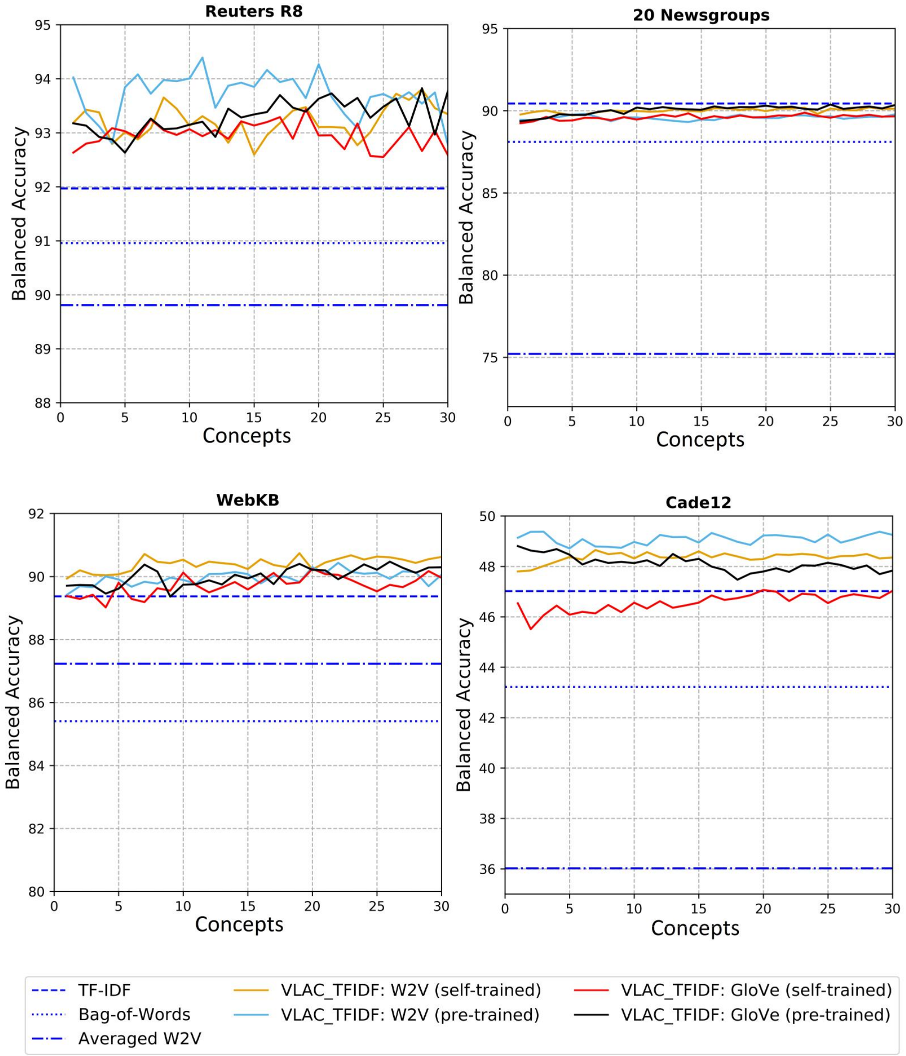

Fig. 5. Results of different word embeddings used when combining features of VLAC with those of TF-IDF compared to TF-IDF, Bag-of-Words, and averaged Word2Vec embeddings.

4.3 Experiment 2

In this experiment, the performance of all models in this study was analyzed across different discriminative thresholds to further validate VLAC. ROC curves were used to analyze the performance of VLAC across different discriminative

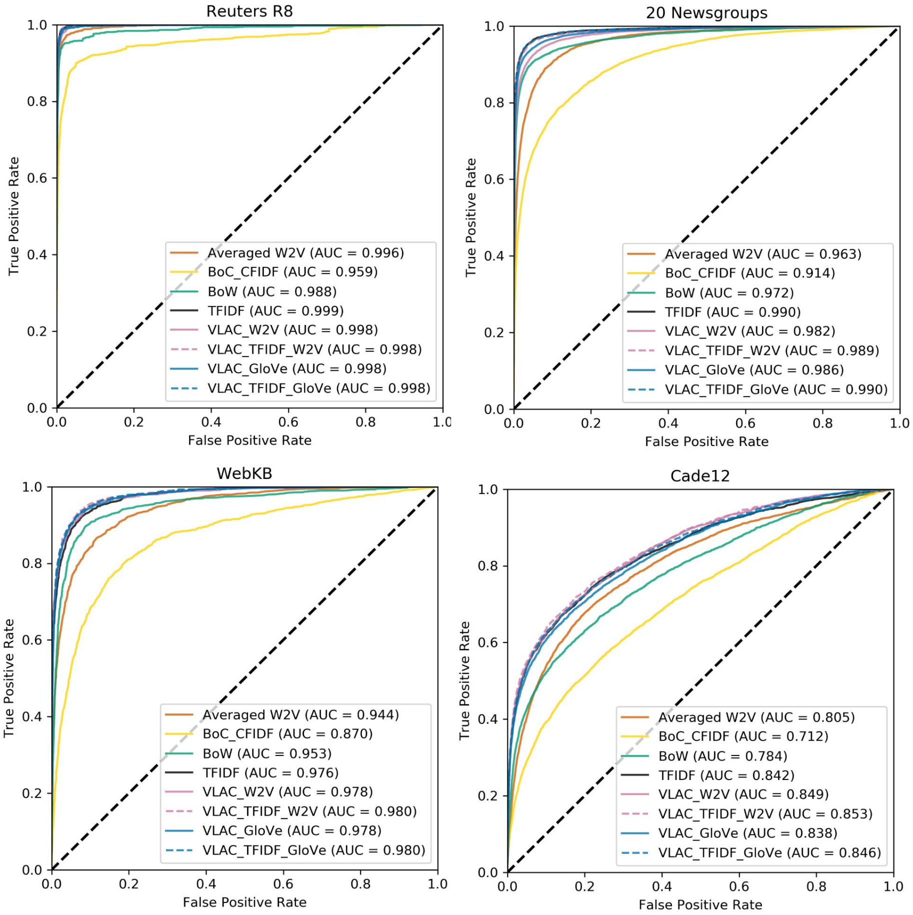

Fig. 6. ROC with macro-averaged AUC scores of eight different models across four data sets.

thresholds. Since balanced accuracy adopts a macro-averaging approach, the scores in this experiment were also macro-averaged.

For the implementation of VLAC, pre-trained word embeddings were used as they typically outperformed self-trained word embeddings. Bag-of-Words, TF-IDF, averaged Word2Vec embeddings, and Bag-of-Concepts (with CF-IDF and at 500 concepts) were included as baselines. Since the features of averaged word embeddings, Bag-of-Words, and TF-IDF are out-of-the-box maximized, all VLAC versions were set at 30 concepts to similarly maximize its number of features.

Results From Figure 6 one can observe that pre-trained VLAC versions outperformed Bag-of-Words, Bag-of-Concepts, and averaged Word2Vec embeddings on all data sets based on their respective AUC scores. Seeing as the curves in Figure 6 behave similarly across models, one can safely assume that the AUC scores are representative of the models performance. Furthermore, the results indicate that VLAC by itself can outperform TF-IDF but requires experimentation to find the optimum set of parameters (number of concepts versus the type of word embeddings). However, when combining features from VLAC with those of TF-IDF, the resulting AUC scores are similar to TF-IDF and higher on the Cade12 and WebKB datasets.

It is interesting to note that Bag-of-Concepts consistently, across all data sets, performs worst out of all methods. Although Kim et al. (2017) [14] demonstrated that Bag-of-Concepts might be able to outperform TF-IDF using a Support Vector Machine, they did not specify which kernel was used in their implementation. In this study a linear kernel was adopted. The differences between results might be due to the kernel that was used in the implementation of the Support Vector Machines. Although it was expected that VLAC would outperform Bag-of-Concepts, such a large difference between Bag-of-Concepts and all other models was not anticipated. Since Kim et al. (2017) [14] demonstrated that Bag-of-Concepts' classification accuracy increases with the number of concepts generated, one can conclude that Bag-of-Concepts is not suited for single-label classification up to 500 concepts.

For VLAC, it is not clear why there is such a large gap in performance between 20 Newsgroups and all other data sets. It could be attributed to many differences between data sets such as vocabulary size, document size, number of sentences per document, and even writing style. With so many differences between data sets it is hard to pin point the exact reason for the differences in performance. Thus, it is hard to pin point the exact reason for these differences.

However, for both 20 Newsgroups and Cade12, which are relatively large data sets compared to the others, the performances do not seem to stabilize at 30 concepts (see Figure 4). This suggests that larger documents typically require larger number of concepts in order to maximize its performance. Future research could focus on studying the effects of document size on classification accuracy.

5 Conclusion

This study presents a novel algorithm for the generation of textual features, namely Vectors of Locally Aggregated Concepts (VLAC). In two experiments the performance of VLAC was tested against several baselines including averaged Word2Vec word embeddings, Bag-of-Words, TF-IDF and Bag-of-Concepts (with CF-IDF). On average, VLAC was shown to outperform all baselines when its features were combined with those of TF-IDF regardless of which word embeddings were used.

Future work may focus on two main disadvantages of using word embeddings generated by Word2Vec and GloVe. First, these models cannot handle out-of-vocabulary words. Instead, one can use word embeddings models, such as FastText, to additionally create character-level n-gram word embeddings which can be combined to construct out-of-vocabulary words. Second, word embeddings generated by Word2Vec and GloVe are the same for each word regardless of its context. Tools such as Embeddings from Language Models (ELMo) [22] and Bidirectional Encoder Representations from Transformers (BERT) [8] create, for a single word, different word embeddings if that word can be used in different contexts. Using contextual word embeddings will allow clusters to be made with better representational ability.

This study has made a first step in demonstrating the feasibility of a novel method for single-label document classification. Although several improvements to this model have been suggested, the results demonstrate that VLAC can reach superior performance in document classification tasks compared to several strong baselines. While this paper has focused on classification, many other tasks, such as information retrieval and document clustering tasks, could potentially be solved by VLAC. [4]

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
2. Arandjelovic, R., Zisserman, A.: All about vlad. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1578–1585 (2013)
3. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International conference for learning representations (2017)
4. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: Proceedings of the 20th international conference on pattern recognition. pp. 3121–3124. IEEE (2010)
5. Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa (2007)
6. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in neural information processing systems. pp. 3079–3087 (2015)
7. Delhumeau, J., Gosselin, P.H., Jégou, H., Pérez, P.: Revisiting the vlad image representation. In: Proceedings of the 21st international conference on multimedia. pp. 653–656. ACM (2013)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Jegou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer vision and pattern recognition. pp. 3304–3311. IEEE (2010)
10. Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. Transactions on pattern analysis and machine intelligence 34(9), 1704–1716 (2012)

[4] Code and results of this study can be found at `https://github.com/MaartenGr/VLAC`

11. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: International conference on machine learning. pp. 143–151 (1996)
12. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. pp. 137–142. Springer (1998)
13. Kelleher, J.D., Mac Namee, B., D'Arcy, A.: Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT Press (2015)
14. Kim, H.K., Kim, H., Cho, S.: Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. Neurocomputing 266, 336–352 (2017)
15. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. pp. 142–150 (2011)
16. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. vol. 752, pp. 41–48. Citeseer (1998)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
18. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: European conference on computer vision. pp. 490–503. Springer (2006)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of machine learning research 12, 2825–2830 (2011)
20. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing. pp. 1532–1543 (2014)
21. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: European conference on computer vision. pp. 143–156. Springer (2010)
22. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
23. Picard, D., Gosselin, P.H.: Improving image similarity with vectors of locally aggregated tensors. In: International conference on image processing. pp. 669–672. IEEE (2011)
24. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. vol. 242, pp. 133–142 (2003)
25. Ramyachitra, D., Manikandan, P.: Imbalanced dataset classification and solutions: a review. International journal of computing and business research 5(4), 1–29 (2014)
26. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd international conference on machine learning. pp. 977–984. ACM (2006)
27. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In: Proceedings of the international workshop on Workshop on multimedia information retrieval. pp. 197–206. ACM (2007)

28. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. International journal of machine learning and cybernetics 1(1-4), 43–52 (2010)