N-GRAM EXTENSION FOR BAG-OF-AUDIO-WORDS

Stephanie Pancoast[1,2], *Murat Akbacak*[3]

[1] Speech Technology and Research Lab, SRI International, Menlo Park, CA
[2] Department of Electrical Engineering, Stanford University, Stanford, CA
[3] Microsoft, Sunnyvale, CA

ABSTRACT

Bag-of-audio-words is one of the most frequently used methods for incorporating an audio component into multimedia event detection and related tasks. A main criticism of the method, however, is that it ignores context. Each "word" is considered in isolation, ignoring its neighbors. We address this issue by representing the document by its audio word N-grams. Unlike words from natural language, audio words are generated by clustering algorithms where the number of clusters is specified by the researcher. We therefore also explore how the performance of the N-gram representation varies with codebook size. With this enhanced representation, we find the average probability of miss noticeably decreases when evaluated on TRECVID 2011 and 2012 datasets, indicating clear improvements on the multimedia event detection task.

Index Terms: Bag-of-audio-words, N-gram models, multimedia event detection

1. INTRODUCTION

Due to the popularity of online videos, multimedia modeling for the purpose of event detection and retrieval has become a recent research focus. Multimedia event detection (MED) and the related multimedia event retrieval tasks require a system that can search user-submitted quality videos for specific events. Video imagery features play a significant role in determining the content; however, the audio component for a video can also be critical. Consider the case of detecting a home run in baseball game videos. Analysis of the frame-level imagery may determine that the setting is a baseball game, but without the capability to capture cheering in the audio, it would be significantly more difficult to discriminate between an uneventful game and one with a home run.

One popular approach for modeling the audio component is referred to as the *bag-of-audio-words* (BoAW) method. The BoAW method has the advantage of being an unsupervised approach and therefore does not require laborious human annotation efforts. The method is inspired by the well-established techniques in the text document (*bag-of-words*, BoW) and image document (*bag-of-visual-words*, BoVW) domains and has been recently used for audio document retrieval [1], song retrieval [2], copy detection [3], and MED tasks [4].

Figure 1 illustrates the basic BoAW method with the additional N-gram formation and term frequency selection (presented as shaded boxes) that are the focus of this paper. There are numerous basic variations to the basic pipeline such as the codebook size and classifier parameters. Results from exploring these variations on our experimental setup is presented in our related work [5]. The BoAW approach first generates a set of "words" (also called the codebook) via a clustering algorithm. This codebook is then used to quantize the features by replacing each feature with the index of the word it is closest to in the codebook. This process is referred to as the vector quantization step. The histogram (also referred to as a *word-vector*) is then generated by counting the number of appearances of each codeword in the file.

This method is nearly identical to the sister BoVW method, but differs more drastically from the BoW method. When working with text documents, the units are words occurring in natural language while in the image and audio domain the words are generated via a clustering algorithm to best represent the original feature space.

The BoAW approach is both similar to and different from the sister methods used in the text and image document domains. These relations allow us to draw parallels from the other fields' extensive research when seeking improvement on the audio variation. The BoAW approach is similar to BoW in that it occurs in one-dimension. The temporal information is lost when the words are thrown into the bag (when the histogram vector is generated). The two methods are, however, fundamentally different due to the nature of the "words." For text documents, the units are words occurring in natural language. Audio-words are generated through a clustering algorithm where the number of codewords needs to be user-specified and therefore are likely to differ from the natural dictionary in terms of cardinality and word distribution. The BoVW method is similar to BoAW in this respect. The original feature space in the image document domain describes components of the image such as an edge geometry or a color, and the "words" also need to be generated via clustering. However, BoVW differs in that it is two-dimensional and the original feature space is often scale-invariant. Both sister methods have explored ways to involve context in order to improve performance, whether in one or two dimensions.

In this paper we capitalize on the similarities with the BoW and BoVW methods. We use an N-gram modeling approach that we have found to enhance performance in text document classification tasks [6, 7] as well as recently in audio document retrieval [8]. In contrast to text-related tasks, the discriminality and generality of audio-words is dependent on the codebook size, thus adding an extra consideration when employing the method. Further, representing the document with word pairs exponentially increases the number of possible terms. We therefore also apply simple term selection at various thresholds to reduce the feature dimension. By employing the N-gram representation with term selection techniques to the basic BoAW algorithm, we see improvements in the MED task.

The paper is organized as follows. In Section 3 we describe the N-gram extension for the BoAW approach. In Section 4 we then discuss the term filtering techniques and in Section 5 we discuss the experimental setup. This is followed in Section 6 with results for the different codebook sizes, term filtering thresholds, and n-gram representations explored in our work.

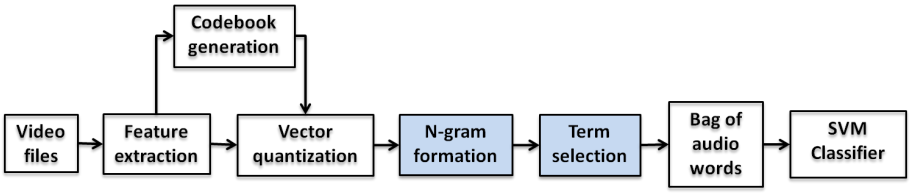

Fig. 1. Diagram of the basic BoAW method with the additional N-gram formation and term frequency selection (presented as shaded boxes) that are the focus of this paper.

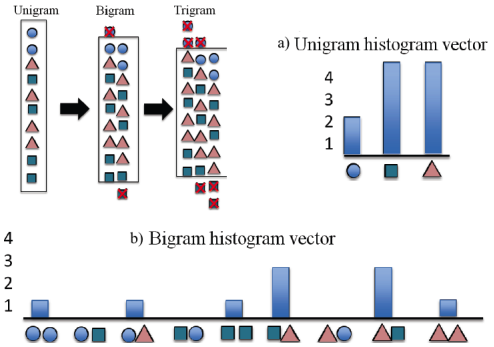

Fig. 2. Illustration of the bigram and trigram formation from the original unigram file as well as the histogram vector of the unigram and bigram for that file.

2. N-GRAM REPRESENTATION

The basic BoAW algorithm maps a single frame to a codeword index, ignoring the frames surrounding it. This is a drawback of the method. Previous work on text document classification has sought to use contextual information by representing words as N-grams instead of, or in addition to, the single word count. An N-gram is a concatenation of N consecutive "words". Figure 2 illustrates how the bigrams and trigrams are generated from the original unigrams as well as the resulting change in the histogram vector. A main observation is the exponentially increasing length and sparsity of the histogram vector. If D terms were used in the original codebook D^N possible N-grams can appear in the documents. For this reason, we adopt term filtering to reduce the vector dimension. Ideally, we would remove the words that are not discriminative in the MED task.

It is often observed that performance degrades when using any sequence beyond length 3, and this was further confirmed by authors [6]. We therefore explore only unigrams, bigrams, and trigrams in this paper. Previous work has used N-grams in combination with the original unigram vectors [6, 7, 8]. We also examine performance when using different combinations of unigrams, bigrams, and trigrams to form a final, larger histogram vector.

2.1. Term Filtering Methods

Despite the large number of words in natural language, many do not occur in a given document, leading to sparse and high-dimensional word vectors. Although the audio words are initially fewer and more evenly spread than natural language words, the formation of N-grams increases the sparsity and dimension so that the dictionary approaches that of the natural language. Like with natural language words, many of the N-grams rarely appear in the dataset and therefore add minimal value to the MED task. Term filtering is used to eliminate these extra terms and decrease the histogram feature dimension.

We explored two term selecting methods:

Term Frequency(TF) One common technique is to remove terms based simply on their frequency [6, 7]. This technique is derived from the assumption that low-frequency words are less likely to contribute to the document classification.

Term Frequency - Inverse Document Frequency (TF-IDF) Words that are common but appear in all documents (such as "the" for English texts) do not hold discriminative power even though their TF is high. Therefore, the TF-IDF is commonly used [9, 2]. TF-IDF for word i is calculated as $\frac{tf_i}{df_i}$ where tf_i is the term count across all files in the training corpus and df_i is the number of training documents in which the word appears.

3. EXPERIMENTAL SETUP

Our experiments were run using what is referred to as a *verification* or *one-against-all* setup. For each video event, a file is labeled as *in-class* or *out-of-class*. Examples include *Parade* and non-*Parade* as well as *Birthday party* and non-*Birthday party*. Therefore, the Set A experiments consist of 5 and Set B 20 binary classification experiments.

We ran our experiments in two passes. In both passes, we used data from the National Institute of Standards (NIST) development set provided for the TRECVID 2011 and 2012 multimedia event detection track [10]. The videos were provided in MP4 format. We extracted the audio components with a sampling rate of 16 kHz. The first pass used a smaller set of data (Set A) to explore the impact of variations to the BoAW method on system performance. We explored different codebook sizes and filtering techniques. The best-performing experimental setup was determined on the smaller set. We then applied the setup to a larger setup (Set B).

To measure system performance, we generate Detection Error Tradeoff (DET) curves, which show the tradeoff between false alarm errors and missed detections. In this paper, the DET curves are generated with plotting software available from the NIST website [11]. While DET curves clearly illustrate system performance over all possible threshold values, it is difficult to compare performance across experiments, as some may perform better in the low probability of false alarm (pFA) region while others perform better in the low miss probability (pMiss) region. The curves are generated by plotting pMiss at fixed values of pFA. We therefore calculate the average pMiss (APM) across all pFA values as a final metric of system performance.

Term Filtering Method	#Terms	APM
TF	10,000	**0.254**
TF-IDF	10,000	0.271
TF	50,000	**0.245**
TF-IDF	50,000	0.268

Table 1. Average probability of miss (APM) for different term filtering methods and number of terms using the bigram only representation based off codebook size of 2000.

3.1. Bag-of-Audio Word System Parameters

We explore the BoAW approach with N-gram extension and different term filtering methods. While we vary the codebook size to examine the results on system performance using these new BoAW modifications, the other parameters (front-end features, histogram normalization, and MED classifier) remain constant and are adopted from the best setup in our previous work [5].

For front-end features we used Mel frequency cepstral coefficients (MFCCs). The features are computed for every 10-ms audio segment and are extracted using a hamming window with 50% overlap. The features consist of 12 MFCCs as well as the log energy. The first and second derivates of each coefficient as well as the log energy are concatenated with the original features to result in a 39-dimensional feature vector.

We used histogram vectors with no normalization as features for support vector machine (SVM) classifiers. The SVM used a histogram intersection kernel. One SVM was trained to perform a binary classification for each video event.

4. RESULTS

We first present results on varying the N-gram representation, filtering technique, and codebook size on Set A. We then select the best performing of these setups and apply that setup to Set B to gain a stronger sense of the degree to which the new representation improves performance.

4.1. Results on Set A

We first examined the results when using the different term filtering methods. For two term thresholds (10,000 and 50,000 terms) we examined results using the bigrams only for codebook size 2000. As is evident by Table 1, the TF approach clearly works better than the TF-IDF term filtering method. When examining the document frequency distribution, the document frequency (DF) of a term is closely related to the TF (those with a high TF value also have a high DF value). Therefore, dividing by the DF will dampen the discriminative of the TF counts, resulting in a weaker set of selected terms.

We then explored four representations (unigrams only, bigrams only, unigrams + bigrams, unigrams + bigrams + trigrams) for various TF thresholds. The threshold is applied to all terms. For example, a TF of 100 indicates that the unigrams, bigrams, and trigrams occurred at least 100 times in the training data of Set A. To ensure that results were not codebook size specific, we present results for codebook size 1,000 in Figure 3 and 5,000 in Figure 4. These figures show that combining bigrams and unigrams performs better than the bigrams or unigram system individually, especially at lower TF threshold values. Also note that further including trigrams lends to performance nearly identical to the bigram+unigram system.

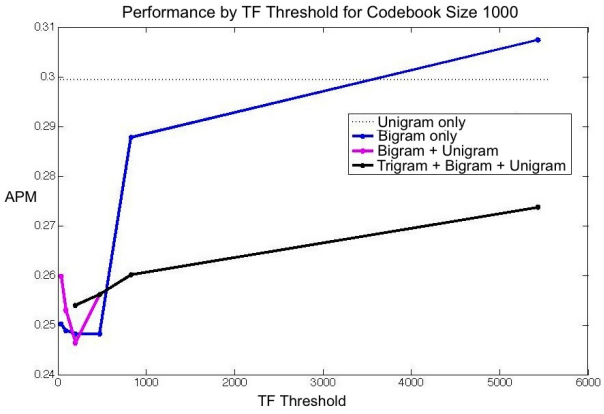

Fig. 3. Average probability of miss (APM) at various term frequency (TF) thresholds when using 1000 codewords.

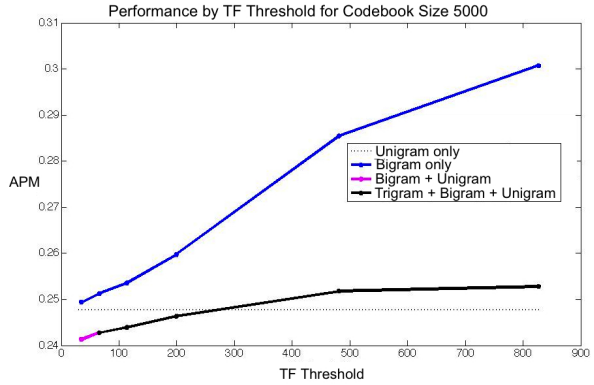

Fig. 4. Average probability of miss (APM) at various term frequency (TF) thresholds when using 5000 codewords.

Many of the frequently occurring bigrams were doubles (the same unigram occurring consecutively). When considering trigrams, the same is true, so it is not surprising that adding trigrams to the unigrams+bigrams system does not provide system gain.

Since the bigram+unigram system performed better than the bigram-only system and the same as the trigram+bigram+unigram system, we used the bigrams+unigram setup for the remainder of the experiments.

The discrimination power of the audio words depends on the codebook size; we therefore explored the change in performance when varying codebook sizes. Results from these experiments are presented in Figure 5. While the bigram+unigram setup clearly shows improvement for smaller codebook sizes, for the larger codebook (5000 words), we see improvement only when expanding the histogram vector to include less frequent bigrams.

4.2. Results on Set B

We applied the bigram+unigram representation to a larger dataset and compared it to the unigram only system. Since the purpose of applying the N-gram system to Set B is to gain a stronger sense of the N-gram effect on the BoAW approach for MED, we did not ex-

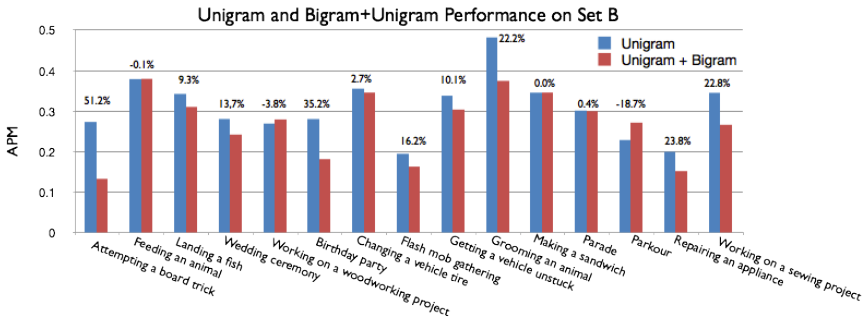

Fig. 6. APM by event for Set B comparing results using bigrams + unigrams and unigrams only.

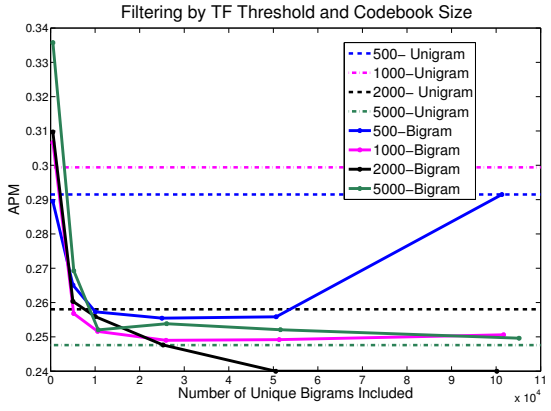

Fig. 5. Average probability of miss (APM) as function of codebook size and number of unique bigrams used to represent the documents. This illustration presents results with the bigram-only representation and compares it to the unigram-online baseline.

periment with model parameters as was done for Set A. We used a codebook size of 1000 and included the bigrams with a TF in the top 4% of those appearing in the training files for Set A.

Figure 6 presents the results on Set B. Of the 15 events in Set B, 11 showed improved performance when incorporating the bigrams into the histogram representation. The APM, when averaged across all video events, decreased from 0.308 to 0.270. Of the events, *Attempting a board trick* and *Birthday party* showed the greatest percent improvement, while *Parkour*'s APM decreased the most.

5. CONCLUSION

We examined the impact of the N-gram BoAW approach for multimedia event detection tasks. We found that overall the N-gram representation improves performance, especially when starting with smaller codebook sizes. When evaluated on the TRECVID 2011 and 2012 dataset, the average probability of miss improves from 0.308 to 0.270.

The N-gram representation addresses the basic BoAW method's lack of context usage. Another criticism is in the quantization step where information regarding the closeness of the original feature vector to the nearest codeword is disregarded. Future work should address this issue by employing some closeness function to weigh the term counts when generating the histogram. Since we found that N-grams improve the BoAW system, we will be exploring this soft-assignment approach to N-grams in the future.

6. ACKNOWLEDGMENTS

We thank Greg Myers and Professor Robert M. Gray for their valuable discussions.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC0067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes nonwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

This material is also based upon work supported by the National Science Foundation under Grant No. DGE-1147470.

7. REFERENCES

[1] G. Chechik, E. Ie, M. Rehn, S. Bengio, and D. Lyon, "Large-scale content-based audio retrieval from text queries," in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, ser. MIR '08. New York, NY, USA: ACM, 2008, pp. 105–112.

[2] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to content-based audio retrieval," *ISMIR 2008*, pp. 295–300, 2008.

[3] Y. Uchida, S. Sakazawa, M. Agrawal, and M. Akbacak, "KDDI Labs and SRI International at TRECVID 2010: Conent-Based Copy Detection," in *NIST TRECVID 2010 Evaluation Workshop*, 2010.

[4] Y. Jiang, X. Zeng, a. S. B. G. Ye, D. Ellis, M. Shah, and S. Chang, "Columbia-UCF TRECVID 2010 Multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in *NIST TRECVID Workshop*, 2010.

[5] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event detection," in *Proceedings of Interspeech*, 2012.

[6] C. Tan, Y. Wang, and C. Lee, "The use of bigrams to enhance text categorization," *Information Processing and Management*, vol. 38, no. 4, pp. 529 – 546, 2002.

[7] J. Fürnkranz, "A study using n-gram features for text categorization," *Austrian Research Institute for Artificial Intelligence*, vol. 3, no. 1998, pp. 1–10, 1998.

[8] S. Kim, S. Sundaram, P. Georgiou, and S. Narayanan, "An N -gram model for unstructured audio signals toward information retrieval," in *Multimedia Signal Processing, 2010 IEEE International Workshop on*, 2010.

[9] J. Yang, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the International Workshop on Multimedia Information Retrieval*. ACM, 2007, pp. 197–206.

[10] TRECVID multimedia event detection 2011 evaluation. [Online]. Available: http://www.nist.gov/itl/iad/mig/med11.cmf

[11] NIST DETware v.2. [Online]. Available: http://www.itl.nist.gov/iad/mig/tools/