A Bayesian Framework for Fusing Multiple Word Knowledge Models in Videotext Recognition

DongQing Zhang and Shih-Fu Chang
Department of Electrical Engineering, Columbia University
New York, NY 10027, USA. {dqzhang, sfchang}@ee.columbia.edu

Abstract

Videotext recognition is challenging due to low resolution, diverse fonts/styles, and cluttered background. Past methods enhanced recognition by using multiple frame averaging, image interpolation and lexicon correction, but recognition using multi-modality language models has not been explored. In this paper, we present a formal Bayesian framework for videotext recognition by combining multiple knowledge using mixture models, and describe a learning approach based on Expectation-Maximization (EM). In order to handle unseen words, a back-off smoothing approach derived from the Bayesian model is also presented. We exploited a prototype that fuses the model from closed caption and that from the British National Corpus. The model from closed caption is based on a unique time distance distribution model of videotext words and closed caption words. Our method achieves a significant performance gain, with word recognition rate of 76.8% and character recognition rate of 86.7%. The proposed methods also reduce false videotext detection significantly, with a false alarm rate of 8.2% without substantial loss of recall.

Keywords: Videotext recognition, Video OCR, Video indexing, Information Fusing. Multimodal Recognition.

1. Introduction

Videotext recognition is difficult due to low resolution, diverse fonts, size, colors, styles, and cluttered background. There are two categories of videotext in digital videos: *overlay text*, which is added by video editors; *scene text*, which is embedded in real-world objects. Although overlay text and scene text share some common properties, overlay text is easier to detect than scene text in general and is the focus of this paper. A complete videotext recognition system involves both issues of detection and recognition. Videotext detection has been extensively studied in recent years [1,2,3,4], but videotext recognition is much less explored. Some relevant works in videotext recognition include template matching [1], SVM classifier [5], and those using document OCR engines [2] etc. Enhancement schemes have been studied by many researchers, for example, temporal averaging of multiple frame [1,4], spatial interpolation [4], font context [6] and word correction by dictionary [1]. But the potential of using language models, especially multimodal models, has not been explored. The most related idea is word correction using edit distance by dictionary [1]. But such method works well only when the character recognition error rate is low.

The language model has been widely adopted in speech recognition [7] and handwritten recognition [8]. To construct a language model, one needs text corpora containing a large number of text documents. The problem encountered by videotext recognition is the difficulty in acquiring sufficient data from videos for language model construction. Language models can be created from general linguistic corpora, but it may be inaccurate. Recognition using multimodality is another way to enhance performance. Today’s broadcast videos usually are associated with many text sources, such as closed captions, and online web documents. These documents can be used to enhance the videotext recognition, since they contain words which are often related to words in videotext. However, solely relying on external document source is not sufficient. Take the example of closed caption, only about 40% to 50% of videotext words can be found in closed caption. Therefore, there is great promise in combining language models from different sources with different modalities.

This paper aims at this problem by constructing a Bayesian framework to fuse the word knowledge models from multiple sources. The framework is established using mixture models and its training approach is derived from the Expectation-Maximization (EM) algorithm. In order to increase the recognition performance of characters and unseen words, a smoothing scheme is derived to back-off the word recognition to the baseline character recognition approach. To validate the framework in the practical domain, we use the closed captions in videos and linguistic corpus to extract the multiple word knowledge models. The knowledge model from closed caption is built by learning a unique distribution model of the time distance between the videotext and their matched counterpart in closed caption.

The general linguistic knowledge model is extracted from the British National Corpus. We also developed a multiple frame integration technique as a post processing stage. Besides using multiple frame averaging [1], we explored a multiple frame voting scheme, which first identify identical text blocks in different frames, then use voting process to select the dominant word recognition output among the text blocks. Figure 1 shows our system diagram for videotext recognition fusing multiple word knowledge models.

We evaluate the system on six news videos from three different channels with about 1200 videotext words. The experiments showed a 51% accuracy improvement comparing the proposed method with the baseline technique. The combined model also performs better than individual models by 4.4%. When used as a post-processing step, the word recognition technique plus temporal voting also help reduce videotext detection false alarms significantly.

The paper is organized as follows: Section 2 briefly describes the pre-processing approaches including detection, binarization and segmentation. Section 3 presents the baseline character recognition system. Section 4 describes the Bayesian framework for word recognition. Section 5 presents a prototype model using closed caption and the British National Corpus. Section 6 describes experiments with the results.

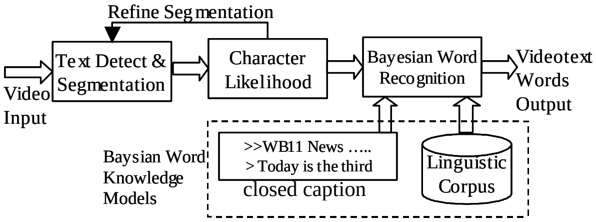

Figure 1. Flowchart of the proposed videotext recognition system. It fuses word knowledge models from closed caption and linguistic corpus.

2. Pre-processing

We first briefly describe the preprocessing stage including videotext detection, binarization and segmentation. Careful design of these modules is important for later robust recognition processes.

2.1. Videotext detection

We use the videotext detection algorithm developed in our prior works [9,10] to extract the videotexts from the videos. The system first computes texture and motion features by using the DCT coefficients and motion vectors directly from the MPEG compressed streams without full decoding. These features are used to detect candidate videotext regions, within each of which color layering is used to generate several hypothetical binary images. Afterwards a grouping process is performed in each binary image to find the character blocks. Finally, a layout analysis process is applied to verify the layout of these character blocks using a rule-based approach.

2.2. Binarization and character segmentation

Binarization and character segmentation is difficult for videotext due to color variation and small spacing between characters. We developed iterative and multi-hypothesis methods to handle these problems.

Fixed threshold value is not suited for videotext binarization, because videotext intensity may show great variations. We developed an iterative threshold selection method to dynamically adjust the intensity threshold value until the broken strokes of characters appear. Such idea is similar to that proposed in [11].

The character segmentation method is based on local minima searching in the vertical projection profile [10]. A segmentation line is identified by thresholding the projection profile.

To reduce the recognition errors caused by character segmentation, multiple segmentation hypotheses are used to produce candidate characters. Prior work in [1] searched for the optimal hypothesis using dynamic programming. In our case, since the number of candidate segmentation points is usually small (one to twenty, mostly less than four), an exhaustive search is performed.

Word segmentation is needed to find complete word segments for recognition. To realize this, the median value of the character spacing is first calculated. If the spacing between two characters is larger than two times the median value, the segmentation line is marked as a word boundary.

3. Character recognition

The character recognition step involves the feature extraction from a single character and shaping of character conditional density functions.

3.1. Character feature extraction

The feature set for character recognition include *{Zernike magnitude; Direction proportion; 1st-order peripheral features; second-order peripheral features; vertical projection profile; horizontal projection profile.}* These features are selected from a larger feature set manually.

For Zernike moment features, readers are referred to the paper [12] for complete description. And the description of other features can be found in [13]. These features lead to an overall dimension of 207.

3.2. Character conditional density function

The character condition density function is modeled using Parzen window [14]. One can also use Gaussian Mixture Model (GMM). However, the GMM has the overfitting problem when the dimension of data space is high. Regularization can be introduced to handle this problem, such as using Bayesian penalty term [14]. However, in our experiments, we found that the Parzen window approach outperforms the regualized GMM.

For Parzen Window, the sample points are generated using a distortion modeling procedure. We apply a variety of geometrical distortions to each standard font image to obtain training samples. Distortions include 4 fonts, 3 aspect ratios, 3 character weights, and 5 sizes. The size variation has little impact on recognition, therefore we average the feature vectors corresponding to different sizes. This leads to 36 sample data for each character. A Gaussian kernel is used for the Parzen window method. The density function can be adjusted by changing the variance of the Gaussian kernel. In order to maximize the character recognition performance, the variance of the Gaussian kernel needs to be tuned using training data.

Given a feature vector for a character image, a baseline system for the character recognition is to compare the likelihood values of different characters and select the character corresponding to the highest likelihood. We will refer to this method as baseline character recognition method throughout the paper.

4. Bayes word recognition framework

The videotext word recognition problem can be formulated using Bayesian method or the maximum a-posterior (MAP) recognition as:

$$\begin{aligned}\hat{w} &= \arg\max_{w}\ p(w \mid \mathbf{x}) \\ &= \arg\max_{w}\left[\log\ p(\mathbf{x} \mid w) + \log\ p(w)\right]\end{aligned} \quad (1)$$

where **x** is the word feature vector, and *w* is a candidate word. $p(\mathbf{x} \mid w)$ is called *word observation model* constructed from the character conditional density function. $p(w)$ is called *language model* in the community of speech recognition. It specifies the prior probably of each word. Here we not only use the linguistic corpus but also the models from other sources, such as closed caption, thus we call $p(w)$ as *Word Knowledge Model* (WKM).

4.1. Word conditional density function

Word observation model is constructed from the single character conditional density function. Suppose after segmentation, *N* character images are segmented from a word image, and the feature vectors of these characters are $x_1, x_2, ..., x_N$. Then the constructed word feature vector is $\mathbf{x} = \{x_1, x_2, ..., x_N\}$. The word observation model denoted by word likelihood function therefore is:

$$\begin{aligned} p(\mathbf{x} \mid w) &= \prod_{i=1}^{N} p(x_i \mid w) = \prod_{i=1}^{N} p(x_i \mid c_i), \quad |w| = N \\ &= 0, \qquad\qquad\qquad\qquad\qquad\quad |w| \neq N \end{aligned} \quad (2)$$

where $c_i \in \mathrm{A}$, A is the alphabet, which currently include 62 characters (26 letters with lower and upper case, plus 10 digits).

4.2. Fusing multiple word knowledge models

As discussed earlier, the language model *p(w)* could be obtained by using linguistic corpus; but it may be inaccurate due to the limit of training data. Combination of multiple models could be a remedy to this problem by adding other relevant knowledge into the general model. These additional sources can be acquired easily in today's distributed information environment, for example closed caption in the video stream, or online documents on related web sites etc.

Suppose that we have obtained or learned the language models from different sources. We denote such models as $p(w \mid K_i)$, where K_i denotes the information source *i*. Each word knowledge model (WKM) covers a subset of all possible words. Suppose the subset covered by each model is S_i. We use a linear combination of these WKMs to form a mixture word knowledge model:

$$p(w) = \sum_{i=1}^{C} a_i\, p(w \mid K_i) \quad (3)$$

subject to:

$$\sum_{w_j \in S_i} p(w_j \mid K_i) = 1 \ \text{and} \ \sum_{i=1}^{C} a_i = 1 \quad (4)$$

where *C* is the number of sources. The combined model will cover all the words that belong to $S = \bigcup_i S_i$.

To obtain the weighing vector $\mathbf{a} = \{a_1, a_2, ..., a_c\}$, we use the videotexts in the training set. The data needed for training is much less than that required by constructing a general *language model* due to the small size of the parameter space. The optimal weighting vector should maximize the joint probability of the training set based on the maximum likelihood training, i.e.:

$$\mathbf{a} = \arg\max_{a} \prod_{w_i \in T} p(w_i | a), \quad \text{subject to} \ \sum_{i=1}^{c} a_i = 1 \quad (5)$$

where *T* is the training set. Although this is a standard constraint maximization problem, it is actually difficult to solve and get the closed form solution. However, it can be solved iteratively by using the Expectation-Maximization (EM) method [15]. The update equations based on EM is:

$$p(j \mid w_i) = \frac{p(w_i \mid K_j)a_j^{old}}{\sum_{j=1}^{C} p(w_i \mid K_j)a_j^{old}}, \ a_j = \frac{1}{N}\sum_{i=1}^{N} p(j \mid w_i) \quad (6)$$

where *N* is the number of the training videotext words.

4.3. Recognition of unseen word

The combined model is usually unable to cover all videotext words in a video. For instance, in news video, about 15% videotext words cannot be found in either closed caption or linguistic corpus. Directly applying language models to those unseen words may change correctly recognized words and thus increase recognition errors.

To handle this problem, we use a method to back-off the word recognition process to character recognition in certain condition. Such method has been used in speech recognition [16]. We derive the back-off method based on the Bayesian recognition framework, where the word knowledge model is decomposed as:

$$\begin{aligned} p(w) &= p(w \mid w \in S)p(w \in S) + p(w \mid w \notin S)p(w \notin S) \\ &= p(w \mid w \in S)(1-c) + p(w \mid w \notin S)c \end{aligned} \qquad (7)$$

where S denotes the word dictionary covered by the knowledge models, and *c* is the probability that a videotext word falls out of *S*. $p(w \mid w \in S)$ is the prior specified by the knowledge model. For an unseen word, this term will be zero. $p(w \mid w \notin S)$ is a hypothetical distribution for all words that are not in *S*, whose value is zero for all seen words. Based on these, we will get the following back-off condition when considering a candidate word w_S from *S*:

$$[\log p(\mathbf{x} \mid w_S) + \log p(w_S \mid w_S \in S)] - \log p(\mathbf{x} \mid w_{\bar{S}}) < d \quad (8)$$

where $w_{\bar{S}}$ is the baseline character recognition output. *d* is the back-off threshold, which can be trained using a straightforward method. Note that back-off is only applied when the baseline recognition result cannot be found in *S*. The derivation of this equation and estimation of *d* using the training set is discussed in [17].

4.4. Post-processing

A unique nature of videotext comparing with document image text is the redundancy of text images: caption text usually stays on the video for a few seconds, resulting in duplicates of the same text blocks in consecutive frames. Prior systems employed temporal averaging to take advantage of such redundancy. However, we found that although temporal averaging is able to reduce the additive noise, it cannot fully avoid the false recognition caused by segmentation error or character corruption by background perturbation. Thus we propose to use a multiple frame voting method using recognition results from each individual frame. To realize this, we group similar text blocks within a local temporal window together. The similarity is measured using sum of pixel-to-pixel color distances between each videotext blocks. The voting process is performed by selecting the most frequent output among all word recognition output in the same group. Such algorithm effectively eliminates the false recognition due to erroneous character segmentation.

The above temporal voting process not only improves the word recognition accuracy, but also improves the text detection accuracy. Detection false alarm is filtered out if the posterior of a word is lower than certain threshold before voting, or the word count of the most frequent word in a group is one.

5. Prototype using closed caption and British national corpus

We realize a prototype of the proposed framework and algorithms by using the closed caption (CC) stream associated with the video and an external linguistic corpus, British National Corpus (BNC). The multiple knowledge models under these two sources can be written as:

$$p(w) = a_{cc}\, p(w \mid CC) + a_{BNC}\, p(w \mid BNC) \qquad (9)$$

5.1. Building the word knowledge model from closed caption

For a word drawn from the CC model, its prior is assumed to only depend on two factors: (1) the time distance between the CC word and the videotext (VT) word being recognized, $\Delta t = t_{sw} - t_{vw}$, and (2) The part-of-speech (POS) of the CC word, *S*. Words far from the videotext word have lower prior probabilities. Words of different POS (e.g., verb vs. noun) have different priors of appearing in the videotext.

Using CC we can construct a CC wordlist (CCW), which includes all words that occur in CC at least once. If there are multiple instances of a word, CCW only keeps one instance. Thus we model the following word prior:

$$p(w = w' \mid CC) = \frac{1}{C} \sum_{w_i = w'} p(w = w_i \mid \Delta t_i, S_i) \qquad (10)$$

where *w'* is the word in CCW that may appear multiple times, w_i is the *i-th* instance of the word *w'* appearing in the CC stream, $\Delta t_i, S_i$ is the time distance and POS of w_i respectively, and C is a normalization constant

$$C = \sum_{w' \in CCW} \sum_{w_i = w'} p(w = w_i \mid \Delta t_i, S_i) \qquad (11)$$

Because training such model is complicated due to the presence of POS, we use a simplified model: when the POS of w_i is a stop word or preposition, the probability is zero, otherwise it depends solely on the time distance. In other words, only non-stop word and non-preposition words are considered in training and recognition.

The likelihood function can be in various function forms, which can be determined by comparing the empirical distribution and the estimated distribution using

Chi-square hypothesis test. We used two hypotheses: Gaussian function and exponential function. The hypothesis test shows that the exponential function is closer to the empirical distribution. The non-causal exponential time distance density function we adopted is as follows:

$$p(w = w_i \mid \Delta t_i, S_i \neq SP) = \frac{1}{2l} e^{-|\Delta t|/l} \qquad (12)$$

SP denotes stop word or proposition word. This is a double exponential model (DPM). For a causal model, it is straightforward to modify the equation (12) by removing the right tail of the DPM.

To train the parameter of this model, a standard maximum likelihood approach is used [15] using the pool of all matched word pairs in CC and videotext. Based on our experiments, $l_l = 0.045$ provides satisfactory results.

According to CC model, a word that cannot be found in CC will be assigned a zero probability of $p(w \mid CC)$.

5.2. Knowledge model from BNC

Word knowledge model is also extracted from the British National Corpus (BNC) [18]. British National Corpus includes a large number of text documents for training language models. BNC also provides the word lists with the use frequency of each word. We use the written English version lists containing about 200,000 words. The list includes all inflected forms of each word stem as well as their frequency. In videotext, stop words are rarely used, but they hold highest frequency in the BNC word list. In order to get a more accurate distribution function, the word frequencies of these stop words are manually re-assigned to a small value. After these processes, the word knowledge model extracted from the BNC is the normalized version of the word frequency:

$$P(w \mid BNC) = Freq(w) \Big/ \sum_{w_i \in BNC} Freq(w_i) \qquad (13)$$

There is spelling difference between British words and English words [19]. However, in BNC word list, both British spelling and American spelling are included [18] for most words. In experiments, we confirmed spelling differences did not result in performance degradation.

6. Experiments

Our experiment data include six news videos from three channels broadcasted on different days. The videos include different stories, different fonts and intensity values of videotext. The format of the videos is MPEG-1 with SIF resolution (352x240 pixels). The overall duration of the test set is about six hours.

A cross validation process is used to evaluate the algorithms. That is, the methods are trained using videos from two channels and are tested using the videos from remaining channels. During the training process, the estimated parameter set includes parameters of the time distance distribution for the closed caption model, the weighting vectors of the mixture model, and the back-off threshold. The variance of the Gaussian kernel for the Parzen window is also determined empirically using the training set.

In the testing stage, the detection program is first carried out to detect the super-imposed text blocks. The overall detection recall rate is 97% and the initial precision rate of detection is 70%. The detected text blocks are then passed to binarization, segmentation, recognition, and post-processing. After word recognition and post-processing, the false detections are filtered, leading to an improved precision rate of 91.8% with degraded recall rate of 95.6 %.

The performance of recognition within the correctly detected set is shown in Table 1. Here one word recognition error is counted as long as there are one or more character recognition errors in the word. The improvement in character recognition is large (+19.6%); the improvement in word accuracy is even more significant (+51%).

Table 1. Recognition Accuracy

Videos	Char Accuracy		Word Accuracy	
	B	K	B	K
w#:1422	67.1%	86.7%	25.8%	76.8%

Legends: B: baseline character recognition, K: Knowledge based recognition, w#: total number of words.

Figure 2 shows some examples of the videotext recognition results, with different types of success and failure grouped together. Under each text image, two recognition results are shown – the left one shows the result using the baseline method while the right one shows the result using the knowledge-based recognition method combining both BNC and CC models. The one in the bold face is the final result selected by our system using the back-off procedure described in Section 4.3.

Table 2. Contribution of CC and BNC

Videos	BNC	CC	CC+BNC	CC Cont
w#:1422	72.4%	48.6%	76.8%	4.4%

Legends: BNC: use BNC only, CC: use CC only, CC+BNC: use both BNC and CC, CC Cont: CC

We also conducted separate tests to study the individual contributions from each knowledge model. In table 2, the “BNC” column shows the performance using the BNC model only, the “CC” column shows the performance using the closed caption model only, the “CC+BNC” column shows the results combining both models. The results show that when used alone, the BNC model is

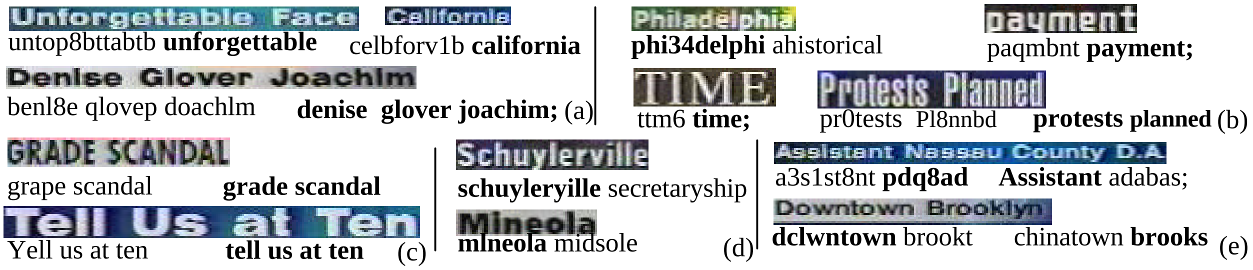

Figure 2. (a) Some results of knowledge models (b) Recognition of videotexts with various styles (c) False recognition corrected by the surrounding CC words (d) Back-off triggered due to unseen words (e) False recognition due to poor segmentation and thresholding.

more effective than the CC model. When they are combined, the CC model adds 4.4% accuracy improvement on top of the result using the BNC model only. When we further analyzed the data, we found the gain most came from the refinement to the word prior probability. Figure 2(c) shows several examples of errors corrected by adding the CC model.

7. Conclusion

We have developed a Bayesian framework for videotext recognition, in which the prior probabilities of words are estimated by combining multiple word knowledge models. Our current prototype includes synchronized closed caption and linguistic corpus, British National Corpus, as knowledge models. We used an EM based method for learning the fusing model. We have also developed a back-off process to handle unseen words in the model. To estimate the priors for words in the closed captions, we used an effective statistical model taking into account the time distances of the closed caption word to the videotext. The experiments show that such multi-modality knowledge fusing method results in significant performance gain. When combining the word recognition and temporal voting in a post-processing stage, the false detection of text detection is also significantly reduced.

8. References

[1] T. Sato, T. Kanade, E. Hughes, and M. Smith, "Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions", Multimedia Systems, 7:385-394, 1999.

[2] R. Lienhart, W. Effelsberg, "Automatic text segmentation and text recognition for video indexing", Multimedia System, 2000.

[3] J.C. Shim, C. Dorai and R. Bolle, “Automatic Text Extraction from Video for Content-Based Annotation and Retrieval", Proc. 14th International Conference on Pattern Recognition, volume 1, pp. 618-620, Brisbane, Australia, August 1998.

[4] H. Li; D. Doermann, O. Kia, “Automatic text detection and tracking in digital video”, IEEE Trans. on Image Processing, Vol 9, No. 1, January 2000.

[5] C. Dorai, H. Aradhye, J.C. Shim, End-to-End Videotext Recognition for Multimedia Content Analysis. IEEE Conference on Multimedia and Exhibition (ICME 2001).

[6] H. Aradhye, C. Dorai, J.C. Shim, Study of Embedded Font Context and Kernel Space Methods for Improved Videotext Recognition, IEEE International Conference on Image Processing (ICIP 2001).

[7] B. Gold, N. Morgan, “Speech and Audio processing”, John Wiley & Sons, Inc (1999).

[8] R. Plamondon, S.N. Srihari, “On-Line and Off-Line Handwriting Recognition: A comprehensive Survey”, IEEE Trans. on PAMI, Vol. 22, No. 1, Janury 2000.

[9] D. Zhang, and S.F. Chang, "Accurate Overlay Text Extraction for Digital Video Analysis", Columbia University Advent Group Technical Report 2003 #005.

[10] D. Zhang, R.K. Rajendran and S.F. Chang, "General and Domain-specific Techniques for Detecting and Recognizing Superimposed Text in Video", Proceeding of International Conference on Image Processing, Rochester, New York, USA.

[11] T. Ridler, S. Calvard, "Picture Thresholding Using an Iterative Selection Method", IEEE transactions on Systems, Man and Cybernetics, August, 1978.

[12] A. Khotanzad and Y.H. Hong, “Invariant Image Recognition by Zernike Moments”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 5, May 1990.

[13] R. Romero, D. Touretzkey, and R.H. Thibadeau, “Optical Chinse Character Recognition Using Probabilistic Neural Networks”, CMU Technical Report.

[14] D. Ormoneit and V. Tresp. Improved Gaussian Mixture Density Estimates Using Bayesian Penalty Terms and Network Averaging. In Advances in Neural Information Processing Systems, volume 8, The MIT Press, 1996.

[15] R.O. Duda, P.E. Hart, D.G. Stock, Pattern Classification. Wiley-Interscience, New York, NY, 2 ed., 2000.

[16] S.M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Trans. on Acoustics, Speech and Signal Processing, 35(3):400--401, 1987.

[17] D. Zhang, S.F. Chang, A Multi-Model Bayesian Framework for Videotext Recognition, ADVENT Technical Report 2003, Columbia University.

[18] British National Corpus, Web homepage: http://www.hcu.ox.ac.uk/BNC/

[19] Dictionary of American and British Us(e)age. URL: http://www.peak.org/~jeremy/dictionary/dict.htm