Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 577–584

Automatic Identification of Broadcast News Story Boundaries Using the Unification Method for Popular Nouns

Zainab Ali Khalaf[1,2]
[2]Department of Computer Science, College of Science, University of Basra, Iraq
Email: zainab_ali2004@yahoo.com

Tan Tien Ping
[1]School of Computer Sciences, Universiti Sains Malaysia USM, 11800 Penang, Malaysia
Email: tienping@cs.usm.my

Abstract—Herein we describe the latent semantic algorithm method for identifying broadcast news story boundaries. The proposed system uses the pronounced forms of words to identify story boundaries based on popular noun unification. Commonly used clustering methods use latent semantic analysis (LSA) because of its excellent performance and because it is based on deep semantic rather than shallow principles. In this study, the LSA algorithm with and without unification was used to identify boundaries of Malay spoken broadcast news stories. The LSA algorithm with the noun unification approach resulted in less error and better performance than the LSA algorithm without noun unification.

Keywords: spoken document; broadcast news; story boundary identification; latent semantic analysis

I. Introduction

Because nouns bear more semantic meaning than other parts of speech and because they are the main characteristics used to identify documents stories [1], natural language processing applications often focus on nouns as essential components of the documents being processed. Names of persons, for example, are useful noun components in natural language processing, especially during automatic sentence clustering. In recent years, spoken document processing has become a popular and interesting topic within the field of natural language processing. In general, spoken document processing adapts natural language processing applications using speech input rather than text input.

Processing spoken documents is challenging because of the word errors generated by the automatic speech recognition (ASR) process [2], [3]. Determining the boundaries of broadcast news stories is another obstacle to processing spoken documents. The lack of overt punctuation and formatting contributes to this problem. In order to retrieve information, the beginning and the end of the segments or paragraphs within a document must be determined [3]-[6]. The process of determining the boundaries of the segments in the text is not an easy process [3]-[7].

Word errors generated by the ASR process can occur when recordings are made in a noisy environment or when pronunciation is unclear. The latter is especially true for vowel letters. An example from a Malay broadcast news story is as follows: The name of a professional badminton player was written four different ways in four sentences when converted from spoken news to written news by the ASR system (lee chong wei, choong wei, chong wee, and chan wee). The conversion problem was related to the vowel sound, in that the [u:] sound can be written as “oo, o, ou, ew, ue, u, and ui” and the [i:] sound can be written as “ee, ea, ei, and ie.” Silent sounds (pronounced n+ unpronounced g) also pose problems for ASR [8], [9].

Identification of story boundaries with the added problem of pronunciation errors is a complicated task. It requires human knowledge of the rules of correct pronunciation of lexical items. To address these problems, we propose a new method to improve story boundary identification in spoken documents using the popular noun unification approach.

II. Related Work

The absence of punctuation and capitalization in spoken documents makes it challenging to automatically identify story boundaries in multimedia documents. Previous attempts have concentrated on three types of cues: visual cues, such as the presence of an anchor’s face [7] or motion changes [7]; audio cues, such as significant pauses or reset of pitch; and lexical cues, such as word similarity measures within speech recognition transcripts or closed captions of video [10], [11]. Cues from completely different modalities (audio, video, and text) are often consolidated to achieve better story boundary identification [7], [12].

Hearst et al. (1997) proposed the TextTiling approach to story boundary identification [10]. It is based on the straightforward observation that different topics usually employ different sets of words and that shifts in vocabulary usage are indicative of topic changes [10]. As a result, pairwise

978-1-4673-4471-5/$25.00

sentence similarities are measured across the text and a local similarity minimum implies a story boundary. Stokes et al. (2004) evaluated word cohesion using a lexical chaining approach; in this method, related words in a text are linked into chains, and a high concentration of chain starting and ending points is an indication of a story boundary [11]. These two approaches were recently used to segment speech recognition transcripts of spoken documents such as broadcast news [12], [13] and meetings [14]. Rosenberg et al. (2006) presented results from a broadcast news story boundary identification system developed for the SRI NIGHTINGALE system, which was applied to English, Arabic, and Mandarin news show to provide input for subsequent question-answering processes [12]. Xie (2008) used word and subword multiple scales for story boundary identification and showed the robustness of subwords for reducing the impact of errors and improving identification of broadcast news story boundaries [13] . Wu (2009) used decision tree and maximum entropy methods to identify the positional story boundaries locally and then used a genetic algorithm to identify the final story boundaries [15].

III. Latent Semantic Analysis

The clustering of sentences can be used to find repeated information, and the clustering process is conducted by grouping similar sentences together. Previous studies have examined a number of different methods that can be used to identify similar sentences. Some of these methods use shallow techniques to detect the similarities in sentences (e.g., word or n-gram overlap), whereas other methods use a deep approach to examining the syntactic or semantic similarities. The latent semantic analysis (LSA) technique can be used to estimate both the similarity of word matching and semantic structures. Accordingly, the problem of synonymy is avoided [16], [17].

Spoken documents are typically scanned and split into sentences throughout the preparation process, and then term-by-sentence matrices (TSMs) are ultimately created. One of the payoffs of using LSA is that it reduces dimensionality and thus results in quicker clustering. When the matrix is prepared, it is subjected to singular value decomposition (SVD) (Figure 1) [16]. The SVD formula can be stated as follows:

$$A = L_{EV} * S * R^{T}_{EV}$$

Any rectangular matrix A (i.e., a TSM matrix) with order t×s is decomposed into three matrices (L_{EV}, S, R^{T}_{EV}). The matrix L_{EV} contains the left eigenvectors of A and describes the relationship between terms (rows) and sentences (columns), or it refer to a term-to-concept similarity matrix resulting from the equation $L_{EV} = A^{T}A$. The matrix S is an m × m diagonal matrix with the entries sorted in decreasing order. The entries of the S matrix are the singular values (eigenvalue), and the S matrix describes the relative strengths of each concept. The R^{T}_{EV} matrix, which is defined by the equation $R^{T}_{EV} = A\,A^{T}$, contains the left eigenvectors of A, and this matrix refers to a sentence-to-concept similarity matrix [16].

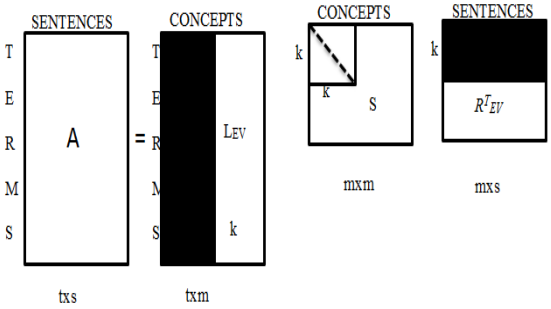

Fig. 1. Singular value decomposition (SVD)

The functionality of LSA will be explained using an example from the term similarity calculation. Consider Table I, which consists of four sentences from technical reports.

TABLE I. EXAMPLE INCLUDES FOUR SENTENCES

S1	Shipment of gold damaged in a fire
S2	Delivery of silver arrived in a silver truck
S3	Shipment of gold arrived in a truck
S4	Gold silver truck

1. The TSM (Table II) is constructed as follows:

TABLE II. TERM-BY-SENTENCE MATRIX (TSM)

	S1	S2	S3	S4
A	1	1	1	0
arrived	0	1	1	0
damaged	1	0	0	0
delivery	0	1	0	0
Fire	1	0	0	0
Gold	1	0	1	1
In	1	1	1	0
of	1	1	1	0
shipment	1	0	1	0
silver	0	2	0	1
truck	0	1	1	1

2. SVD is used to decompose the A Matrix into three matrices.

L_{EV} =

0.3966 -0.1282 -0.2349 0.0941
0.2860 0.1507 -0.0700 0.5212
0.1106 -0.2790 -0.1649 -0.4271
0.1523 0.2650 -0.2984 -0.0565
0.1106 -0.2790 -0.1649 -0.4271
0.3012 -0.2918 0.6468 -0.2252
0.3966 -0.1282 -0.2349 0.0941
0.3966 -0.1282 -0.2349 0.0941
0.2443 -0.3932 0.0635 0.1507
0.3615 0.6315 -0.0134 -0.4890
0.3428 0.2522 0.5134 0.1453

S =

4.2055 0.0000 0.0000 0.0000
0.0000 2.4155 0.0000 0.0000
0.0000 0.0000 1.4021 0.0000
0.0000 0.0000 0.0000 1.2302

R_{EV} =

0.4652 -0.6738 -0.2312 -0.5254
0.6406 0.6401 -0.4184 -0.0696
0.5622 -0.2760 0.3202 0.7108
0.2391 0.2450 0.8179 -0.4624

R^{T}_{EV} =

0.4652 0.6406 0.5622 0.2391
-0.6738 0.6401 -0.2760 0.2450
-0.2312 -0.4184 0.3202 0.8179
-0.5254 -0.0696 0.7108 -0.4624

The rank (r) of a matrix is the smaller of the number of linear independent rows and columns. SVD is used to reduce the rank and thereby the file size of the text. A reduced-rank SVD is performed on the matrix, in which the k largest singular values are retained, and the remainder is set to 0. The resulting representation is the best k-dimensional approximation of the original matrix in the least-squares sense [16]. Each sentence and term is now represented as a k-dimensional vector in the space derived by the SVD. In most applications the dimensionality k is much smaller than the number of terms in the TSM. In the above example, SVD ranks the concepts by importance for the text. By reducing the rank to 2, only the first two concepts are kept. Thus, the ranking matrices for the example are:

L'_{EV} =

0.3966 -0.1282
0.2860 0.1507
0.1106 -0.2790
0.1523 0.2650
0.1106 -0.2790
0.3012 -0.2918
0.3966 -0.1282
0.3966 -0.1282
0.2443 -0.3932
0.3615 0.6315
0.3428 0.2522

S' =

4.2055 0.0000
0.0000 2.4155

R'^{T}_{EV} =

0.4652 -0.6738
0.6406 0.6401
0.5622 -0.2760
0.2391 0.2450

Basically, to compute the similarity between the sentences the ranking matrix of R'^{T}_{EV} is used as input for the cosine distance equation. The cosine distance is a very popular way to measure the similarity and to compute the distance between any two sentences. Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product as:

$$\text{Similarity} = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} \quad \ldots(1)$$

To calculate cosine similarities for the example, the R'^{T}_{EV} matrix was used to calculate cosine similarities for each sentence as follows:

$$sim(Si, Sj) = (Si \bullet Sj) / (|Si|\,|Sj|)$$

In our example, the similarity for S1 is calculated as:

$$sim(S1, S2) = (S1 \bullet S2) / (|S1|\,|S2|)$$
$$sim(S1, S3) = (S1 \bullet S3) / (|S1|\,|S3|)$$
$$sim(S1, S4) = (S1 \bullet S4) / (|S1|\,|S4|)$$

$$sim(S1,S2) = \frac{\left((0.4652 * 0.6406) + (-0.6738 * 0.6401)\right)}{\sqrt{\left((0.4652)^2 + (-0.6738)^2\right)} * \sqrt{\left((0.6406)^2 + (0.6401)^2\right)}} = -0.1797$$

$$sim(S1,S3) = \frac{\left((0.4652 * 0.5622) + (-0.6738 * -0.2760)\right)}{\sqrt{\left((0.4652)^2 + (-0.6738)^2\right)} * \sqrt{\left((0.5622)^2 + (-0.2760)^2\right)}} = 0.8727$$

$$sim(S1,S4) = \frac{\left((0.4652 * 0.2391) + (-0.6738 * 0.2450)\right)}{\sqrt{\left((0.4652)^2 + (-0.6738)^2\right)} * \sqrt{\left((0.2391)^2 + (0.2450)^2\right)}} = -0.1921$$

S3 returns the highest value; pair S1 with S3. The same method then is used to compute the similarity for S2, S3, and S3, S4. Consequently, similar sentences (cosine distance > threshold) are placed together to create a new sentence cluster. Then, a new matrix is created from this cluster and from the rest of the sentences. After applying SVD, all sentences are compared in a pairwise manner. This process is repeated until the distance of the similarity between the document sentences is larger than the previously indicated threshold.

IV. Proposed System

Previously developed systems for identifying news story boundaries depend on the dictation form of words. In contrast, the proposed framework uses the pronounced form. Table III shows some examples of the differences between the dictation and pronounced forms for some popular nouns.

TABLE III. EXAMPLES OF THE DIFFERENCES BETWEEN THE DICTATION AND PRONOUNCED FORMS OF SOME POPULAR NOUNS

Dictation form	Pronounced form with syllables
Abdul Rahman Abdulrahman Abdurrahman Abdulrrahman	ab/dur/rah/ma/n abdurrahman
Yassin,Yasin Yassen, Yasen Yasain, Yassain	Yas/si/n Yassin
Mohammed Mohamed	Moham/ma/d

Mohammad Mohamad Mohamat	Mohammad
Noor, Nour Nur, Nor	No/or Noor

The pronounced forms are more difficult in writing than in reading, as can be seen in the following examples:

1. Most of the sun letters or solar letters (t, v, d, r, z, s, l, and n) can be written with or without a duplicate letter, such as "s" in Yasain or "ss" in Yassain (both are correct). Duplicate consonants in popular nouns are considered to be a common diacritic, with the first being a consonant and the second a vowel [18], [19].

2. Dummy letters have no relation to neighboring letters and no correspondence to pronunciation; in other words, they are empty letters that have no sound (e.g., /h/ in Sarah, Fatimah, John, and Johnny) [8], [9].

3. Auxiliary letters with another letter constitute a diphthong (i.e., two letters combined to represent a single phoneme). These may be further categorized as a standard single-letter representation that uses another letter, as with "oo, ou, u, o in noor, nour, nur, and nor." These are irregular in dictation form. Table IV shows some examples of diphthongs and other ambiguous sounds [8], [9].

TABLE IV. THE DIFFICULTIES IN WRITING POPULAR NOUNS

Combination sound	Example
ai, ay, ei, y	Maitham, Maytham, Meitham, Mytham
oo, ou, o, u	Noor, Nour, Nor, Nur Fong, Foong Choy, Chooy
dh,z	Nadhem, Nazem
s, z	Asman, Azman
ee, ei	Swee, Swei
(ss,s), (dd, d), (mm, m), (rr, r), (tt, t), (vv,v), (zz,z), (ll, l), (nn, n)	Yassain, Yasain Aladdin, Aladin Mohammed, Mohamed Abdurrahman, Abdulrahman Abdultawab, Abduttwab Razzaq, Razaq, Razzak, Razak Abudllah, Abdulah Alnoor, Annoor

Avoiding the problem of writing popular nouns in different ways and thus reducing their impact on story boundary identification involves writing them in generalized and unified ways. This process requires use of an edit distance algorithm (Figure 2). This algorithm controls weights for the characters added and deleted and for the sun letters and dummy letters that are written but not pronounced in popular nouns.

The new system proposed herein proceeds in six stages:

Stage 1: Decode the spoken broadcast news to text using the sphinx 3 ASR system.
Stage 2: Use the maximum a posteriori (MAP) and maximum likelihood linear regression (MLLR) algorithms to improve the ASR acoustic model.
Stage 3: Apply the part of speech tagger (POS) to tag each word with its corresponding part of speech.
Stage 4: Use the generalized noun algorithm to unify the popular nouns (see section V).
Stage 5: Apply preparation processing (see section VI).
Stage 6: Identify the story boundaries using the LSA algorithm (see section III).

V. GENERALIZED NOUN ALGORITHM

The noun unification approach depends on phonetics (i.e., on the pronunciation of popular nouns rather than on the written form). The pronounced form of a word is based on the principle that "only the pronounced sounds are written down, even if they have no corresponding letters in dictation form. Also, what is not pronounced is left unprinted, even if it has a corresponding letter in dictation form"[1][8], [9]. Accordingly, some letters are either inserted or deleted in the pronounced form. There are many reasons for un-standard popular nouns, including the following:

1. The vowel combination makes it more difficult to find one form for the same noun.
2. Sun letters may or may not be duplicated.

[1] Alabbas, pp 5

```
Procedure LevenshteinDistance(S, T)
{
        Str1 ←Char( S) // Split S to array of characters
        Str2 ←Char( T) // Split T to array of characters
        m ←ArrayLen( Str1) // m=length of Str1 array
        n←ArrayLen( Str2) // n=length of Str2 array
        D[m,n] ← 0 // set initial values to Distance matrix D

         For (i←1 to m) // iterates until all char are validated

                D[i,0] ← i

         For (j←1 to n) // iterates until all words are validated

                D[0,j] ← j

        For (i←1 to m)
                For (i←1 to m)
                {
                 if Str1[i] = Str1[j] then
                        D[i, j] := D[i-1, j-1] // no operation required
                 else
                        {
                        if ( Str1[i] and Str1[j] == vowel) then weight=0.3; // substitution vowel letter
                        else // sound that have some relation like (“z/s”, “d/t”)
                         if ( relation(Str1[i], Str1[j] )==true) then weight=0.5;
                        else
                         weight=1;
                         D[i, j] := minimum
                         (
                 D[i-1, j] + 1, // a deletion
                 D[i, j-1] + 1, // an insertion
                 D[i-1, j-1] +weight // a substitution
                 )
                        }
                 }

}
```

Fig. 2. Edit distance algorithm

3. No two phones are exactly identical; within the same language, people pronounce things differently, and between different languages, no two sounds are ever exactly identical.
4. The distinction between vowels and consonants is not always clear cut, and there is a fuzzy boundary region between them in both human pronunciation and automatic recognition.

5. Some sounds are silent (dummy sounds) and are written in dictation form but are not pronounced [3].
6. Some foreign letters can be pronounced and written down using different letters. For instance, consider “Nazem” and “Nadhem:” The letters “z” and “dh” are used to represent the same name.

Converting popular nouns in a document to the general form using the generalized noun algorithm proceeds as follows:

```
Procedure Generalized Noun (Doc)
{
W ← Split (Doc , “ ”) // splits the text on every space
For (i←0 to N) // iterates until all words are validated
{ // check is W[i] noun or not using Part-of-Speech (POS) Tagger
If POS(W[i])= Noun then
        {
                //Find vowel combination and generalized it
            Loop
                Case (Ch← W[i]) :
                        // Group A include (vowel+h letter)
                Group A: replace Ch with μ;
                 // Group B include [ei] sound
                 Group B: replace Ch with β;
                 // Group C include (sun letteers)
                 Group C: replace Ch with λ;
                 // Group D include [ai] sound
                 Group D: replace Ch with Ω;

                                ...
        }  // End case }  // End Loop }  // End if
        } }
```

Poplar Nouns (PN)	Unified (PN)	Levenshtein Distance (LD)		Remark
		Before Unified	After UniFied	
Zainab Zaynab Zeyab	Zβab	LD(Zainab,Zaynab)=1 LD(Zainab,Zeynab)=2 LD(Zaynab,Zeynab)=1	LD(Zβab, Zβab)=0	β refers to [ei] sound group {ai,ay,ey,ei,ea}
Razzaq Razzak Razak Razaq	Raλaq Raλak	LD(Razzaq,Razzak)=1 LD(Razzaq,Razak)=2 LD(Razzaq,Razaq)=1 LD(Razzak,Razak)=1 LD(Razzak,Razaq)=2 LD(Razak,Razaq)=1	LD(Raλaq, Raλaq)=0 LD(Raλak, Raλak)=0 LD(Raλaq, Raλak)=0.5	λ refers to duplicated letters group "sun letters" {t, v, d, b, r, z, s, l and n}
Mohammed Mohamed Mohammad Mohamad Mohamat Mohammet	Mohλad Mohλed Mohλat Mohλet	LD(Mohammed,Mohamed)=1 LD(Mohammed,Mohammad)=1 LD(Mohammed,Mohamad)=2 LD(Mohammed,Mohamat)=3 LD(Mohammed,Mohammet)=1 LD(Mohamed,Mohammad)=2 LD(Mohamed,Mohamad)=1 LD(Mohamed,Mohamat)=2 LD(Mohamed,Mohammet)=2 LD(Mohammad,Mohamad)=1 LD(Mohammad,Mohamat)=2 LD(Mohammad,Mohammat)=1 LD(Mohamad,Mohamat)=1 LD(Mohamad,Mohammet)=3 LD(Mohamat,Mohammet)=3	LD(Mohλad, Mohλad)=0 LD(Mohλad, Mohλed)=0.5 LD Mohλad, Mohλat)=0.5 LD(Mohλad, Mohλet)=0.8 LD(Mohλed, Mohλed)=0 LD(Mohλed, Mohλat)=0.8 LD(Mohλed, Mohλet)=0.5 LD(Mohλat, Mohλat)=0 LD(Mohλat, Mohλet)=0.8 LD(Mohλet, Mohλet)=0	λ refers to duplicated letters group "sun letters" {t, v, d, b, r, z, s, l and n}
Johnny Johnnie Jonny Jonnie	JμλΩ	LD(Johnny,Johnnie)=2 LD(Johnny,Jonny)=1 LD(Johnny,Jonnie)=3 LD(Johnnie,Jonny)=3 LD(Johnnie,Jonnie)=1 LD(Jonny,Jonnie)=2	LD(JμλΩ, JμλΩ)=0	Ω refers to [ai] sound group {ie,y,uy}

TABLE V. MEASUREMENT OF THE SIMILARITY BETWEEN TWO STRINGS USING LEVENSHTEIN DISTANCE

Table V illustrates the measurement of the similarity between two strings using Levenshtein distance for several examples. The number of transformations (deletions, insertions, or substitutions) required to transform one string into another were measured before and after the generalized noun algorithm was applied.

VI. PREPROCESSING STAGE

The preprocessing module performs tagging, removal of stopping words, stemming, feature selection, and TSM creation. During the tagging process, each word is tagged with its corresponding POS. For example, the sentence "ali pergi ke sekolah" (Ali goes to school) is tagged as "ali/noun pergi/verb ke/preposition sekolah/noun." In this study we used the Qtag POS tagger. The next step is to remove stopping words, which removes all of the frequent and common words that do not carry important information. This step reduces the size of the spoken document. Such words include auxiliary verbs and prepositions (e.g., adalah/(is, are), akan/will, was/ialah, ke/to, pada/at). The removal of such words helps to improve the quality of the story boundary identification results by retaining only the words that contain significant information. This step can be performed using the stopping word list, which includes 1312 common Malay stopping words.

Stemming refers to reducing morphological variants of words to their stem, base, or root form, and it is used to improve the effectiveness of information retrieval (IR). The effect of stemming depends on the nature of the language vocabulary, and in some cases stemming may degrade retrieval performance [20]. Thus, a stemmer can improve the effectiveness of IR for some text corpora more than others [16], [17], [20]. In the system proposed here, an affixation stemmer for the Malay dataset was used. The words permakanan (diet) and makanannya (his/her food), for example, contain the base word "makan," and the common stem of the various forms of the word was weighted using the tf-idf (term frequency-inverse document frequency) weighting approach in the term-by-document matrix (TDM). The use of the stemming algorithm can increase retrieval performance by reducing morphological

variants of words and the time required for processing; at the same time, use of the roots of the words increases the similarity probability between the words in the clustering module.

After stemming is completed, feature selection is performed. One of the major challenges facing artificial intelligence applications is how to reduce the number of high dimensional data spaces. Dimensionality reduction is the process of reducing the number of random variables (words here) under consideration (for instance, retaining the significant words or the high frequency words) [16]. The efficiency of the relevant algorithms can be improved by decreasing the dimensionality of the size of the effective vocabulary and data spaces. In such cases, feature selection can be applied. Feature selection chooses an effective subset from a huge set of features. In this study, we used the open source library "weka" to select the useful features, and only the selected keywords (words) were used in the subsequent building of the TSM.

In the TSM, each row defines the terms contained in a sentence. Each cell entry contains the frequencies of occurrence of a term in a sentence. This TSM can be used to calculate the similarity between terms using story boundary identification methods. To illustrate, suppose we have the following set of five sentences:

S1 = w1 w2 w3 w4 w5 w6
S2 = w7 w2 w3 w4 w5 w6
S3 = w6 w8 w4 w5
S4 = w1 w9 w10 w4 w6
S5 = w10 w2 w2 w4 w11 w5

A data set can be represented by the TSM using the frequency weight matrix shown in Table VI.

TABLE VI. TERM-BY-SENTENCE MATRIX (TSM)

	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11
S1	1	1	1	1	1	1	0	0	0	0	0
S2	0	1	1	1	1	1	1	0	0	0	0
S3	0	0	0	1	1	1	0	1	0	0	0
S4	1	0	0	1	0	1	0	0	1	1	0
S5	0	2	0	1	1	0	0	0	0	1	1

VII. EXPERIMENTS AND RESULTS

To demonstrate the performance of the proposed algorithms, a transcript produced manually from spoken broadcast news was used [21] to identify the story boundaries. The databases used for this are called the mass-news corpus, and they consist of Malay broadcast news documents that were collected at Universiti Sains Malaysia as the output of the Malay ASR system [21]. The news stories used for this evaluation were collected in March 2011. The data set includes ~25 hours of transcribed speech. The ASR system was trained using a ~15 hour portion of the database, and the test sets included the remaining ~10 hours. The broadcast news stories included multiple speakers and recording in noisy environments. None of the test sets overlapped with the ASR training set. Table VII shows the Malay data source details that were used in this study.

TABLE VII. DATA SOURCE DETAILS

Size of language model	150 MB
Size of dictionary	1.74 MB
Number of news shows	18
Number of news stories in all news shows	379
Number of sentences in news database	4698
Number of words in the news	81116
Number of popular nouns in the news	39698 (49%)
Word error rate before adaptation	34.5%
Word error rate after adaptation	33.9%
Story length	Around 1 to 167
The rate of the audio signal extract	10 ms

Errors resulting from the process of story boundary identification were measured using the F-score, precision, and accuracy. To evaluate[2] the effectiveness of the story boundary identification module, we tested it using Malay spoken documents that contained ~380 stories in different domains (e.g., politics, economics, sports, local news, and international news). We evaluated two corpora. The first corpus was a gold standard file (GSF) corpus that represents the manual transcription of the Malay broadcast news. The second corpus was the ASR result (Hypothesis Result (HR)) for the Malay broadcast news. The GSF corpus was segmented into stories by human experts. In this experiment different k-dimensional clustering spaces were built where k ϵ [32, 50, 80, 100, 125, 150, 200]. This paper reports only the best results.

Table VIII shows the results of the story boundary identification for LSA with and without the noun unification process.

[2]The tools that used in this study
Java, Python, Apache Lucene package (java package) was used for compute F-measure and the other Measurement, Sphinx 3 as ASR system, WEKA package for feature selection, SPSS for Wilcoxon signed-ranks test, Qtag tool for part of speech tagger, an affixation stemmer tool, Jama package for Matrix computations.

TABLE VIII. STORY BOUNDARY IDENTIFICATION MODULE PERFORMANCE

	LSA			
	Without		With	
	GSF	HR	GSF	HR
Precision	0.759	0.680	0.895	0.814
Recall	0.617	0.559	0.914	0.769
F-Measure	0.681	0.613	0.904	0.791

The results of the story boundary identification algorithms were evaluated statistically using the Wilcoxon signed-ranks test. By applying the same statistical significance test, the results of the proposed algorithm were compared statistically with the results of the baseline algorithm (i.e., LSA without the noun unification process). The proposed system using the popular noun unification algorithm achieved an F-measure of 0.791, whereas the value was 0.613 for the baseline system when tested on the same set of Malay broadcast news stories.

VIII. CONCLUSIONS

Identifying broadcast news story boundaries plays an important role in many natural language processing applications, such as topic identification and story classification. The proposed system uses the pronounced forms to identify story boundaries based on popular noun unification. LSA is commonly used in clustering methods because of its excellent performance and because it is based on deep semantic rather than shallow principles. In this study, the LSA algorithm with popular noun unification achieved a better result than the general LSA algorithm in identifying news story boundaries for the same test set. The LSA algorithm with popular noun unification achieved an overall F-measure of 0.791 versus 0.613 for the general LSA algorithm when identifying news story boundaries for the same test set.

We predict that further work by adding ASR confidence measure to distinguish between correct and incorrect words in ASR result before any processing, e.g. story boundaries identification. In future work, we will apply this algorithm for English language.

ACKNOWLEDGMENT

ZAK owes her deepest gratitude to USM for its support of her PhD research. She also would like to extend thanks to Basra University for their helpful support.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, Modern information retrieval vol. 463: ACM press New York, 1999.
[2] C. Chelba, et al., "Retrieval and Browsing of Spoken Content," IEEE signal Processing Magazine, vol. 25, pp. 39-49, 2008.
[3] M. Ostendorf, et al., "Speech Segmentation and its Impact on Spoken Document Processing," 2007.
[4] M. Abbas, et al., "Evaluation of Topic Identification Methods on Arabic Corpora," Journal of Digital Information Management, vol. 9, pp. 185-192, October, 2011.
[5] D. Li, et al., "Initial Experiments on Automatic Story Segmentation in Chinese Spoken Documents Using Lexical Cohesion of Extracted Named Entities" in ISCSLP, 2006, pp. 693-703.
[6] M.-m. LU, et al., "Multi-Modal Feature Integration for Story Boundary Detection in Broadcast News," IEEE, pp. 420-425, 2010.
[7] W. Hsu, et al., "Discovery and fusion of salient multimodal features toward news story segmentation," in Proceedings of SPIE, 2004, pp. 244-258.
[8] Wikipedia. (2013). Silent letter - http://en.wikipedia.org/wiki/Silent_letter.
[9] A. a. Galina. (2007-2013). English Vowel Sounds- http://usefulenglish.ru/phonetics/english-vowel-sounds.
[10] M. A. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic passages," Computational linguistics, vol. 23, pp. 33-64, 1997.
[11] N. Stokes, et al., "SeLeCT: a lexical cohesion based news story segmentation system," AI COMMUNICATIONS, vol. 17, pp. 3-12, 2004.
[12] A. Rosenberg and J. Hirschberg, "Story segmentation of broadcast news in English, Mandarin and Arabic," in Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, Stroudsburg, PA, USA, 2006.
[13] L. Xie, "Discovering salient prosodic cues and their interactions for automatic story segmentation in Mandarin broadcast news," Multimedia Systems, vol. 14, pp. 237-253, 2008.
[14] S. Banerjee and A. I. Rudnicky, "A TextTiling based approach to topic boundary detection in meetings," 2006.
[15] C. H. Wu and C. H. Hsieh, "Story segmentation and topic classification of broadcast news via a topic-based segmental model and a genetic algorithm," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 17, pp. 1612-1623, 2009.
[16] J. Geiß, "Latent semantic sentence clustering for multi-document summarization," Ph.D, University of Cambridge, 2011.
[17] Z. A. Khalaf and T. T. Ping, "Unsupervised Identification of Story Boundaries in Malay Spoken Broadcast News," Journal Of Emerging Technologies In Web Intelligence, vol. 5, pp. 28-34, 2013.
[18] Z. A. Khalaf, et al., "BASRAH: Arabic Verses Meters Identification System," in IALP, Penang-Malaysia, 2011, pp. 41-44.
[19] M. Alabbas, et al., "BASRAH: an automatic system to identify the meter of Arabic poetry," Natural Language Engineering-Cambridge University Press 2012, pp. 1-19, 2012.
[20] W. B. Frakes and R. Baeza-Yates, "CHAPTER 8: STEMMING ALGORITHMS Information Retrieval: Data Structures & Algorithms," ed. Englewood Cliffs, NJ: Prentice Hall, 1992, pp. 131-160.
[21] T. Tien-Ping, et al., "Mass: A Malay Language LVCSR Corpus Resource," Cocosda'09. Urumqi, China, 2009.