International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 11, No. 1, pp. 37 – 55, 2014

NOVEL NOUN PRONUNCIATION UNIFICATION APPROACH TO IMPROVE STORY BOUNDARY IDENTIFICATION IN THE TRANSCRIPTION OF MALAY NEWS BROADCASTS

ZAINAB A. KHALAF*

School of Computer Sciences, Universiti Sains Malaysia (USM),
11800 Penang, Malaysia
zainab_ali2004@yahoo.com

TAN TIEN PING

School of Computer Sciences, Universiti Sains Malaysia (USM),
11800 Penang, Malaysia

This paper introduces a novel method for improving story boundary identification (SBI) for a speech recognition outcome. We explore an SBI improvement method using latent semantic analysis (LSA) with noun unification. The proposed system uses the phonetic forms of words (pronunciation forms) to identify story boundaries based on noun unification and edit distance to estimate the cost of edit operations for nouns and to compare this cost with a predetermined threshold generated by a training dataset. SBI commonly uses latent semantic analysis for its excellent performance and because it is based on deep semantics rather than shallow principles. In this study, the LSA algorithm with and without unification was used to identify the boundaries of Malay spoken broadcast news stories. The LSA algorithm with the noun unification approach resulted in less errors and better performance than the LSA algorithm without noun unification. The preliminary results of the current work for SBI using LSA with noun unification are encouraging compared with the common LSA with the general approach for selecting bag-of-words.

Keywords: automatic speech recognition; latent semantic analysis; word pronunciation; story boundaries identification.

1. Introduction

Identifying word errors generated by automatic speech recognition (ASR) is one of the major challenges confronting the task of natural language processing (NLP). To ensure that the handling and management of large news video text are performed efficiently, spoken broadcast news must be segmented into units of stories. Separating the news into units can enable us to detect the part where one story ends and another begins in a spoken document. Being conscious of the fact that story boundary detection has undergone various phases of research, this process is deemed impractical because of its less-than-satisfactory performance. Because the problem is believed to be difficult and too general,

[*] *Department of Computer Science, College of Science, University of Basra, Iraq*

no specific single feature is considered adequate to handle the story boundary detection process for the bulk of broadcast news.

Another shortcoming of NLP is the identification of topic boundaries. To determine whether the handling and management of large news video text are conducted effectively, spoken documents must be segmented into units (or topics). Separating a document into topics enables the NLP system to offer users the topics in which they are interested immediately in a spoken document [Diao et al. (2010); Senay et al. (2011)]. Separating documents into units also enables us to detect the place where one story ends and another begins in a stream of a medium, such as text, video, or speech. According to [Jain (2010)], "clustering is a more difficult and challenging problem than classification." This process is extremely difficult in spoken documents, and no specific single feature is considered adequate to handle the story boundary detection process for the bulk of spoken documents. For example, a user searching for news related to "the war in Syria" would prefer to receive a short story about the war instead of every story that contains details about the war [Mengle and Palmer (2004)].

In recent years, some effort has been exerted to developing methods to segment broadcast news into stories to enhance the performance of the NLP task [LU et al. (2010); Ostendorf et al. (2007)]. Although NLP has been the focus of many studies, more work is needed to find better ways to address the shortfalls of NLP performance caused by problems with ASR.

In this paper, we present a new approach called noun unification. Our approach achieves significant improvements over the previous approaches, such as the common LSA approach. The main idea in the proposed noun unification system is that it depends on word pronunciations rather than written forms for the words and tries to avoid the word error rate in ASR transcription. The remainder of this paper is organized as follows: Section 2 presents related work. Section 3 discusses the noun problem, and Section 4 presents the proposed system. The speech corpora used in the current study is explained in Section 5. Finally, we report the experimental results in Section 6 and conclude in Section 7.

2. Related Work

According to previous studies, SBI contains two main subfields: linear subfields and hierarchical subfields. Linear SBI involves the sequential analysis of different topical changes within texts, whereas hierarchical SBI considers a more sensitive approach to subtopic structures.

Hearst (1997) pioneered the application of a linear SBI algorithm in text analysis. The algorithm called TextTiling divides texts in linear time based on the similarity of two text sections. Cosine similarity is the calculation used to determine the degree of similarity [Hearst (1997)].

Two vectors operate within this calculation. These vectors aim to calculate the number of matching terms based on the term frequency in each section. Galley et al. published LcSeg, a TextTiling-based algorithm, in 2003 [Galley (2003)]. LcSeg differs from TextTiling because it utilizes TF-IDF term weights, which have been determined to improve the results of SBI. C99, an algorithm introduced by Choi (2000), uses clustering processes and matrix-based rankings to determine the degree of similarity between texts. C99 is also similar to its predecessors in that it uses words [Choi (2000)]. U00 is an example of a Dynamic Programming (DP) approach by Utiyama and Isahara (2001). DP aims to find paths within a graph at a minimal cost. DP is applied with SBI to demonstrate the similarities between each segment, such as sentence boundary placements. DP ensures low cost by penalizing the use of common vocabulary within sections of text [Utiyama (2001)].

Many researchers have tried different SBIs with various methods of DP, including Fragkou et al. (2004), Sun et al. (2008), Du et al. (2010), Mulbregt et al. (1998), Blei and Moreno (2001), Malioutov and Barzilay (2006), and Kazantseva and S. (2011). Fragkou et al. (2004) suggested an equation that uses DP to reduce the segmentation cost when segmenting text [Fragkou (2004)]. However, sub-optimal segmentation, which can be caused by smooth topic transitions known as weak boundaries, must be addressed. Sun et al. (2008) developed an equation that modifies Fragkou's attempt [Sun (2008)]. Additionally, Misra et al. (2009) built on the DP algorithm proposed by Utiyama and Isahara called U00 (2001) through the use of topic models [Misra (2009)]. While other approaches calculate the likelihood of co-occurring words, Utiyama and Isahara instead considered the likelihood of topic co-occurrences. Different methods for SBI have been used, including the Hidden Markov Model (HMM), as initially proposed by Mulbregt et al. (1998) [Mulbregt (1998)]. In 2001, Blei and Moreno demonstrated an Aspect HMM (AHMM), a process that merges an HMM with an aspect model [Blei (2001)].

In modern attempts, topic models have been used to analyze text similarity. Using an approach based on topic models, similarities can be analyzed relative to the relationships of related words (a hierarchical approach) in addition to the more standard exact word repetitions. Yaari (1997) first suggested a hierarchical algorithm that uses cosine similarity and agglomerative clustering as the basis of its approach [Yaari (1997)]. Eisenstein (2009) suggested a hierarchical Bayesian algorithm with a foundation in LDA [Eisenstein (2009)]. Latent semantic analysis (LSA) was used by Choi et al. (2001) with the C99 (Choi 2000) algorithm, which enabled the examination of sentence similarities by considering a sentence to be the sum of its LSA feature vectors. Choi et al. found a strong advantage of LSA-based metrics over the cosine measure used in the original C99 algorithm [Choi (2001)].

Because the TextTiling (Hearst 1997) approach is efficient and uncomplicated, it was recently applied to the segmentation of speech documents, such as broadcast news segmentation [Xie and J. Zeng (2008)] and meetings [Banerjee and Rudnicky (2006)]. Later, principal component analysis (PCA) was used for story segmentation founded on

LSA; PCA considers the differing meanings of words across various contexts (Jerome and Bellegarda 2005) but is not necessarily superior to the usual lexical approach, though it improves the separability among topics [Jerome and Bellegarda (2005)].

Word cohesion was introduced by Stokes et al. by examining lexical chaining in terms of story segmentation. They developed the SeLeCT system, which couples related words into lexical chains and typifies segment boundaries using a large number of chain start and end points. A standard lexical chaining approach counts the number of chain starts and ends at inter-sentence positions [Stokes et al. (2004)].

In 2007, Chan et al. proposed the log-normal distribution to express the statistical behavior of lexical chains for a more efficient means of story segmentation [Chan and L. Xie (2007)]. In addition, Malioutov and Barzilay (2006), Heinonen (1998), and Fragkou (2004) suggested different lexical-cohesion-based segmentation approaches aimed at discovering the best segmentation distribution using whichever criteria are considered optimal. In addition, TextTiling and lexical chain analysis were recently used on subword sequences (i.e., character/syllable-grams) of the speech transcription of Chinese broadcast news as subword units, resulting in significantly less speech recognition errors [Xie and J. Zeng (2008)], [Yang (2008)]. A methodology that uses combined concept matching from LSA and the sturdiness of subword units (characters and syllables) was proposed by Yang (2008). Yang and Xie (2008) used the LSA vectors of subword units in conjunction with the assessment of inter-sentence lexical scores from the use of TextTiling-based story segmentation on Chinese ASR transcripts [Yang (2008)]. The use of Laplacian eigenmaps (LE) has been suggested to improve automatic story segmentation in automatic transcriptions of Chinese broadcast news [Xie et al. (2011)]. This approach used LE maps to produce a sentence connective strength matrix and to show that stories have an inherent geometric structure (Xie L. 2012). In our work, however, we focus on linear and hierarchical topic segmentation. In our work, however, we focus on hierarchical topic segmentation.

3. Noun Problem

In any language, numerous factors affect the nouns, especially for proper names.

- Vowels (especially in proper names) are central to understanding the acoustic characteristics of speech. Due to the early appearance of vowels in speech evolution, vowels are considered to be an important milestone in the study of speech evolution. Vowels serve a specific function in language and serve an indispensable role in “word variations” [Khalaf and Ping (2013a); Narang (2011)].
- The majority of sun letters or solar letters (t, v, d, b, r, z, s, l and n) can be written with or without duplicate letters, such as “s” in Yasain or “ss” in Yassain (both are correct). Duplicate consonants in proper names are considered to be a common diacritic, in which the first letter is a consonant and the second letter is a vowel [Alabbas et al. (2012); Khalaf et al. (2011)].
- Some sounds are silent (dummy sounds) and are written in dictation form but are not pronounced, including dummy letters with no relation to neighboring letters and no

correspondence to pronunciation; they are empty letters that have no sound (e.g., /h/ in Sarah, Fatimah, John and Johnny) [Galina (2007-2013); Khalaf and Ping (2013a); Wikipedia (2013)].

- Auxiliary letters with another letter constitutes a diphthong (i.e., two letters combined to represent a single phoneme). These letters may be categorized as a standard single-letter representation that uses another letter, such as oo, ou, u and o in noor, nour, nur and nor, respectively. These letters are irregular in dictation form [Khalaf and Ping (2013a)].

Numerous factors affect the results of ASR. These factors include the accent, emotional status and gender of the speaker, as well as the amount of noise, source of noise and speech type (planned speech or spontaneous speech). According to these variances, variations in word pronunciation occur, particularly relative to a speaker's accent, which serves to degrade the results of ASR. To demonstrate these variations, we note the following reasons:

- No two phonemes are exactly identical; within the same language, people pronounce things differently, and between different languages, no two sounds are identical [Khalaf and Ping (2013a)].
- The distinction between vowels and consonants is not always evident, and a fuzzy boundary region exists between vowels in both human pronunciation and automatic recognition [Khalaf and Ping (2013a)].
- Some foreign letters can be pronounced and written using different letters. For instance, consider "Nazem" and "Nadhem". The letters "z" and "dh" are used to represent the same sound [Khalaf and Ping (2013a)].

An example from a Malay broadcast news (MBN) 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 because 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 [Galina (2007-2013); Khalaf and Ping (2013a); Wikipedia (2013)]. The identification of story boundaries with the problem of pronunciation errors is a complicated task. This task requires human knowledge of the rules for the correct pronunciation of lexical items. To address these problems, we propose a new method to reduce the ASR WER and to improve SBI in spoken documents by noun unification.

To illustrate the problem, we consider an example from the native Malaysian dictionary. The proper name "Anwar" has different spelling (and pronunciation) forms, as shown in Table 1.

Table 1. Variation of the proper name "Anwar"

Word dictation	**Pronunciation**
Annuar	a n u w a
Annuar (2)	a n u w a r
Annur	a n u r
Anuar	a n u w a
Anuar (2)	a n u w a r
Anwar	a n w a
Anwar (2)	a n w a r
Anuwar	a n u w a
Anuwar (2)	a n u w a r
Anwar	a n w a
Anwar (2)	a n w a r
Anuar	a n u w a
Anuar (2)	a n u w a r

Figure 1 shows the finite state automata (FSA) for the spelling variations of the proper name "Anwar," and Figure 2 shows the FSA for the pronunciation variations of the proper name "Anwar."

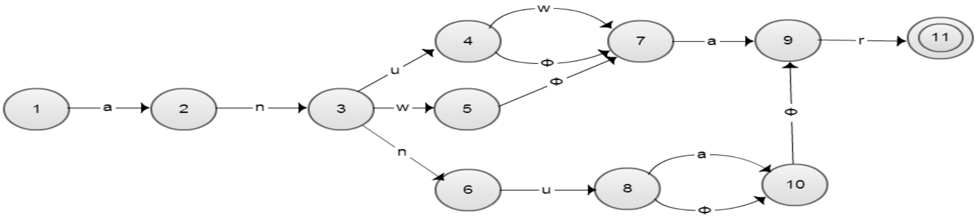


Fig. 1. FSA for the spelling variations of the proper name "Anwar"

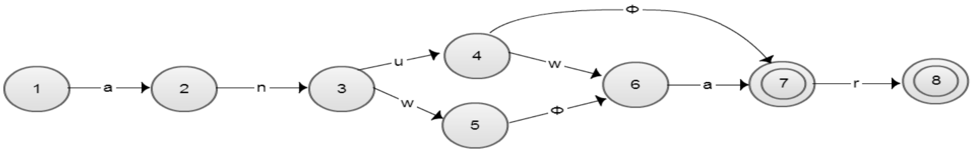


Fig. 2. FSA for the pronunciation variations of the proper name "Anwar"

To avoid the problem of writing nouns in different spelling forms and to reduce their effect on story boundary identification, we propose a new approach using unified and generalized pronunciations for nouns.

4. Proposed system

The proposed SBI system consists of four main phases: sentence segmentation, indexing, noun unification, and term and sentence matching clustering. The transcription is obtained by decoding the broadcast news audio files.

4.1. Phase 1: Sentence Segmentation

Sentence segmentation, the first stage, segments the text in the transcription into sentences before they are grouped into stories. Numerous stories exist in broadcast news. These stories include local news, international news, politics, economy, sports, and other topics.

4.2. Phase 2: Indexing

Indexing is a pre-processing stage for sentences. The indexing stage involves tokenization, stop word removal, stemming, term selection, and term representation.

4.1.1. Tokenization

Tokenization is a technique that is used to break sentences into words. Each word is represented as a term. The tokens can be nouns, adjectives, verbs, or other parts of speech.

4.1.2. Stop word removal

Stop word removal removes words (terms) that are less meaningful relative to others. Stop words are words consisting of conjunctions, interjections, and prepositions, such as “dari,” “dan,” “ke,” and “kerana”. Stop word removal can reduce the word density in the clustering process. This step can be performed using the stop word list, which includes 1312 common Malay stop words.

4.1.3. Stemming

Stemming refers to the reduction of the morphological variants of words to their stem, base, or root form and is used to improve the effectiveness of NLP tasks. For example, “membeli” contains “mem” (a verb) as a prefix, and “kerajaan” consists of “ke” as a prefix and “an” (a noun) as suffix. The effect of stemming depends on the nature of the language vocabulary, and in some cases, stemming may degrade clustering performance. Thus, a stemmer can improve the effectiveness of NLP tasks for some text corpora better than it can for others [Khalaf and Ping (2013a)]. In the proposed system, an affixation stemmer for the Malay dataset was used.

4.1.4. Term selection

One of the major challenges confronting artificial intelligence applications is how to reduce the number of high-dimensional data spaces. Dimensionality reduction is the process of reducing the number of the random variables (in this case, words) under consideration (for instance, retaining the significant words or the high frequency words). Term selection is an important process in natural language processing because it is used to build the bag-of-words model (vector space model) by selecting the meaningful words in the transcription. In this step, we select only nouns and stemming verbs for term representation.

4.1.5. Term representation

In this step, we represent the sentences and terms in the transcription using the vector space model. We use a binary term vector to indicate whether a term is present or absent in a sentence. This model is selected because the transcription is unstructured. Unstructured text is unsuitable for visualization. In this case, the Boolean representation is used to show that all of the selected terms in every sentence are equally important. Most research papers use Term Frequency Inverse Document Frequency (TF-IDF) to determine the importance of a term because not every term is equally important. TF-IDF filters the common terms, but some terms that appear frequently might have important meanings.

4.2. Phase 3: Noun unification

The noun unification approach depends on phonetics (i.e., on the pronunciation of nouns rather than on their written forms). The pronunciation forms of a word are based on the principle that "only the pronounced sounds are written down, even if they have no corresponding letters in dictation form. Additionally, what is not pronounced is left unprinted, even if it has a corresponding letter in dictation form" [Alabbas et al. (2012); Khalaf et al. (2011)]. In this study, we used the ASR pronunciation model to determine the pronunciation of a word.

The main stages in the proposed algorithm for noun unification are as follows:

1. Extraction of nouns using parts of speech (POS).
2. Determination of noun pronunciation using pronunciation dictionary.
3. Determination of the best alignment based on edit distance.
4. Determination of the confusion groups using a confusion matrix.
5. If the edit distance satisfies the threshold value, then unify the nouns.
6. Repeat for all nouns in the word bag.

4.2.1. POS Tagger

The POS module performs tagging. During the tagging process, each word is tagged with its corresponding part of speech class. For example, the sentence "Ali pergi ke sekolah" (Ali goes to school) is tagged as "Ali/noun, pergi/verb, ke/preposition, sekolah/noun."

4.2.2. Pronunciation Dictionary

The pronunciation or lexical model is called the lexicon. The lexicon represents possible human-generated words and their permissible pronunciations, which are organized as a sequence of phonemes that identify a set of words or phrases. Table (2) shows some mapping examples between words and their phoneme sequences. The lack of correspondence between many phonetic sequences and the actual use of words has hindered the investigation of many phonetic sequences in the decoding process.

Table 2. Examples of lexical mappings from words to phonemes.

Word	**Pronunciation**	**Meaning**
berselawat	b @ r s @ l a w a t	bless
ketinggalan	k @ t i NG g a l a n	miss
spontannya	s p o n t a n NJ @	spontaneous

4.2.3. Edit Distance

The purpose of this step is to match the hypothesis word pronunciation in ASR transcription by finding a similar word pronunciation with a minimal edit distance in the pronunciation dictionary. The edit distance is controlled by the weights of the phoneme groups generated from the confusion matrix for the training data. A binary search can be achieved easily because the pronunciation dictionary is alphabetically arranged, thereby accelerating the time required for error detection.

The function ED (Pi, Pj) demonstrates the distance between word pronunciation Pi and word pronunciation Pj, where Pi is any word in the ASR transcription and Pj is a word in the pronunciation dictionary. Similar phonemes, such as “a” and “@,” are often misrecognized compared with other phonemes. Therefore, we adopted a modified edit distance that compensates for the confusability between phonemes [Qin (2013)].

Algorithm 1 Pseudo-code of the proposed edit distance algorithm

```
For each $noun_k$ in old_word_bag
    Assert (group, $noun_k$)
    For each $noun_{k+1}$ in old_word_bag
        N number of phonemes in $noun_k$
        M number of phonemes in $noun_{k+1}$
        $i = 1 ...N, j = 1...M$
        $W(i, j)$ is the confusability between phone i and phone j
        $P(i, j)$ is the probability of misrecognizing phone i and phone j
        $N(i\ \ j), N(j\ \ i)$ are the number of times phone i was recognized as phone j and vice versa
        $N(i), N(j)$ are the number of phone i and phone j in the training data
        $ED(0, 0) = 0; ED(i, 0) = i; ED(0, j) = j;$
        $P(i, j) = P(j, i) = (N(i\ \ j) + N(j\ \ i)) / (N(i) + N(j))$
        if $(i = j)$ then $W(i, j) = 0$ else $W(i, j) = 1 - P(i, j)$
        $ED(i, j) = \min(ED(i - 1, j)+1, ED(i, j - 1) + 1, ED(i - 1, j - 1) + W(i, j))$
        If $ED(i, j) <$ threshold then assert (group, $noun_{k+1}$)
        Else
    Continue
    UN = Unified(group)
    Add(UN, new_word_bag)
    Deassert(group)
    Delete(group, old_word_bag)
Continue
```

4.2.4. Confusion Matrix

In this study, we created a confusion matrix for 36 Malay phonemes for the training data of the manual transcription and training corpus generated by ASR. We studied and identified the relationship between the reference phoneme (actual phoneme in the manual transcription) and the hypothesis phoneme (predicated ASR outcome phoneme) using a statistical matrix.

A phoneme confusion matrix can be created by aligning the ASR outcome hypothesis phonemes with the corresponding reference phoneme sequence using the alignment of a speech recognition system. Thus, we calculated an alignment between the hypothesized phoneme and actual phoneme. To provide insight into the actual phoneme substitutions that occur in speech recognition results, the confusion matrix was used. Many approaches can be used to perform the alignment (e.g., time alignment or Levenshtein distance). We adopted the Levenshtein distance in this study.

Vowel phonemes have a high confusion rate. However, the analysis results show that /@/ and /a/ have a top rate of confounded pronunciation equal to 1556. The top rate of confounded pronunciation in consonant phonemes, i.e., between /t/ and /d/, is 839.

To calculate the probability of misrecognizing the phoneme, suppose P(i, j) is the probability of misrecognizing phoneme i and phoneme j. We estimate P(i, j) from the training data using the following equation

$$P(i, j) = P(j, i) = (N(i \rightarrow j) + N(j \rightarrow i))/(N(i) + N(j)), \tag{1}$$

where N(i) and N(j) are the numbers of phoneme i and phoneme j, respectively, in the training data and N(i j) and N(j i) are the number of times phoneme i was recognized as phoneme j, and vice versa. In this thesis, P(i, j) is estimated from the confusion matrix for the training data. For example, the probability of misrecognizing phoneme a and phoneme @ is expressed as follows:

$$P(a, @) = P(@, a) = (N(a \rightarrow @) + N(@ \rightarrow a))/(N(a) + N(@))$$

$$P(a, @) = P(@, a) = (1556 + 623)/(123021 + 50764) = 0.0125$$

4.2.5. Noun Unification

The main aim of the noun unification approach is pronunciation generalization. The pronunciation of phonemes appears inside the group of nouns collected in the dynamic group by the assert command (Algorithm 1) and are generalized by using the phonemes with the highest frequencies.

Algorithm 2 Pseudo-code of the noun unification algorithm

```
Doc = ASR transcription
Get_best_alignment (group)
Freq = Frequency (phone)
Unified_noun = get_high_freq(freq)
Return Unified_noun
```

4.2.6. *Example*

Let G_N = {“a n w a”,“ a n w a r”,“ a n a u r”,“ a n u w a”, “a n u w a r”} be a group of noun pronunciations (NP) generated by Algorithm 1. The largest number of NP phonemes is six.

To find the confusion phoneme sequence in G_N, all nouns in G_N were aligned using the multi-alignment edit distance to select the best alignment. Each sequence of noun phonemes (phoneme-to-phoneme alignment) was checked to find the confusion phoneme(s), and the phoneme with the highest frequency was selected. According to G_N, the a and n phonemes occur five times, whereas the confusion phonemes u and ф occur three and two times, respectively. Thus, the highest frequency (i.e., three times) was chosen for u, and so on. The unified pronunciation of group G_N is “ a n u w a r.” Table 3 shows the unified approach.

Table 3. Unified approach.

	P_1	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**
NP_1	a	n	u	w	a	
NP_2	a	n	u	w	a	r
NP_3	a	n	u		a	r
NP_4	a	n		w	a	
NP_5	a	n		w	a	r
Phones	a	n	u/ф	w/ф	a	r/ф
High Freq.	a/5	n/5	u/3	w/4	a/5	r/3
Unified	a	n	u	w	a	r

4.3. *Phase 4: Clustering by using Latent Semantic Analysis Algorithm*

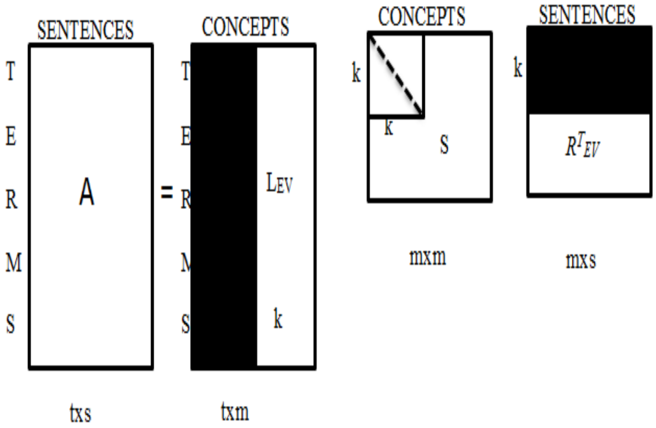
The clustering of sentences can be used to locate repeated information, which is a process that is performed by grouping similar sentences. Some studies have examined a number of different methods that can be used to identify similar sentences. Some methods employ shallow techniques to detect similarities in sentences (e.g., word or n-gram overlap), whereas other methods employ a syntactic or semantic analysis by applying the LSA technique to estimate the similarities between word matching and semantic structures. Accordingly, the problem of synonymy is prevented [Geiß (2011); Khalaf and Ping (2013b)].

A spoken document is typically scanned and split into sentences throughout the preparation process and TSMs are subsequently created. An advantage of using LSA is that it reduces dimensionality, which results in rapid clustering. When a matrix is prepared, it is subjected to singular value decomposition (SVD), as shown in figure 3 [Geiß (2011); Khalaf and Ping (2013b)]. The SVD formula is expressed as follows:

$$A = L_{EV} \times S \times R_{EV}^{T}$$

Any rectangular matrix A (TSM matrix here) with order t×s is decomposed to three matrices (L_{EV}, S, R_{EV}^{T}). The matrix L_{EV} consists of the left eigenvectors of A and either describes the relationships between the terms (rows) and the sentences (columns) or refers to the term-to-concept similarity matrix, which is created using the equation $L_{EV} = A^{T}A$. The matrix S is a m×m diagonal matrix with the entries sorted in decreasing order. The entries of the S matrix comprise the singular values (eigenvalue) and describe the relative strengths of each concept. R_{EV}^{T} is a matrix that is defined by the equation $R_{EV}^{T} = A\ A^{T}$; it includes the left eigenvectors of A and is associated with the sentence-to-concept similarity matrix [Geiß (2011); Khalaf and Ping (2013b)].

.Fig. 3. SVD



The functionality of LSA will be explained by an example from the term similarity calculation. Consider table 2.4, which displays 4 sentences from technical reports (Ngo et al. 1990).

Table 4. Example with 4 sentences

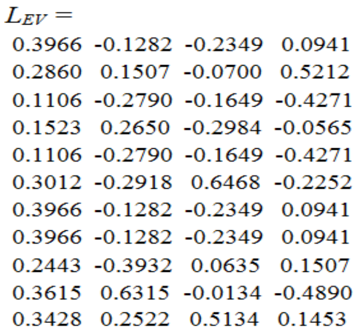
S1	Shipment of gold damaged in a fire
S2	Delivery of silver arrived in a silver truck
S3	Shipment of gold arrived in a truck
S4	Gold silver truck

1. The TSM matrix (A-table 5) is constructed as follows:

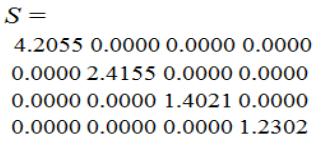
Table 5. TSM.

	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

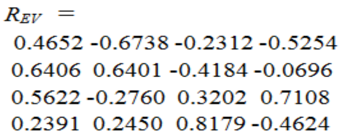
2. SVD is used to decompose the A matrix into three matrices.



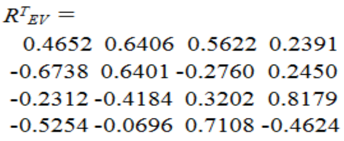
$$L_{EV} = \begin{matrix} 0.3966 & -0.1282 & -0.2349 & 0.0941 \\ 0.2860 & 0.1507 & -0.0700 & 0.5212 \\ 0.1106 & -0.2790 & -0.1649 & -0.4271 \\ 0.1523 & 0.2650 & -0.2984 & -0.0565 \\ 0.1106 & -0.2790 & -0.1649 & -0.4271 \\ 0.3012 & -0.2918 & 0.6468 & -0.2252 \\ 0.3966 & -0.1282 & -0.2349 & 0.0941 \\ 0.3966 & -0.1282 & -0.2349 & 0.0941 \\ 0.2443 & -0.3932 & 0.0635 & 0.1507 \\ 0.3615 & 0.6315 & -0.0134 & -0.4890 \\ 0.3428 & 0.2522 & 0.5134 & 0.1453 \end{matrix}$$



$$S = \begin{matrix} 4.2055 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 2.4155 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.4021 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.2302 \end{matrix}$$

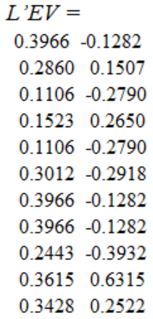


$$R_{EV} = \begin{matrix} 0.4652 & -0.6738 & -0.2312 & -0.5254 \\ 0.6406 & 0.6401 & -0.4184 & -0.0696 \\ 0.5622 & -0.2760 & 0.3202 & 0.7108 \\ 0.2391 & 0.2450 & 0.8179 & -0.4624 \end{matrix}$$

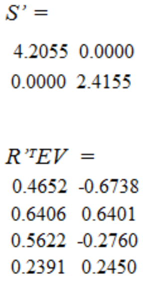


$$R^{T}_{EV} = \begin{matrix} 0.4652 & 0.6406 & 0.5622 & 0.2391 \\ -0.6738 & 0.6401 & -0.2760 & 0.2450 \\ -0.2312 & -0.4184 & 0.3202 & 0.8179 \\ -0.5254 & -0.0696 & 0.7108 & -0.4624 \end{matrix}$$

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 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 to the original matrix in terms of the least squares [Geiß (2011); Khalaf and Ping (2013a)]. Each sentence and each term are represented as a k-dimensional vector in the space derived by the SVD. In the majority of the applications, the dimensionality k is much smaller than the number of terms in the TSM. In the previous example, SVD ranks the concepts by importance to the text. By reducing the rank to 2, only the first 2 concepts are retained. Thus, the ranking matrices for the example are as follows:



$$L'EV = \begin{matrix} 0.3966 & -0.1282 \\ 0.2860 & 0.1507 \\ 0.1106 & -0.2790 \\ 0.1523 & 0.2650 \\ 0.1106 & -0.2790 \\ 0.3012 & -0.2918 \\ 0.3966 & -0.1282 \\ 0.3966 & -0.1282 \\ 0.2443 & -0.3932 \\ 0.3615 & 0.6315 \\ 0.3428 & 0.2522 \end{matrix}$$



$$S' = \begin{matrix} 4.2055 & 0.0000 \\ 0.0000 & 2.4155 \end{matrix}$$

$$R'^{T}EV = \begin{matrix} 0.4652 & -0.6738 \\ 0.6406 & 0.6401 \\ 0.5622 & -0.2760 \\ 0.2391 & 0.2450 \end{matrix}$$

To compute the similarities between two sentences, the ranking matrix of R'^{T}_{EV} is employed as the input for the cosine distance equation. The cosine distance is common for measuring similarity and computing the distance between any two sentences. Given two vectors with attributes A and B, the cosine similarity θ is represented using a dot product as

$$\text{Similarity} = \text{Cos}(\theta) = (A.B)/\|A\|\|B\|. \qquad (2)$$

To calculate cosine similarities for the example, the R'^{T}_{EV} matrix is utilized for each sentence as illustrated in the following rule: sim(Si, Sj) = (Si• Sj)/(|Si| |Sj|). In our example, to calculate similarity for S1:
sim(S1, S2) = (S1• S2)/(|S1| |S2|) sim(S1, S3) = (S1• S3)/(|S1| |S3|) sim(S1, S4) = (S1• S4)/(|S1| |S4|).

sim(S1,S2) = (((0.4652 * 0.6406) + (−0.6738 * 0.6401)))/(√((0.4652)^2 + (−0.6738)^2) * √((0.6406)^2 + (0.6401)^2)) = −0.1797

sim(S1,S3) = (((0.4652 * 0.5622) + (−0.6738 * −0.2760)))/(√((0.4652)^2 + (−0.6738)^2) * √((0.5622)^2 + (−0.2760)^2)) = 0.8727

sim(S1,S4) = (((0.4652 * 0.2391) + (−0.6738 * 0.2450)))/(√((0.4652)^2 + (−0.6738)^2)* √((0.2391)^2 + (0.2450)^2)) = −0.1921

S3 returns the highest value: pair S1 with S3. The same method is used to compute the similarities between S2, S3 and S3 S4.

Consequently, similar sentences (cosine distance>threshold) are joined to create a new sentence cluster. A new matrix is subsequently created from this cluster and the remainder of the sentences. After applying SVD, all sentences are compared by pairwise comparison. This process is repeated until the distance of the similarity between the document sentences is larger than the previously indicated threshold.

5. Speech corpora

Almost all experiments were conducted on Malay broadcast news. To demonstrate the performance of the proposed algorithms, a transcript produced manually from spoken broadcast news was used to evaluate the ASR results [Tan et al. (2009)]. The databases used for this transcript are collectively called the Malay broadcast news corpus; these broadcast news documents are collected at Universiti Sains Malaysia [Tan et al. (2009)]. The data set includes ~25 hours of transcribed speech. The broadcast news stories included multiple speakers and recordings in noisy environments. Furthermore, the data set includes different types of news, such as local, political, and sports. None of the test sets overlaps with the ASR training set. For each complete corpus, the testing and training data consist of two corpora. The first corpus is a gold standard file (GSF) corpus representing the manual transcription of Malay broadcast news, and the second corpus is the speech recognition transcription outcome for Malay broadcast news. The training dataset is the speech file that used for the ASR system training by using a ~15-hour portion of the database. This dataset included 30 speakers with ~10,000 utterances. The ASR system was tested by using a ~10-hour portion of the database. This dataset included 18 speakers with 18 broadcast news shows and 379 stories. The word error rate in the testing dataset is 34.5% and 33.9% before and after acoustic model adaptation. The number of utterance is 4698.

The testing dataset was segmented into stories by three experts (Malay native speaker) and was used as the gold standard data for automatic story boundary identification.

6. Experimental results and discussion

A total of 15,000 HTML documents from the March 2011 Malay broadcast news networks were collected and used to train LSA. These documents are news stories for different news domains and involve 38,218 term types in the corpus for building the LSA space. We used a TF-IDF weighting scheme with log weighting of the term frequencies and document frequencies to construct the document using the term matrix. The system then compared the folded-in test data directly with the SVD training data using cosine distance metrics (thus placing the vectors into the training and test sets of a classifier) as the folding-in procedure projected the test data into the SVD space. We performed different experiments by varying the final LSA dimensions. Different k-dimensional clustering spaces were built in this experiment. We report only the most remarkable results.

We conducted experiments using the LSA (as a baseline) with and without the noun unification approach. To train the edit distance to determine the best threshold, we used three speech file datasets including 6,474, 2,241, and 1,983 utterances with different speakers. We trained the edit distance with different thresholds (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9) using the three datasets. The threshold for the best edit distance was set to 0.6.

We also used 1 dataset to create a confusion Matrix that include 15734 utterances. The number of phonemes in the Malay pronunciation is 36. These phonemes are segmented into 3 groups: 6 vowels, 3 diphthongs, and 27 consonants. Fig. 4 shows the distribution of the Malay phonemes in the trained and testing data (ASR transcription).

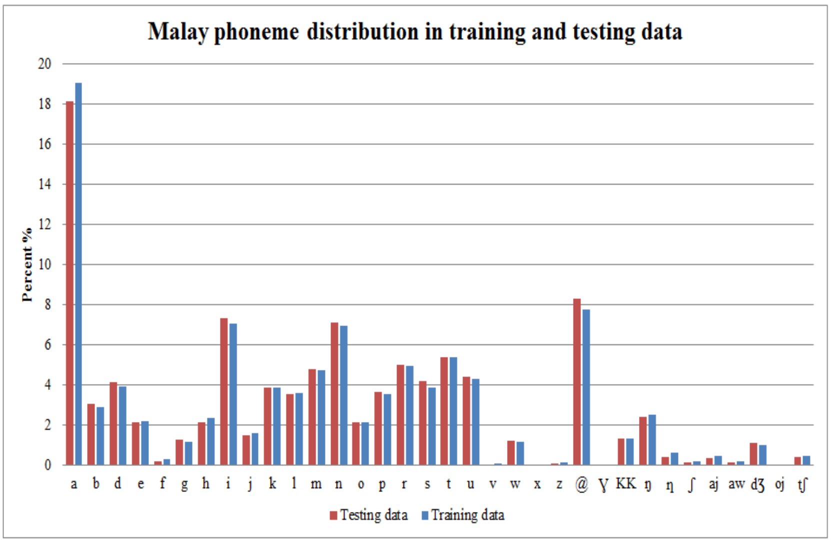


Fig. 4. Distribution of Malay phonemes in the trained data and the testing data

Word-bag selection often increases the clustering accuracy by eliminating noise features. A *noise term in a word-bag* is a term that degrades the effectiveness of the clustering performance when added to the document representation. Therefore, for baseline LSA without noun unification, the LSA using the common approach was applied to the selection of terms in a word-bag using the *WEKA* package with the best first approach. In the proposed approach, we used stemming verbs and noun unification as a word-bag for LSA.

Errors resulting from story boundary identification were measured using the F-measure, precision, and accuracy. We evaluated the effectiveness of the story boundary identification module using Malay spoken documents containing ~380 stories in different domains (e.g., politics, economics, sports, local news, and international news). We evaluated two corpora. The first corpus was a 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 5. Performance of the SBI module

	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 proposed system using LSA with the noun unification algorithm achieved an F-measure of 0.791, whereas this value was 0.613 for the LSA system without the noun unification algorithm.

The researcher performed a descriptive analysis of the SBI results using the Statistical Package for Social Sciences (SPSS) version 19. The researcher computed the Wilcoxon value (W-value) and the p-value (p) for each independent variable for LSA method, with and without the noun unification approach. The independent variable represents the F-measure of the segmentation stage for 18 broadcast news shows (N).

The null hypothesis H_0 suggests equivalent probabilities of the F-measure for the Malay broadcast news shows. The alternative hypothesis H_1 suggests different probabilities. The results show that the null hypothesis can be rejected in favor of H_1. For all three SBI methods, the probability that H_0 applies is $p \leq 0.05$. Table 4.11 shows all statistical results for the independent variables.

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

For the independent variable 3 in the LSA with and without the noun unification approach, the W-value is 2, whereas the critical value of W for N=18 at $p \leq 0.05$ is 40 and the p-value is 0.00028. Therefore, the result is significant at $p \leq 0.05$.

7. Conclusions

The identification of broadcast news story boundaries is critical to many natural language processing applications, such as topic identification and story classification. The proposed system uses the pronunciation forms to identify story boundaries based on noun unification. LSA is frequently used in clustering methods due to its excellent performance and strong foundation of semantic principles. In this study, the LSA model was used to identify boundaries of spoken MBN stories using a noun unification approach. The new algorithm resulted in fewer errors and better performance compared with the original LSA algorithm. The proposed system using LSA with the noun unification algorithm achieved an F-measure of 0.791, whereas for the LSA system without the popular noun unification algorithm achieved an F-measure of 0.613. Based on the findings of this study, several improvements to the system were identified. For instance,

1. Apply the proposed algorithms to other languages, such as English and Arabic languages.
2. Apply the proposed algorithms to non-planning read documents (spontaneous spoken document), such as interviews and meetings.

References

ALABBAS, M., KHALAF, Z. A. & KHASHAN, K. M. (2012). BASRAH: an automatic system to identify the meter of Arabic poetry. Natural Language Engineering-Cambridge University Press 2012, 1-19.

BANERJEE, S. & RUDNICKY, A. I. (2006). A TextTiling based approach to topic boundary detection in meetings.

BLEI, D. M. A. M., P. J. (2001). Topic segmentation with an aspect hidden markov model. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’01. New Orleans, Louisiana, USA.

CHAN, S. K. & L. XIE, A. H. M.-L. M. (2007). Modeling the statistical behavior of lexical chains to capture word cohesiveness for automatic story segmentation. in Proc. Interspeech.

CHOI, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. Seattle, WA, USA.

CHOI, F. Y. Y., WIEMER-HASTINGS, P., AND MOORE, J. (2001). Latent semantic analysis for text segmentation. In Proceedings of EMNLP. Pittsburgh, PA, USA.

DIAO, H., BAI, Z. & YU, X. (2010). The Application of Improved K-Nearest Neighbor Classification in Topic Tracking. IEEE, 978-1-4244-8035-7.

EISENSTEIN, J. (2009). Hierarchical text segmentation from multi-scale lexical cohesion. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Boulder, CO, USA.

FRAGKOU, P., PETRIDIS, V., AND KEHAGIAS, A. (2004). A Dynamic Programming Algorithm for Linear Text Segmentation. Journal of Intelligent Information Systems, 23(2), 179-197.

GALINA, A. A. (2007-2013). English Vowel Sounds- http://usefulenglish.ru/phonetics/english-vowel-sounds [Online]. Moscow, Russia. [Accessed].

GALLEY, M., MCKEOWN, K., FOSLER-LUSSIER, E., AND JING, H. (2003). Discourse segmentation of multi-party conversation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. Sapporo, Japan.

GEIß, J. (2011). Latent semantic sentence clustering for multi-document summarization. Ph.D, University of Cambridge.

HEARST, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics, 23, 33-64.

JAIN, A. K. (2010). Data Clustering: 50 Years Beyond K-Means1. Pattern Recognition Letters, 31, pp. 651-666.

JEROME, R. & BELLEGARDA, A. (2005). Latent semantic mapping. IEEE signal processing magazine, 5, 70-80.

KHALAF, Z. A., ALABBAS, M. & PING, T. T. Year. BASRAH: Arabic Verses Meters Identification System. In: IALP, 2011 Penang-Malaysia. IEEE, 41-44.

KHALAF, Z. A. & PING, T. T. Year. Automatic Identification of Broadcast News Story Boundaries Using the Unification Method for Popular Nouns. In: FEDCSIS, 2013a Poland. IEEE.

KHALAF, Z. A. & PING, T. T. (2013b). Unsupervised Identification of Story Boundaries in Malay Spoken Broadcast News. Journal of Emerging Technologies in Web Intelligence, 5, 28-34.

LU, M.-M., XIE, L., FU, Z.-H., JIANG, D.-M. & ZHANG, Y.-N. (2010). Multi-Modal Feature Integration for Story Boundary Detection in Broadcast News. IEEE, 420-425.

MENGLE, A. V. & PALMER, D. (2004). Trainable News Broadcast Boundary Identification Using Feature Density.

MISRA, H., YVON, F., JOSE, J. M., AND CAPPE, O. (2009). Text Segmentation via Topic Modeling: An Analytical Study. In Proceeding of the 18th ACM Conference on Information and Knowledge Managemen. Hong Kong.

MULBREGT, P. V., CARP, I., GILLICK, L., LOWE, S., AND YAMRON, J. (1998). Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In Proceedings of 5th International Conference on Spoken Language Processing. Sydney, Australia.

NARANG, V. D., G. ; MISRA, D. Year. Development of Acoustic Space in 3 to 5 Years Old Hindi Speaking Children. In: International Conference on Asian Language Processing (IALP), 2011 Penang-Malaysia. 236-239.

OSTENDORF, M., B. FAVRE, R. GRISHMAN, D. HAKKANI-TUR, M. HARPER, D. HILLARD, J. HIRSCHBERG, H. JI, J. G. KAHN, Y. LIU, S. M., E. MATUSOV, H. NEY, A. ROSENBERG, E. SHRIBERG, WANG, W. & WOOTERS., C. (2007). Speech Segmentation and its Impact on Spoken Document Processing.

QIN, L. (2013). Learning Out-of-Vocabulary Words in Automatic Speech Recognition. Ph.D., Carnegie Mellon University.

SENAY, G. E., LINAR`ES, G. & LECOUTEUX, B. (2011). A Segment-Level Confidence Measure For Spoken Document Retrieval. ICASSP, pp. 5548-5551.

STOKES, N., CARTHY, J. & SMEATON, A. F. (2004). SeLeCT: a lexical cohesion based news story segmentation system. AI COMMUNICATIONS, 17, 3-12.

SUN, Q., LI, R., LUO, D., AND WU, X. (2008). Text segmentation with LDA-based Fisher kernel. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies.

TAN, T.-P., HAIZHOU, L. L., KONG, T. E., XIONG, X. & AL., E. (2009). Mass: A Malay Language LVCSR Corpus Resource. Cocosda'09. Urumqi, China.

UTIYAMA, M. A. I., H. (2001). A statistical model for domain-independent text segmentation. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. Toulouse, France.

WIKIPEDIA. (2013). Silent letter - http://en.wikipedia.org/wiki/Silent_letter [Online]. [Accessed].

XIE, L. & J. ZENG, A. W. F. (2008). Multi-scale TextTiling for automatic story segmentation in Chinese broadcast news. in Proc. Asia Inf. Retrieval Symp. LNCS.

XIE, L., YANG, Y. L. & LIU, Z. Q. (2011). On the effectiveness of subwords for lexical cohesion based story segmentation of Chinese broadcast news. Information Sciences, 181, 2873-2891.

YAARI, Y. (1997). Segmentation of expository texts by hierarchical agglomerative clustering. In Proceedings of the Conference on Recent Advances in Natural Language Processing. Tzigov Chark, Bulgaria.

YANG, X. L. A. Y. (2008). Subword lexical chaining for automatic story segmentation in Chinese broadcast news. in Proc. PCM.