Text Adaptation for Mobile Digital Teletext

Chengyuan Peng, Petri Vuorimaa
Department of Computer Science and Engineering
Helsinki University of Technology
E-mail: pcy@tml.hut.fi, Petri.vuorimaa@hut.fi

Abstract

Small and varying screen sizes of mobile devices pose a big problem for digital Teletext service to display its content. It is difficult to display all the text information on a small screen, where page scroll or transparent page to live video is not practical. This paper presents an adaptive text extraction method which can automatically extract key information from original text and keep semantic meanings as close as possible. We combine both statistical methods and coarse coding algorithm from neural science to shorten long text sentences in terms of generalization. The experiment results show that the methods are efficient and effective.

1. Introduction

Digital TV has a popular information service called digital Teletext, which is a simplified web service specific to DTV. There are two possibilities to transmit pages for mobile digital Teletext: one way is to transmit two or more versions' pages, one version for original DTV and another one for mobile DTV as did in some web services; another way is to transmit only one version's pages and receiver processes the content separately. The first possibility is unacceptable because the bit rate for transferring content via digital terrestrial broadcast is at most 15 mbps for mobile transmission. Also the screen sizes of different mobile devices are varied and it is not possible to cover all versions.

A short text summary can be a substitute for a full text page on a mobile device with limited screen size. One of the constraints is that if ordinary digital Teletext pages are extracted in a receiver there must need extra computing resources to process these pages. Usually a digital Teletext page consists of a title, large text and a few images. The objective of this paper is to automatically summarize a long page and finally to get a shorter one containing one or two short sentences, which are extracted text other than original sentences.

The number of sentences and the number of words extracted in the sentences should be able to adapt to different screen sizes. Essentially, the resulted text from these sentences should encompass the most relevant points of the input text as much and concise as possible. Furthermore, the adopted extraction methods must as far as possible be linguistic ones, do not require extensive world knowledge, and do not make use of corpus and training data.

The problem in part is similar to text summarization or text extraction task. In general, summarization involves semantic analysis and applying world knowledge for clearest presentation, and uses natural language processing methods and/or statistical techniques to achieve a significant reduction in the quantity of text with minimal reduction in information content [1]. Although effective and human-quality text summarization is difficult to achieve without natural language understanding, various approaches have been studied in this area [1]. Currently, there are two main approaches to automatically extraction from a text document, e.g., statistical techniques and natural language processing technique [1]. Some of the techniques may be used independently or combined when building summaries.

In spite of great advance in extraction technique, many problems still exist. This paper gives methods based on statistical and neural representation techniques. The aim is to find true summarizing concepts using statistical cues (e.g., head line words, high frequency words), task prior knowledge, and features of text to select key sentences to form summaries. More precisely, we use coarse coding approach, i.e., by definition a neuron's representation responds to many inputs. The advantage of using this method is that more accurate representations can be represented by suitable combination of the coarse representations. The basic idea in this paper is to see a summary as a generalization of representation units (key phrases) in terms of meaning.

2. General extraction processes

General summary extraction processes for a page in this paper include processing the words, forming an idea of the overall meaning of a sentence, and weighing the sentence in making a decision. The input text page is first segmented. The segmentation process produces output text having one sentence per line by splitting at

Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)
0-7695-2100-2/04 $ 20.00 IEEE

exclamation marks, question marks or full stops, which indicate sentence boundaries.

After that, the system will calculate the frequency of each word and select high frequency words that are not in a stop-list. Then each sentence will be assigned a weight according to the key words frequencies because words that are repeatedly used in the text are relevant for the domain and will be more likely to remain in the text. The weight of a sentence is calculated according to function (1), which is a linear combination of the frequencies of high-frequency words in a sentence.

$$W = \sum_{k=1}^{M} (Fk + C) + \sum_{i=1}^{N} Fi \quad \text{-----------------------------} \quad (1)$$

Where, M: number of words appeared in page tile; N: number of words with most high-frequency; W: weight of the sentence; F_k: frequency of a word from page title; C: a constant weight assigned for page title; word; F_i : frequency of one of the most high-frequency words.

In next stage, the system ranks sentences in order of weights and selects important sentences with the highest weights from these weighted sentences. The number of sentences selected depends on the screen size of a mobile device. The system will then find key phrase clusters locally in a selected sentence. Key phrases can be recognized reliably with coarse coding algorithm. These key phrases provide precisely the elements that are required for stating the event patterns of interest in a summary. Finally, the system will assemble portions of extracted clauses using coarse coding representation in order to present the summary in natural language as much and fluent as possible.

3. Coarse coding representation

Coarse coding that is a kind of distributed representation encodes features accurately using as few units as possible [2]. It pays to use units that are very coarsely tuned so that each feature activates many different units and each unit is activated by many different features. The principle underlying coarse coding can be generalized to non-continuous space by thinking of a set of neurons as the equivalent of a receptive field. The receptive field of a neuron in visual cortex is the area of the visual field to which the neuron is sensitive [4]. The location of a feature in the visual field is accurately pin-pointed when it falls within the receptive fields of a number of neurons. The joint activity of several neurons indicates that the feature is located at the intersection of the active units' receptive fields.

The neuron becomes active if there is movement or change within this field [4]. In visual cortex, individual neurons often have large receptive fields which have considerable overlap with other neurons. Accuracy increases with receptive field diameter and coarse coding is only effective when features that must be represented are relatively sparse [2].

Coarse coding scheme was once used to model the mapping from a sentence containing polysemous prepositions to a representation of the sentence's meaning [5]. The useful view for a text summary task from [5] is that the primary unit of linguistic storage is not the word, but is some larger piece, such as phrase, clause and sentence. On this view, words are laid in memory with their frequent left and right neighbors, and the meaning that is stored with them is the meaning of the unit as a whole rather than the separate meanings of the individual words in the unit. This large, heterogeneous set of phrases and word combinations is not a static list, but is stored in distributed fashion because the tasks generally require the simultaneous consideration of many pieces of information.

Meanings are patterns of activation across a pool of neurons. A word is akin to a neuron with a receptive field that may vary in size and the degree to which it overlaps with the receptive fields of other words. The meaning of words is usually tightly linked to their typical contexts of occurrence. The mapping from a key word to meaning is mediated by a coding scheme which varies in its coarseness. If we use coarser coding, we will get wider meaning span. We can obtain a text phrase with variable words by adjusting the diameters of a neuron's receptive field.

Next we consider a useful statistical measure in text summarization - word frequency usage. Intuitively, words that cover a large semantic territory will be used often [5]. The characteristics of words that facilitate access are likely to include high frequency, high valence match with adjoining words, and high routinization of word combinations. In many languages some adjuncts are more tightly bound to their head nouns which are often high-frequency words than others. The basic noun group together with these adjuncts constitutes the complex noun group.

Finally, we discuss the concept about phrases combination to form a meaningful sentence. The meanings of most natural language utterances are not obtained by concatenating the meanings of component words, but that the concepts are the intended building blocks [5]. There is a continuum of context-dependence, with some words tightly associated with their typical neighbors, and others relatively independent. If we idealize a text summarization to be a distributed representation of neurons, we have a method for predicting what invariance will be extracted, and how the degree of specificity of an extracted invariance is related to the pool of utterances it summarizes.

We use 1-D tile coding as one of the forms of coarse coding. The algorithm was once used in reinforcement learning as a way of representing the values of a vector of continuous variables as a large binary vector with few 1s and many 0s [3]. The binary vector is not represented explicitly, but as a list of the components that are 1s. It is a technique for creating a set of Boolean features from a set of continuous features. Based on this theory, we assume that the meaning of the whole is the sum of the meanings of the part plus some additional semantic component that cannot be predicted from the parts and therefore we can derive a correct representation of the meaning of a sentence.

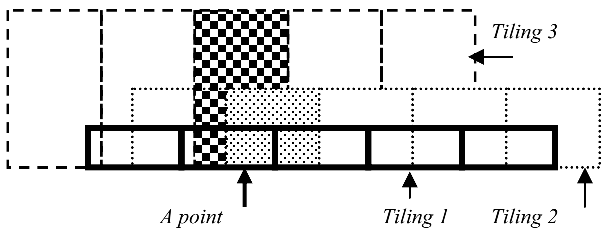

Figure 1. One-dimensional tile coding

The basic idea is to lay offset tilings with some amount of tiles over 1-D continuous feature space. A point in the continuous feature space will be in exactly one tile for each of the offset tilings when a tiling moves. A point is representing a state with features that overlapped. For example, in Figure 1, there are three offset tilings, each has five tiles. Given a state (i.e. a point) which a binary feature is present, the state lies in three tiles which are overlapped the point: the second tile of tiling 1, the second tile of tiling 2, and the third tile of tiling 3 and therefore the state's location is coarsely coded.

A point coordinate in domain space must first be transformed to tiling space coordinate in each tiling. The transformation can be calculated in the following simple geometric coordinate transformation:

$X = X' /(X1/X2)$ -- (2)

Where, X: a point coordinate in tiling space; X′: a point coordinate in domain space; X1: the length of domain space; X2: the length of tiling space. Then each point coordinate in tiling space must also be transformed to tiling's relative coordinate when a tiling moves. The number of the tiles (i.e. the coarseness of tile coding) and the movement of a tiling are sensitive to the frequent left and right neighbors of a word in a sentence.

A coarsely-coded representation of a state responds to many inputs and therefore promotes generalization. The key idea in this summarization is to use the concept of generalization of sparse key words to represent a full sentence's meaning.

4. Extraction rules and example

We employ a part of features of sentences or words, e.g., sentence location, sentence length, word frequency, and headline words. The first step is to count word frequencies and select first a few high-frequency words and title words. The second step is to calculate sentence weights and select first a few sentences with highest weights. The number of sentences selected depends on screen size of a mobile device. The final step is to extract words from the selected sentences and it concerns key phrase and connection words extractions or clustering.

Rules of key phrases selection are: According to sentence length to decide to use how many high-frequency words; Partition a sentence six times using six tilings and select one tile from each partition. The selected tile is the position of a neighbor near the high frequency word in a sentence. Rules of connection words selection or clustering are: Among connection words, low-frequency words are much more than high-frequency words, and are almost stop words; Select first a few high-frequency words; if there are no enough words, select lower frequency words; if there are many of them, select first and/or second words; Move tilings both forwards and backwards, select tiles. Overlap the selected tiles in the center of the high-frequency word.

Table 1. An example text

no.	sentence	w/t
title	Digital camera is UK's top gadget	
(1)	A **digital camera** *has been* voted as the **top** gizmo of the **year** *in a* magazine poll of **gadget** *lovers*.	36/4
(4)	Small, affordable **digital cameras** were big this **year**, *but pocket video devices* are expected to be the **gadgets** of 2004.	35/4
(7)	But, he says, *2003* was definitely the **year** of the **digital camera** *with* sales overtaking film **cameras** *for the first* time.	31/5
(13)	Better *models have* since *been released* and Mr *Vaughan predicts* 2004 will be the **year** *of video in pockets*, on *mobiles but also video jukeboxes*.	24/5

Table 1 lists extracted sentences from a text which originally includes 17 sentences, sentence numbers in the text, and weights of extracted sentences and tiles used. The key words except for stop words and their frequencies are in the form *high-frequency words*(*title word score, frequency in text,*)*total frequency* as follows: ***digital(3, 4)7***: ***camera(3,5)8***: ***UK(3,1)4***: ***top(3,2)5***: ***gadget(3,5)8***: *mobile(0,4)4:* ***year(0,8)8***: *3G(0,4)4:* *video(0,4)4;* *model(0,4)4*. We then use tile coding algorithm and extractions rules to extract key phrases based on top six key words (shown in bold face) and obtain their positions in sentences by moving a tiling forward 8 times. As a result, each key word has some amount of tiles fell in the tilings. In Table 1, for example, first sentence has four tiles and therefore the key phrases selected are underlined. The combination of these key phrases can become a very short summary, although the semantic meaning is not very complete.

The identified key phrases need to be merged with other words in the sentences to form a summarized sentence with more complete semantic meaning. We partition the rest of pieces of words in a sentence separately. The first a few high-frequency words to be selected include also stopwords. We move tilings forward and backward 8 times. In Table 1, the words that are italic are selected by partitioning the piece of words. Finally, we combine the underlined and italic words to form a more complete sentence summary.

5. Experiments and evaluation

The methods were tested and evaluated on a set of 100 pages from BBC TV schedules and world news. The content of the pages covers from news, sports, business, to technology, travel, and entertainment. We use compression ratio and extraction accuracy to measure the performance of the extraction methods. The outcome is summarized in Table 2. Compression ratio here means the percentage of removed words relative to the total words in a sentence. Extraction accuracy indicates the degree to which its extraction corresponds to the human-generated extractions, i.e., how much it captures the whole meaning of the text. Very short summary is designed to measure the performance of key phrase extraction. Short summary is to measure the performance of merging of key phrases and connected words.

Table 2. Statistical results

Summary	Compression ratio (words)	Extraction accuracy
very short	0.5747250	0.5505125
short	0.1470778	0.9090536
whole page	0.6366977	0.9983333

The question of interest in both very short and short summaries is how accurately a feature is encoded as a function of the radius of the zones. In theory, accuracy is proportional to $N*R_{K-1}$ [2], where N is the number of zones fall in one point, R is the radius of the zones, and K is the dimension of the tiling space. In our cases, K is one, and therefore the accuracy is proportional to N. In this paper, N indicates the length of a phrase in words. The more accurate the meaning is the wider word span or longer length of a phrase is. This means that coarse coding is only effective when the features that must be represented are relative sparse. The number of tilings used does not affect the word span in this problem. It can only contribute the accuracy of positions.

The sparser the tile in a tiling (i.e. fewer grids in a tiling, we use six grids in this experiment) is, the wider word span (i.e. wider reception field of a neuron) is, which means more words are to be selected. The word span also depends on a sentence length. The longer the sentence is, the wider the word span is (cf. transformation (2)). The wider word span is, the more complete the meaning of a summary is, but less compression. The meaning also depends on the number of key words selected and used. With these relations, extracted summary can automatically adapt to different screen size.

6. Conclusions

Adaptive text extraction based on coarse coding approach has three main merits. Firstly, the generated text is consistent with the original text in the connectives between key phrases. Secondly, various lengths of summary can be extracted easily and can achieve higher performance on simple tasks by adjusting such as the number of tiles in tilings and the amount of high-frequency words. Thirdly, the coarse-coding-based extraction does not need prepared knowledge or understanding natural language and neural network training, can be used for texts of any domain, and works fast and effectively. It is feasible and portable, and can be used in practical system.

Our approach of using coarse coding was inspired by the conceptually simple fact that the meaning of a sentence can be a form of generalization. The key to the whole problem is to do the tile coding on a feature space. Although the full linguistic complexity of the texts is often very high, e.g., with long sentences and interesting discourse structure problems, the relative simplicity of the extraction method allows much of this linguistic complexity to be bypassed. Limitations under this method can occur mainly due to errors in the high-frequency words analysis (e.g. sometimes trivial information also mistakenly picked out) and the connection word extraction.

We expect to bring broader and more complete text extraction capabilities in the future. We didn’t consider some difficult tasks, e.g. information may need to be combined across several sentences. This perhaps requires complex semantic analysis, discourse processing, grouping of the content using world knowledge, and eliminating text redundancy, etc.

7. References

[1] A. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, “Summarizing Text Documents: Sentence Selection and Evaluation Metricx”, *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR’99)*, pp. 121-128.

[2] G.E. Hinton, J.L. McClelland, and D.E. Rumelhart, *Distributed Representations, Parallel distributed processing: Explorations in the microstructure of cognition*, *Vol. 1. Foundations*, Cambridge, MA: MIT Press, 1988.

[3] R. Sutton, *Reinforcement learning: An introduction*, Cambridge, MA: MIT Press, 1998.

[4] S. Haykin, *Neural networks: a comprehensive foundation.* Upper Saddle River (NJ) Prentice Hall, 1999.

[5] C.L. Harris, "Coarse coding and the lexicon”, *Fuchs & Victorri*, 1994, pp. 205-230.