Text Compression and Prediction Using Language Model

Wut Yi Mon, Mya Thida Kyaw
University of Computer Studies, Mandalay
wutyimon@gmail.com

ABSTRACT

Language model in Natural Language Processing is one of the most important fields carried out in the world of Artificial Intelligence. The purpose of this paper is to express about stipulated abbreviation method and statistical language model in Natural Language Processing. The system solves the problem of improving the efficiency of natural language text input under degraded conditions by taking advantage of the informational redundancy in natural language. It takes advantage of the duality between prediction and compression. It allows the user to enter the English text and text in compressed form, using a simple stipulated abbreviation method. The system decodes the abbreviated text by using statistical language model. This paper is typically based on processing of sentences in English between the computer and the user. This is implemented by using C# language, Component One Software and Microsoft access database.

1. INTRODUCTION

The problem of text input with devices under degraded conditions is not new, disabled users, for instance, have had to interact with computers using sometimes severely degraded means, including mouth sticks, symbol-scanning systems, eye-gaze tracking, and so forth. The problem has renewed currency, however, because of the increased prevalence of small and embedded computing systems (handheld computers, cell phones, digital video recorders, and the like) for which traditional text input and verification modalities (keyboard and monitor) are impractical.

Natural language text is highly redundant. The traditional approach to take advantage of this redundancy relies on prediction of the user's text. For instance, many cell phones have the technology to predict the most likely word based on the initial letters typed by the user. The user is required to merely verify the prediction rather than typing the remaining characters. The compression method can achieve text input efficiency improvements where the prediction method has not. The compression method limits the cognitive load increase. Compression methods are Stipulated Compression Method and Natural Compression Method. Previous approach is based on the idea of prediction of the text, required the user to take overt action to verify or select the system's predictions. These predictive methods are only useful and have only found acceptance among severely disabled users.

This paper is based on the duality of prediction and compression. A good statistical model can generate good predictions, can be used for compression as well. It used Stipulated compression Method in nonabbreviatory and abbreviatory. If the user enters compressed text, the compression of which is based on a good predictive model, we can then use that model to decode the compressed text into the full text. The cognitive load increase is limited to that induced by the ability to fluently generate compressed text. The generation of the compressed text is not an interactive task that requires task switching,

verification of system proposals, selection of options, and so forth. The advantage of the compression approach over the previous prediction approach is clear [1].

2. RELATED WORKS

Text input methods based on predicting what the user is typing have been widely investigated; see the work by Darragh and Witten (1992). Such systems can be found in a variety of tools for the disabled, and some commercial software, most notably the T9 system from Tegic. Methods based on static lookup in a fixed dictionary of codes for words or phrases include Vanderheiden's Speedkey (1987), along with a wide range of commercial keyboard macro tools that require user customization. All rely on the user's memorization of the codes, which must be extensive to provide much compression advantage. Systematic stipulated compression models can be found hidden in stenographic methods such as Speedwriting, though there is no provision for automated decompression.

A recent dynamic prediction approach is used by Dasher (Ward and MacKay, 2002), a system in which the predicted characters stream onto the screen towards the constructed sentence, in shaded boxes of sizes proportional to their likelihood, and the user has to choose the next character using a mouse or an eye-tracking device. Dasher's predictions are based on a text compression algorithm called Prediction by Partial Match (PPM) (Cleary and Witten, 1984; Moffat, 1990). Some human factors research on the design of command abbreviations for small vocabularies has been performed. John et al. (1985), for instance, show that vowel dropping leads to more easily recalled abbreviations but slower throughput than abbreviations based on escaped special characters. Extrapolation of such results to abbreviation of arbitrary text is problematic, but the results are not inconsistent with the possibility of throughput benefits under reasonable conditions [1].

3. APPROACHES TO NATURAL LANGUAGE PROCESSING

Natural language processing approaches fall roughly into four categories: symbolic, statistical, connectionist, and hybrid. Symbolic and statistical approaches have coexisted since the early days of this field, Connectionist NLP work first appeared in the 1960's. For a long time, symbolic approaches dominated the field, In the 1980's, statistical approaches regained popularity as a result of the availability of critical computational resources and the need to deal with broad, real-world contexts. Connectionist approaches also recovered from earlier criticism by demonstrating the utility of neural networks in NLP. This section examines each of these approaches in terms of their foundations, typical techniques, differences in processing and system aspects, and their robustness, flexibility, and suitability for various tasks [2].

3.1. Statistical Approach

Statistical approaches employ various mathematical techniques and often use large text corpora to develop approximate generalized models of linguistic phenomena based on actual examples of these phenomena provided by the text corpora without adding significant linguistic or world knowledge. In contrast to symbolic approaches, statistical approaches use observable data as the primary source of evidence.

Statistical approaches have been used in tasks such as speech recognition, parsing, part-of-speech tagging, and so on [2].

4. COMPRESSION METHOD

There are two possibilities.

Stipulated Compression: First, we can conform the user's behavior to a particular model by stipulating a compression method, so long as the stipulated method is simple and easily learnable. In practice, the learnability requirement means that the compressed forms of words must be abbreviations of some sort. In fact, the literature

has traditionally distinguished prediction approaches from abbreviation approaches (Vanderheiden and Kelso, 1987), which have been taken to be of this stipulated variety.

Natural Compression: Alternatively, we can try to conform the model to the user's natural behavior by allowing a natural compression method, one that users would naturally turn to when compressing text. As it turns out, there seems to be a more or less standard compression method, a kind of ad hoc abbreviation form, well understood by average writers of English, and best exemplified by the old advertising slogan "If u cn rd ths, u cn gt a gd jb". The sentence "We have conducted a thorough evaluation of this disabbreviation method." would be abbreviated as "W hv cndctd a thrgh evltn of ths dsbrvtn mthd" [1].

5. STIPULATED ABBREVIATION METHOD

Stipulated Abbreviation Method is used to compress the original text. Dropping all vowels in the sentences but retain the first letter even when it is a vowel and dropping of consecutive duplicate consonants. This method processes as a weighted finite-state transducer.

5.1. Component Transducers

Weighted finite-state transducers constitute a simple general technology for modeling probabilistic string-to-string transformations. Their nice closure properties, especially closure under composition, make them ideal for the present application in that the model can be composed as a cascade of simpler transducers in an elegant fashion. These include:

An *n*-gram language model (*LM*): The language model, which implements the $p(W)$ component of the generative model and implemented as a finite-state acceptor. Numbers and unknown words are replaced by special tokens.

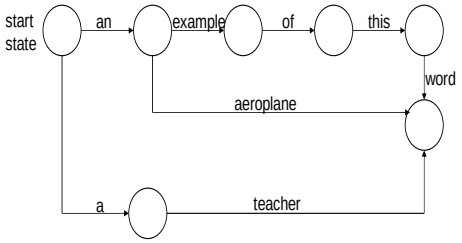

Figure 1. Language model

A spelling model (*SP*): This transducer serves the purely technical purpose of converting words into the sequence of characters that compose them. This change in token resolution is required since the language model operates on word tokens and the following transducers in the cascade operate on character tokens. This transducer is constructed by creating a separate path of states for each word, in which the word is first transduced to the null symbol, € , followed by the transduction of € to each of the word's letters, as illustrated for two words in Figure 2. To complete the loop, there is an added transition from the final state to the initial state that generates a space (represented as U in Figure 2). To compact the transducer, we determinize it on the input symbols.

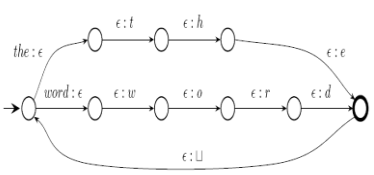

Figure 2. Spelling model

A compression model (*CMP*): This transducer implements the stipulated abbreviation model, removing non-initial vowels and doubled consonants. The transducer has a unigram memory of the last character seen. Starting from the second letter, any vowel is transduced to €. A consonant is transduced to €, if it is the same as the previous letter. This is illustrated in Figure 3, for an alphabet restricted to two letters—a vowel (a) and a consonant (b). Special symbols for unknowns and numbers, as well as punctuation marks are left intact. *CMP* implements the $p(A \mid W)$ component of the model, and is deterministic,

i.e., for any *W* and *abbrev*, it is either 0 or 1, depending on whether that sequence of words can be abbreviated as that sequence of letters.

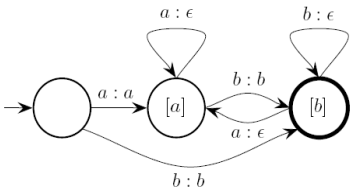

Figure 3. Compression model

An unknowns model (UNK): This transducer replaces the special tokens for unknowns and numbers with sequences of characters or digits, according to a simple generative model: reading the token <*NUM*> or <*UNK*> as input, it enters a loop emitting arbitrary digits or characters, respectively. The composition of these four transducers forms the entire abbreviation model as illustrated in Figure 4. The composed transducer is deterministic in the forward direction (with the exception of UNK), i.e., a given sequence of words has a single abbreviation [1].

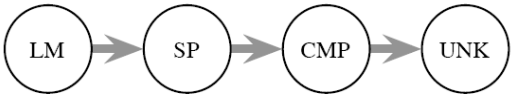

Figure 4. Abbreviation model

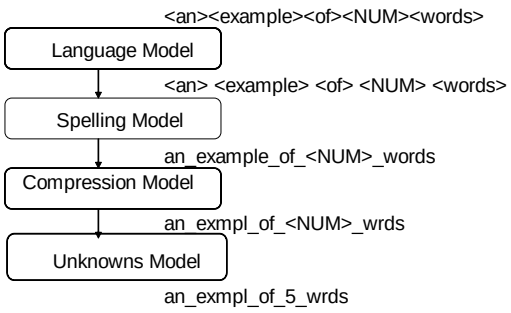

Figure 5. An example of abbreviation process[3]

6. STATISTICAL LANGUAGE MODEL

Statistical language model can generate good predictions. To decode text that has been abbreviated using the stipulated method. This model decodes word sequences that corresponding to the abbreviated character sequences. This model is constructed by composing a language model, representing the probability of a word-sequence, P(W) and an abbreviation model, representing the probability of the abbreviation A, given W, P(A/W). The composed model therefore models the joint probability, P(W,A)=P(W)P(A/W). Given a particular abbreviated form, A, we seek the most likely word-sequence, W, that could have generated it, i.e., argmax_w P(W/A) [1].

6.1. Bayes' Theorem

Let X be a data tuple. In Bayesian terms, X is considered "evidence". As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, we want to determine P(H/X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X. In other words, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X. P(H/X) is the posterior probability, or a posteriori probability, of H conditioned on X.

In contrast, P(H) is the prior probability, of H. The posterior probability, P(H/X), is based on more information than the prior probability, P(H), which is independent of X. Bayes' theorem is useful in that it provides a way of calculating the posterior probability, P(H/X), from P(H), P(X/H), and P(X) [4].
Bayes' theorem is

$$P(H/X)=P(X/H)P(H)\ /P(X)$$

In this paper, by Bayes' rule,

$$\text{argmax}_w\ P(W\ /\ A)=\text{argmax}_w\ P(W)\ P(A\ /\ W)\ /P(A)$$

P(W / A) =the probability of word-sequence given abbreviation
P(W) =the probability of word-sequence
P(A / W)=the probability of abbreviation given word-sequence
P(A) =the probability of abbreviation

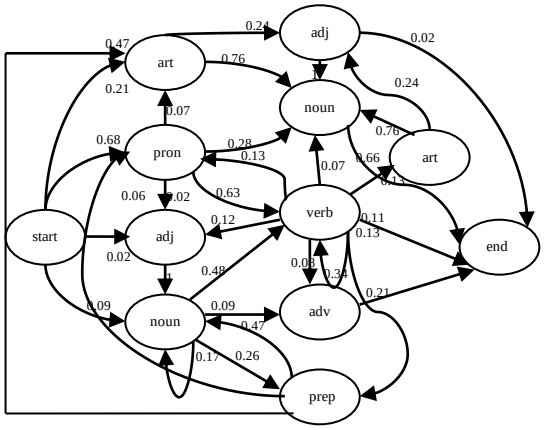

Figure 6. Structure of word sequences

7. SYSTEM DESIGN

This paper based on the natural language processing that is the artificial intelligence field. The system has two parts basically. These are encoding process and decoding process. In encoding process, the system allows the user to enter the English text and also allows to enter text in compressed form, using stipulated abbreviation method and special compressed form. The system encodes this text by using stipulated abbreviation method. In decoding process, the abbreviated text is decoded by using statistical language model to obtain the original text. Figure 7 and Figure 8 show the structure of encoding process and decoding process of the system.

7.1. System Flow Diagram of Encoding Process

The abbreviated text that has been abbreviated by using stipulated abbreviation method and input English sentence that the user wants to test must be entered into the system. The stipulated abbreviation method is used to compress the input text. The input text passes through this method to obtain the encoding text.

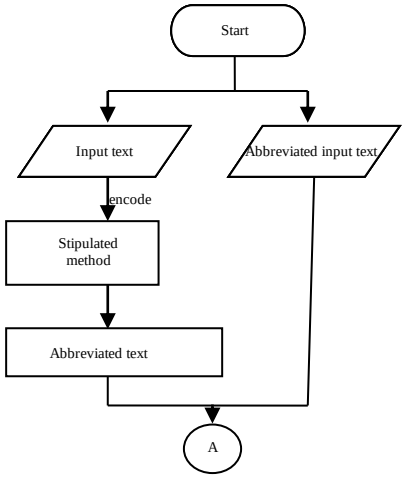

Figure 7. The encoding process

7.2. System Flow Diagram of Decoding Process

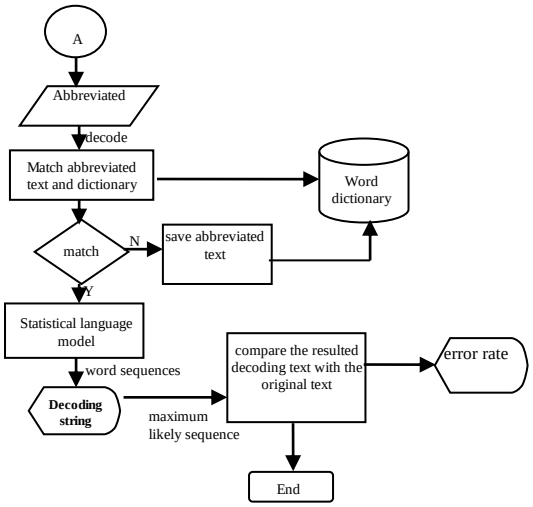

Figure 8. The decoding process

The abbreviated form has been compressed by using stipulated abbreviation method. It is entered into the decoding process of the system. This system matches the abbreviated text and dictionary. The dictionary consists of short forms, definitions and abbreviated text. If there is something wrong with checking dictionaries, the

abbreviated text is checked with the dictionaries whether it matches or not. If there is no suitable abbreviated text in database, the user has to add the new abbreviated text. If the abbreviated text matches with the new ones, it goes on the next step. And then the system is to decode the abbreviated text that has been abbreviated using the stipulated method. To decode the abbreviated text, the system uses statistical language model. This model can generate good predictions and decode word sequences that correspond to the abbreviated character sequences. Bayes' rule is constructed by composing a language model, representing the probability of a word-sequence, P(W) and an abbreviation model, representing the probability of the abbreviation A, given W, P(A/W). These rules choose the most likely word sequence among the word sequences. And then, the system compares the resulted decoding text that has been decoded using statistical language model and the original input text. The system displays the error rate and decoding string.

8. IMPLEMENTATION

This system contains two sections. These are Entry section and Abbreviation section. In the Entry section, there are two forms. These are short form (eg; aren't) and word definition (eg; are not). In this section, the user can type the new short form and new word definition. And then, the user can abbreviate the new one. The new abbreviated text will be saved in the database. Moreover, the user can modify and delete the existing abbreviated text in the database.

In the Abbreviation section contains five parts. In the input string part, the user can enter the input string. In the encoding string part, the system encodes the input string to obtain the encoding string. In the decoding string part, the system decodes the encoding string to obtain the decoding string. In the error rate part, system compares the input string and decoding string to produce the error count and correct percentage. In the graph part, the system counts the number of word in the input string and displays the bar chart. This chart compares the original text and correct text.

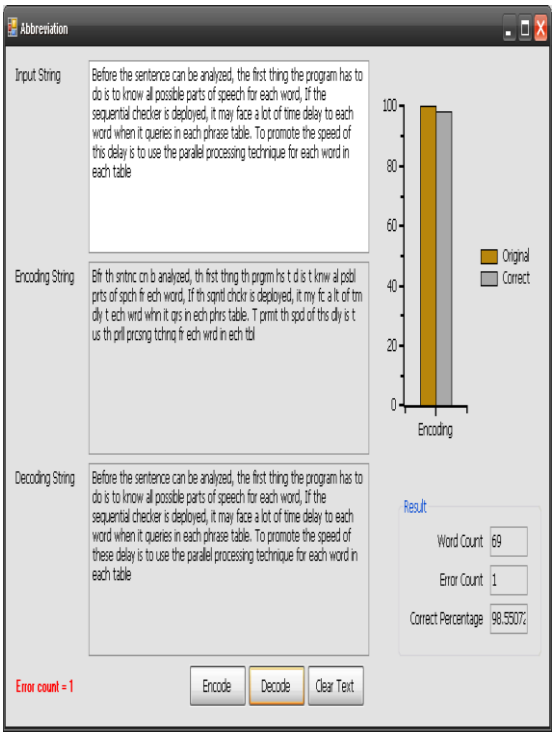

Figure 9. The abbreviation form of the system

9. PERFORMANCE ANALYSIS

We experimented with the training dataset size for the language model, using additional Myanmar Time Journal text as shown in Table1. As can be expected, accuracy improves with training data size. At 1.50 million words, the language model achieves 3.34% accuracy rather than the original simple trigram's 4.57%. Enlarging the training dataset size to 3.68 million words reduces the error even further. Since there is no shortage of plain English text, the only bounds on the training data sizes are dictated by performance considerations. At 1.50M words, the system still disabbreviates at rates well below 1 second per sentence. At 3.68M words, disabbreviation takes several seconds per sentence.

Table 1. Performance of the language model of training data size

Training data size million words	Average error rate percent	Standard deviation of error percent
0.40M	4.01%	0.25%
1.50M	3.34%	0.21%
3.23M	2.90%	0.19%

10. CONCLUSION

This system is implemented using stipulated abbreviation method and statistical language model. Stipulated abbreviation method provides a collection of tools for constructing weighted finite-state transducers, including n-gram language model, spelling model, compression model and unknown model. It uses to encode the input English text. Statistical language model serves to decode the abbreviated text. This system uses Bayes' rule to obtain the maximum likely word sequence. We trained the language model on a training set of Myanmar Time Journal articles, after performing several preprocessing steps. We performed evaluation studies on a held-out corpus of 5 sections of Myanmar Time Journal text (from January 2001) of about 50000 words each, for a total of roughly 280500 words. We limit the vocabulary of the model to about 15000 most frequent words. Increasing vocabulary size improves decoding accuracy but increases the language model size and consequently decoding time. This system can be enable modifications and extensions to the "Forgiving" abbreviation model. In this model, informal user experimentation has shown that whereas the stipulated model is fairly simple to learn, users will sometimes forget to drop all of the vowels or repeated consonants. Unfortunately, this leads to a failure to decode, as the basic model assumes strict adherence to deterministic letter dropping rules. A minimal change to the original compression model makes it nondeterministic in the forward direction by allowing a small probability,δ, of not dropping the required vowels and repeated consonants.

REFERENCES

[1] Stuartm M.Shieber and Rani Nelken, "Abbreviated Text Input Using Language Model", Division of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138 USA.

[2] Mats Dahllof, "Natural Language Processing", Dept of Linguistic and Philology, December 2, 2004, Uppsala University.

[3] Sturat M.Shieber and Ellie Baker, "Abbreviated Text Input", Division of Engineering and Applied Sciences, Harvard University.

[4] Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques*, University of lllinois at Urbana-Champaign.

[5] David Jones, " Introduction to Natural Language Processing", Dept of Linguistic and Philology, July 2, 2003, Uppsala University.

[6] David Jones, " Statistical Language Model", Dicision of Engineering and Applied Sciences, Uppsala University.