

Honours Year Project Report

SMS Short Form Identification and Codec

By

Lee Ming Fung

Department of Computer Science

School of Computing

National University of Singapore

2004/2005

Honours Year Project Report

SMS Short Form Identification and Codec

By

Lee Ming Fung

Department of Computer Science

School of Computing

National University of Singapore

2004/2005

Project No: H079130

Advisor: Assistant Professor Kan Min-Yen

Deliverables:

Report: 1 volume

Abstract

In this thesis, we present a system that translates cryptic Short Messaging Service (SMS) messages with little recognized shortforms into readable messages in longform. The inherent fickleness of shortforms in user SMS messages makes shortform ambiguous, complicating the task of interpretation.

While traditional translation systems uses a dictionary-based approach where new found shortforms have no direct translation, our system proposes a system where text messages are first categorized into word, acronyms and shortforms with the latter passed into a matching framework based on phonetic similarity and maximum entropy for translation. To the best of my knowledge, there has not been any previous work done on such an approach.

Subject Descriptors

I.2.6 Learning

I.2.7 Natural Language Processing

Keywords

Text Processing, Machine Learning, Maximum Entropy, SMS Shortform codec, SMS Shortform Identification

Implementation Software and Hardware

PC Athlon XP 2500 with 512MB RAM, Windows XP, Java 2 SDK, SE 5.0, MAXENT, JWord Packages.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my project supervisor Assistant Professor Kan Min-Yen for his insightful advice and patience throughout the course of my research.

I would also like to thank my friends Victor Kasenda, Tan Siru, Allan Yap, Lee Thiam Chye, Tan Yee Fan and Qiu Jiang for their help and support during the course of this project.

List of Figures

Figure 1:	Shortform Identification and Categorization Model Overview	10
Figure 2:	Overview of Classification Process	14
Figure 3:	Shortform Decoder System Architecture.....	16
Figure 4:	Examples of phonetic mapping	17
Figure 5:	Example of context predicates for text to phonetic translation.....	18
Figure 6:	Example of context predicates for phonetic to word translation	19
Figure 7:	Overview of Process of Translating shortform(phonetic substitution).....	20
Figure 8:	Examples from corpus of actual words and corresponding shortforms.....	21
Figure 9:	Examples of segments and spaces	22
Figure 10:	Contextual Predicates Extraction examples for segmented shortforms.....	23
Figure 11:	Contextual Predicates example for training example for Letter Omission.....	24
Figure 12:	Contextual Predicates example for training example for shortform encoding	26
Figure 13:	Contextual Predicates example adapted from Phonetic Model	26
Figure 14:	Identification and categorization performance evaluation.....	29
Figure 15:	Phonetic and Letter Omission Model evaluation results	29
Figure 16:	Comparison between the baseline model and SMS Codec System.....	30

List of Tables

Table 1:	Examples of Shortforms.....	2
Table 2:	Letter Omission (Shortforms) Examples.....	4
Table 3:	Truncation (Shortforms) Examples	4
Table 4:	Phonetic Substitutions (Shortforms) Examples.....	4
Table 5:	Table of Categories	11
Table 6:	List of features for Shortform Identification and Categorization.....	13

Table of Contents

Honours Year Project Report	i
Abstract	ii
Acknowledgements	iii
1 Introduction.....	1
1.1 Background.....	1
1.2 What is a Shortform?	2
1.3 Motivation for Shortform Identification and Codec	3
1.4 Difficulties in Shortform Identification and Codec	5
2 Related Works.....	7
2.1 Acronym Detection.....	7
2.2 Maximum Entropy Modelling	8
3 Shortform Identification and Categorization.....	10
3.1 Identification and Categorization Model	11
4 Shortform Decoding	15
4.1 Shortform Decoder System Architecture.....	15
4.2 Phonetic Similarity Measurement.....	16
4.3 Phonetic Model.....	19
4.4 Lexical Similarity Measurement.....	20
4.5 Letter Omission Model.....	22
5 Shortform Encoding	25
6 Experiments.....	27
6.1 Definition of Evaluation Metrics	27
6.2 Individual Model and Overall Decoder Evaluation	28
7 Conclusion and Future Work	31
8 References.....	32
Appendix A – List of CMUDICT Phonemes.....	1
Appendix B – Extract from CMUDICT	1
Appendix C– List of Ranges for Features	1

1 Introduction

1.1 Background

SMS service has developed rapidly since its introduction and is very popular throughout the world. In 2001, more than 250 billion SMS were sent, comparing to the 16 billion sent in 2000. It is particularly popular amongst young urbanites as it allows for voiceless communication, useful in noisy environments (for instance, bars) that would defeat a voice conversation, and also buffered communication since the message the sender wants to convey can be accessed by the receiver any time.

Because of the limited message lengths and tiny user interface of mobile phones, SMS users commonly make extensive use of abbreviations and shortforms, particularly the use of numbers for words (for example, "4" in place of the word "for"), and the omission of vowels, as in the phrase "txt msg" which actually stands for "text message". This causes SMS to be credited with creating a language. Daily Telegraph, a newspaper in England once reported in 2003, 'Girl writes English essay in phone text shorthand'.

While predictive text software that attempts to guess words (AOL's T9) or letters (Eatoni's LetterWise) reduces the labor of time-consuming input, abbreviations of words are still popular, especially with the younger users where SMS shortforms forms a part of their culture.

There have been a few companies creating SMS lingo translation software such as Geneva Software Technologies Limited (GSTL). This SMS culture has also spawned websites such as Canada's transl8it.com which offered SMS lingo translation services through direct matching and also prompted collaborations amongst service providers and translation companies such as Singapore's GistXL Pte Ltd to come up with software such as GistXL, an English (or Singlish) to Simplified Chinese SMS translation platform embedded in SingTel's Singapore network, resulting in a GistXL win of a Merit Award in the recent Singapore National Infocomm Awards held last year (2004).

1.2 What is a Shortform?

SMS text messages are actually an amalgam of actual words, acronyms, abbreviations and shortforms. For purposes of discussion, we define word tokens as a consecutive sequence of characters in SMS text messages delimited by whitespace. In addition, we define the abovementioned terms below.

I define shortform as a shortened version of a single word. It may have been shortened through a process of truncation, omission of letters, or substitution of chunks of consecutive letters in a word with a shorter chunk of consecutive characters that are phonetically equivalent. Below, we see a few examples of shortforms:

Table 1: Examples of Shortforms

SHORTFORMS EXAMPLES				
Shortform	frm	b4	hv	Goin
Longform	from	before	have	Going

To make the distinction of shortforms from words that constitute standard English, we define actual words as follows:

Actual words are defined as words that are recognized as part of the English Language and can be found in an English dictionary or lexicon.

In this thesis, we also use the term *longform* for short form translations:

Long form are defined as shortforms translated into a lengthened version that is an actual word.

The definitions of *acronyms* vary from different sources. In this paper, for the purposes of giving a clear representation of acronyms in SMS text messages, we adopt the following definition of *acronyms*.

Acronyms are defined as:

- a collection of consecutive letters formed through shortening of phrases, termed by either taking the first letters of each word in a phrase or parts of certain words in a phrase. Examples include *PAC* for *political action committee*, *WAC* for *Women's Army Corps*, *radar* for *radio detecting and ranging*

It is common in SMS text messages to make up acronyms from several words and rely on the other person's cognitive processes to interpret it. Common examples include "btw" for "by the way" and "lol" for "laughing out loud". Many of such acronyms have become universally recognized in SMS text messages.

1.3 Motivation for Shortform Identification and Codec

Distinction between shortforms from acronyms is necessary because this thesis is primarily concerned with shortforms, with each shortform expanding into one single actual word.

As such, we exclude acronyms from the set of shortforms that we want to expand. The primary reason for this exclusion is that acronyms are possibly obtained from several words and the nature of acronyms implies existence of several possible expansions for each acronym.

Examples are as follows:

Examples:

ACL

- Association for Computational Linguistics
- Atlantic Container Line
- Advanced Computing Laboratory

BCG

- Boston Consulting Group
- Brockman, Coats, Gedelian (BCG Systems Inc.)

These acronyms essentially cannot be generically translated in the context of information poor SMS messages which often does not include the expansion of acronyms anywhere in them.

Shortforms, on the other hand, often can be translated into actual words. Occurrences of shortforms in SMS text messages often results from orthographic transformations in the forms of letter omissions, word truncations and substitution of parts of words with phonetically similar letter sequences. Thus, it is helpful to categorize shortforms in the ways that they are formed in order to gain insight for their translation.

Tables 1 to 3 below show examples of the 3 types of shortforms.

Table 2: Letter Omission (Shortforms) Examples

SHORTFORMS (LETTER OMISSION) EXAMPLES				
Shortform	frm	shld	hv	yr
Longform	from	should	have	your

Table 3: Truncation (Shortforms) Examples

TRUNCATION (SHORTFORMS) EXAMPLES				
Shortform	Swimming	goin	jus	ar
Longform	Swimming	going	Just	are

Table 4: Phonetic Substitutions (Shortforms) Examples

PHONETIC SUBSTITUIONS (SHORTFORMS) EXAMPLES				
Shortform	4	w8	wif	l8r
Longform	four	wait	with	later

As stated in *Background*, shortforms in SMS text messages are very popular for a variety of reasons. However, not everyone can form the same cognitive links and we find that some people either new to SMS have difficulties interpreting the messages. There are also people who have an

aversion to shortforms and would prefer the SMS messages in long form. This motivates the need for an SMS decoder.

People would also like to be able to shorten their SMS messages so as to reduce the number of messages sent in conversations. Therefore, it would be useful to have a coder that can reduce the length of an SMS messages through a lossless transformation in terms of information encoded.

1.4 Difficulties in Shortform Identification and Codec

The problem of shortform identification lies in distinguishing shortforms from acronyms. This problem exists because in addition to that fact that unlike actual words, shortforms and acronyms do not have the luxury of standard lexicons to differentiate between themselves.

Acronyms in SMS text messages often exist in a form different from those in traditional texts such as news articles, scientific journals, conference papers etc... Acronyms in traditional texts often occur near their expanded definitions.

For example:

The results of the survey, covering 40 major airports around the globe last year, were issued by the Airports Council International (ACI) and the airlines body, the International Air Transport Association (Iata).

This enables references to be drawn between the observed words near the acronyms and the acronyms themselves. Currently, there are several systems relying on such references.

(Please see *2.1 Acronym Detection*)

However, acronyms in SMS text messages often do not have such qualities. Acronyms in SMS text messages often occur on its own in between shortforms and actual words without any of their expanded forms near them.

For example:

btw do u want to go zouk this fri ? I m sick of studyin in *SOC* everydae.
Heard u got a *HP* interview?

It is also difficult to translate shortforms since the receiver of a message may not know what the shortform refers to in the first place.

Shortforms are formed in a diverse variety of ways such as omission, truncation and phonetic substitution. If the receiver does not know how the sender of the messages is encoding the actual words in messages into shortform, the translation is entirely up to how the receiver interprets the shortforms to decipher the message.

2 Related Works

As far as I know, there has been no work done on identification, encoding and decoding of shortforms in SMS messages. For full text sentences, there has been much work on detection of acronyms but due to the nature of the techniques involved, they are not suitable for application to the domain of SMS messages.

In this section, I present related research on acronym detection and introduce maximum entropy, a machine learning technique that I have used in my proposed model for shortform identification and codec.

2.1 Acronym Detection

Acronym Identification and detection has been much researched. Systems such as *Acrophile* (LS Larkey, P Ogilvie, MA Price, B Tamilio, 2000) have been created that automatically searches acronyms from acronym-expansion pairs from domain specific databases. By acronym-expansions pairs, we refer to a pairs each containing acronyms and their full expanded form or meaning.

It was suggested that of compression could be used in identification of acronym definitions (Yeates, Stuart, Bainbridge, David, and Witten, Ian, 2000). In their work, they formulated a coding scheme (the acronym model) which encodes each acronym as a function of the text present in the surrounding text-window and focuses solely on identifying acronym-meaning pairs. In their work, their system considers an acronym-meaning pair valid if the acronym model compresses well. In other words, the number of bits required to encode the acronym using the acronym model is less than the number of bits required to encode it with a general purpose compressor.

However, such approaches are not applicable to the domain of SMS messages where the topics are diverse and there is a lack of a database with standard acronym-definitions pairings. This is

simply due to the nature of SMS messages where the acronyms may be whimsically created. The SMS text messages also commonly do not have the expansions of the acronyms.

2.2 Maximum Entropy Modelling

Maximum Entropy, first introduced to the field of Natural Language Processing (NLP) by Berger, S Della Pietra and V Della Pietra (1996), has seen intensive research in the application of the technique to a variety of NLP related work such as Named Entity Recognizers, Part of Speech Tagging (POS) and others. This thesis proposes the application of Maximum Entropy machine learning in a variety of issues in this project.

Maximum Entropy machine learning framework allows the estimation of a function that determines the best choice from a series of outcomes given the extracted features of the input.

For example, in the Shortform Identification and Categorization Model, we can view the categorization of word tokens into actual word, acronyms and shortforms as a classification problem in which the objective is to estimate a function

$$f: W \rightarrow C$$

which maps an object $w \in W$ to its correct class $c \in C$.

Where

W is the set of words and sentence,

C is the set of tags for acronyms, actual words and the various shortforms.

$C = \{acronym_tag, actual_word_tag, sf_letteromission_tag, sf_phoneticmodel_tag, sf_truncation\}$

This function, f , can be implemented with a conditional probability distribution $p(y | x)$ where x is the context and y is the class. The task of Maximum entropy modelling is to choose the distribution p most indicative of this conditional probability with the limited empirical evidence available by weighting the various qualities of x .

The class of x is chosen based on the highest $p(y | x)$ given x :

$$Class(x) = \arg \max_x p(y | x)$$

The principle of maximum entropy modelling is simple: model all that is known and assume nothing about that which is unknown. In other words, given a collection of facts, choose a model which is consistent with all the facts, but otherwise as uniform as possible. The probability distribution that satisfies the above property is the one with the highest entropy. To train this model, a corpus of training examples in the form of tuples, $\tau = \{(o_1, c_1), \dots, (o_n, c_n)\}$ where $c_1 \dots c_n$ are the contexts with their corresponding outcomes $o_1 \dots o_n$, is required.

The application of maximum entropy in generation of models for predicting the translation of SMS shortforms is motivated by previous successful adaptation of maximum entropy modelling to natural language processing areas such as part of speech tagging (Ratnaparkhi, 1998) through the use of comparatively knowledge poor features.

Maximum entropy's accuracy in predicting a set of outcomes based on a set of comparatively knowledge poor features is useful for application to the domain of SMS shortform prediction. The primary reason is that SMS text messages hold no standard grammatical structure and does not show any apparent application of linguistic rules. Thus, the flexibility of maximum entropy in implementing constraints in the translation models based on features extracted from lexical and phonetic representation makes it suitable for the SMS related classification and translation tasks. This project uses the implementation of maximum entropy called `opennlp.maxent`¹ package.

¹ `opennlp.maxent` Package version 2.3.0 <http://maxent.sourceforge.net>

3 Shortform Identification and Categorization

In order to find the most suitable way to translate the shortform into longform, we have to first identify the shortforms in the SMS text messages. In addition, we also categorize the shortforms by the ways in which they are formed. For example, “ard” is actually “around” omitting “oun”, “get” is “forget” through substitution of the prefix “for” for a phonetically similar “4” and “somethin” is just “something” with truncation of the last letter.

This paper proposes a Shortform Identification and Categorization model based on maximum entropy to identify shortforms from actual words and acronyms/abbreviations and categorize the shortforms into the shortforms formed from letter omission and those formed through phonetic substitution of parts of words. Figure 1 below shows the overview.

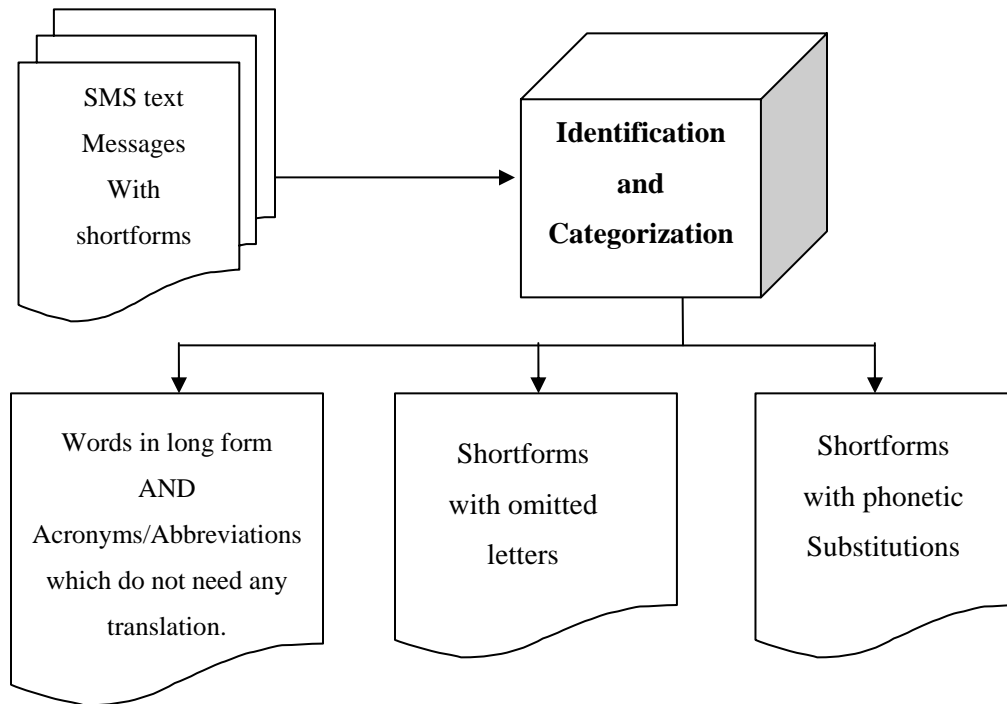


Figure 1: Shortform Identification and Categorization Model Overview

Table 5: Table of Categories

TAGS	CATEGORIES
W	Actual Word
A	Acronyms
P	shortform formed through phonetic substitution
O	shortform formed through omission of letters
T	shortform formed through truncation

Categorization is applied to each word token. When given a category, we either pass the word token straight to the output or through the other models (see *Section 4 Shortform Decoding*) for transformations or translations.

3.1 Identification and Categorization Model

The difficulties in shortform identification have been discussed in *Section 1.4 Difficulties in Shortform Identification and Codec*. In identification and categorization of shortforms, we select a set of generic features that are applicable to all the possible word tokens. In this section, we discuss the makeup of this feature set.

We extract a list of actual words, W , from the *Carnegie Mellon University pronouncing dictionary* (CMUDICT) and make use of W to collect data on the frequencies of each possible alphabetic bigram. Each bigram in each word token in an SMS text message is scored by its frequency. Bigram statistics are collected as follows:

$$BigramSum_{wordToken} = \sum_b \frac{Count(b)}{TotalBigramCount \times WordTokenLength}$$

$$Bigramproduct_{wordToken} = \prod_b \frac{Count(b)}{TotalBigramCount \times WordTokenLength}$$

$$BigramMax_{wordToken} = \arg \max_b (Count(b))$$

$$BigramMin_{wordToken} = \arg \min_b (Count(b))$$

where

$BigramSum_{wordToken}$ refers to the sum of bigram frequencies in a word token normalized by the word token length,

$Bigramproduct_{wordToken}$ refers to the product of bigram frequencies in a word token normalized by the word token length,

$BigramMax_{wordToken}$ refers to the maximum bigram frequency in a word token and

$BigramMin_{wordToken}$ refers to the minimum bigram frequency in a word token.

$Count(b)$ refers to the number of occurrence of bigram b . and

$TotalBigramCount$ refers to the total number of bigrams occurrences counted in W

A collection of word tokens, made up of a mixture of actual words, acronyms and various shortforms are compiled. The bigram statistics defined above are calculated for each word token and then normalized over the number of bigrams (length of word token-1). These normalized bigram statistics are then which given a rating based on the range on the frequencies. The resultant ratings are to be encoded as features for training a maximum entropy model.

In addition to the ratings mentioned above, we have other related features such as the ratings for the frequencies of the starting and ending bigrams of word tokens, the relative positions of the maximum and minimum bigram frequencies in a word token. These are included to give indication of the nature of bigram distribution throughout a word token. There are also other similar statistics. All of these are tabulated in *Table 6: List of features for Shortform Identification and Categorization*.

In addition to bigram statistics, we need to account for the case of the letters because the presence and distribution of uppercase letters in a word token help to determine if a word token may be an acronym. As such, we include also the percentage of uppercase letters in a word token as a feature. Other miscellaneous features include the word token length and features that indicate the presence of digits as well as dictionary features to indicate whether the word token can be found in a dictionary. The list of features encoded is shown below. It is important to note that all the features encoded are actually ratings assigned based on the range of the values. For a list of the range of the ratings, please see *Appendix C– List of Ranges for Features*.

Table 6: List of features for Shortform Identification and Categorization

SHORTFORM IDENTIFICATION AND CATEGORIZATION FEATURES	
Bigram Statistic-Related Features	
<i>BSr</i>	Normalized sum of all bigram frequencies in a word token.
<i>BPr</i>	Normalized Product of all bigram frequencies in a word token.
<i>BMxr</i>	Maximum bigram frequency in a word token.
<i>BMnr</i>	Minimum bigram frequency in a word token.
<i>BgDiff</i>	Difference between the maximum and minimum frequencies
<i>BgSr</i>	Frequency of starting bigram in a word token
<i>BgEr</i>	Frequency of ending bigram in a word token
<i>PosBMx</i>	Relative position of bigram with max freq in word token
<i>PosBMn</i>	Relative position of bigram with min freq in word token
Features related to case, digits and symbols	
<i>PcUr</i>	Percentage of word token in Uppercase characters
<i>PcDr</i>	Percentage of word token in digits
<i>Pos1stD</i>	Relative position of 1 st digit in word token
<i>PoslastD</i>	Relative position of last digit in word token
<i>PcSy</i>	Percentage of word token in Symbols (exclusive of punctuation)
<i>Pos1stSy</i>	Relative position of 1 st symbol in word token
<i>PoslastSy</i>	Relative position of last symbol in word token
Dictionary based and Other Features	
<i>Wx</i>	The word token exists in a dictionary.
<i>!Wx</i>	The word token does not exist in a dictionary.
<i>TkLen</i>	Length of word token.

For word tokens which are surrounded by other word tokens, the context of the features are extracted from the immediate surrounding word tokens and added to the predicates for classification.

The process of training the identification and classification model involves the use of 1000 examples complete with contextual predicates generated from word tokens using the features

highlighted above. The result of the training is a Maximum Entropy model with weights for each feature that classifies and tags word tokens given the word token itself and its context in the form of contextual predicates.

The contextual predicates for a word token are generated by a Context Generator based on the features given above in *Table 6: List of features for Shortform Identification and Categorization*. The contextual predicates are passed into the trained classification model to predict the tag for the word tokens. An overview of the whole process is shown below in *Figure 2: Overview of Classification Process*.

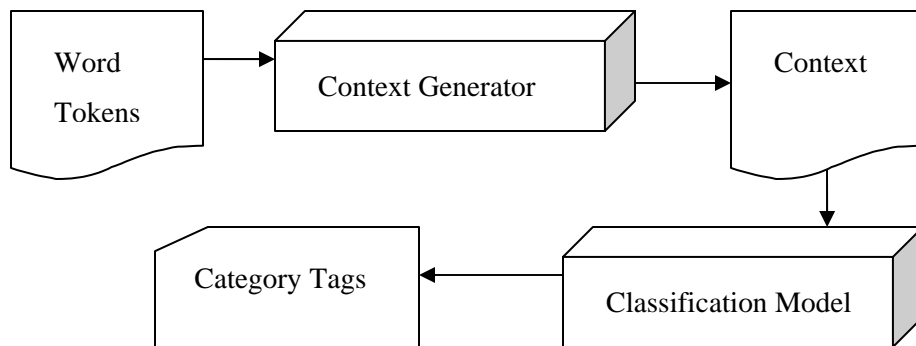


Figure 2: Overview of Classification Process

4 Shortform Decoding

As discussed in *Section 3 Shortform Identification and Categorization*, due to the restrictions on the length of a single SMS text message, SMS text messages are known to exhibit various qualities such as omission of letters, character substitution of syllables and truncation.

The different categories of shortforms warrant different models for their translation. This is because the cognitive process behind the encoding of each type of shortforms is different. For example, for phonetic substitution of syllabi, a person will be searching for something that sounds alike even though the resulting shortform is different in spelling while, for shortforms from letter omissions, the person will be looking at which letters can be omitted while retaining enough letters in the shortform for a person to recognize it.

In this section, we discuss the methodology behind decoding shortforms and the system architecture of our implementation.

4.1 Shortform Decoder System Architecture

In *Section 3 Shortform Identification and Categorization* above, we introduced shortform classification as a prelude to shortform decoding. In this section, we discuss our proposed methods for shortform decoding in greater detail.

We obtain the classified shortforms from categorization model. With these classified shortforms, we pass the shortforms with omitted letters and truncated shortforms into the Letter Omission Model and the shortforms with phonetic substitution into the Phonetic Model. The 2 models then translate the shortforms and the longform output is added to its location among the actual words and acronyms.

An overview of the architecture is shown below.

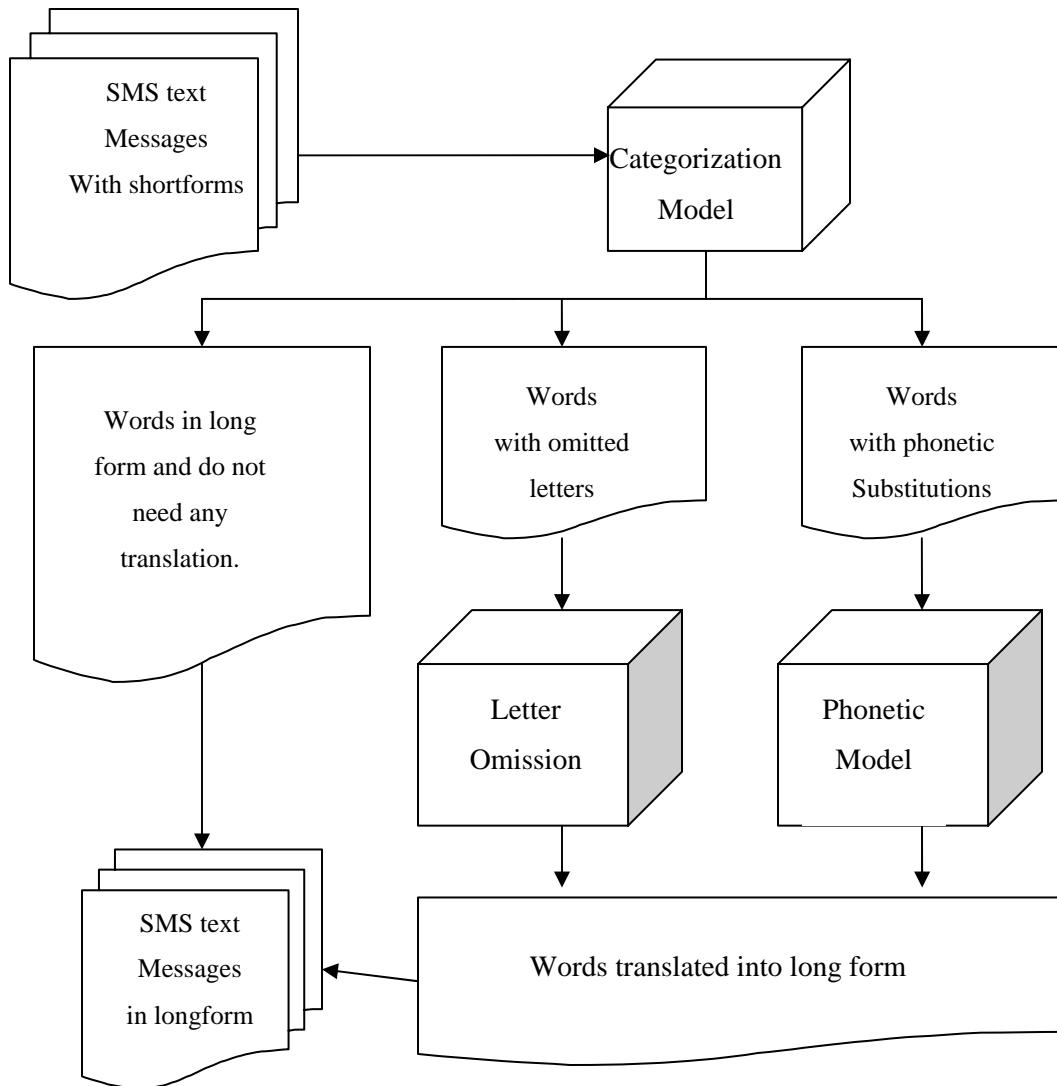


Figure 3: Shortform Decoder System Architecture

4.2 Phonetic Similarity Measurement

Word similarity can be measured on 3 different levels. They include sounds, graphemes and phonemes. Similarity on the grapheme level was explored by Soundex (Knuth, 1973). This paper does not consider physical sounds because the subject of the codec itself is text messages from short messaging service (SMS).

For translation to phonetic level, we use the CMU pronouncing dictionary (CMUDICT) to map parts of words to phonemes. CMUDICT is a pronouncing dictionary from Carnegie Mellon University that houses a lexicon that maps words to their corresponding phonemes. A list of the phonemes used in CMUDICT and an extract are found in the Appendix. Below, we present examples of such phonetic mapping.

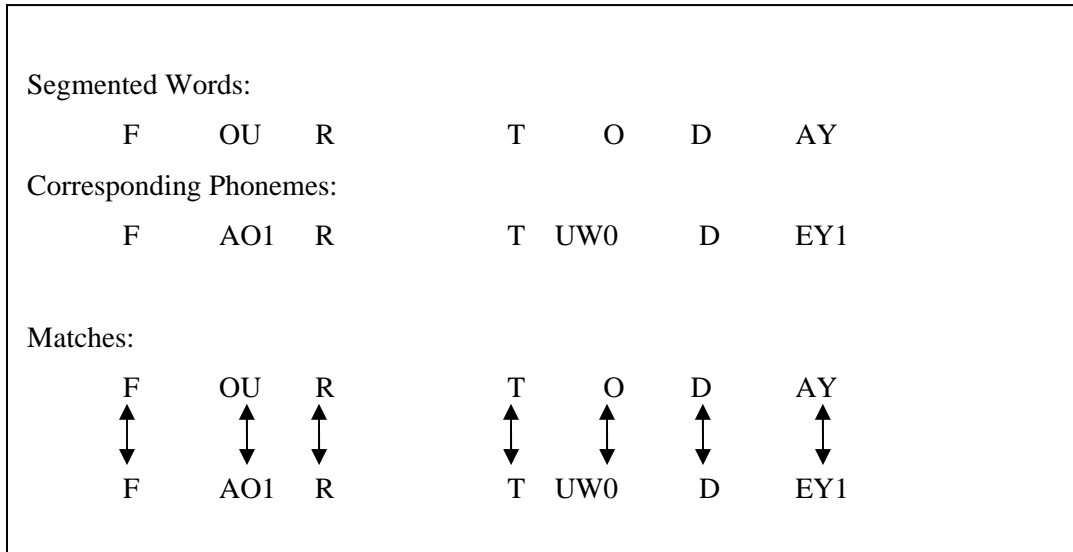


Figure 4: Examples of phonetic mapping

With this set of mappings from word segments to phonemes, we are now able to translate words into phonemes for similarity matching.

Translation from lexical grapheme level to phoneme level involves the process of segmenting the text by identifying the individual letters that map to single letter phonemes and tagging them as ‘islands’ and the word segments between them as ‘seas’. We recurse through possible segmentation configurations for the ‘seas’ and find the optimal configuration for which we can get the optimal bigram score. The scoring is done by summing the frequency of the mappings of the individual segments to the phonemes for each configuration and selecting the greatest.

With the segmented text, our next aim is to find the phonetic translation. We let the phonemes be the outcomes and each word segment and segment bigram be features. An example is illustrated in the following example.

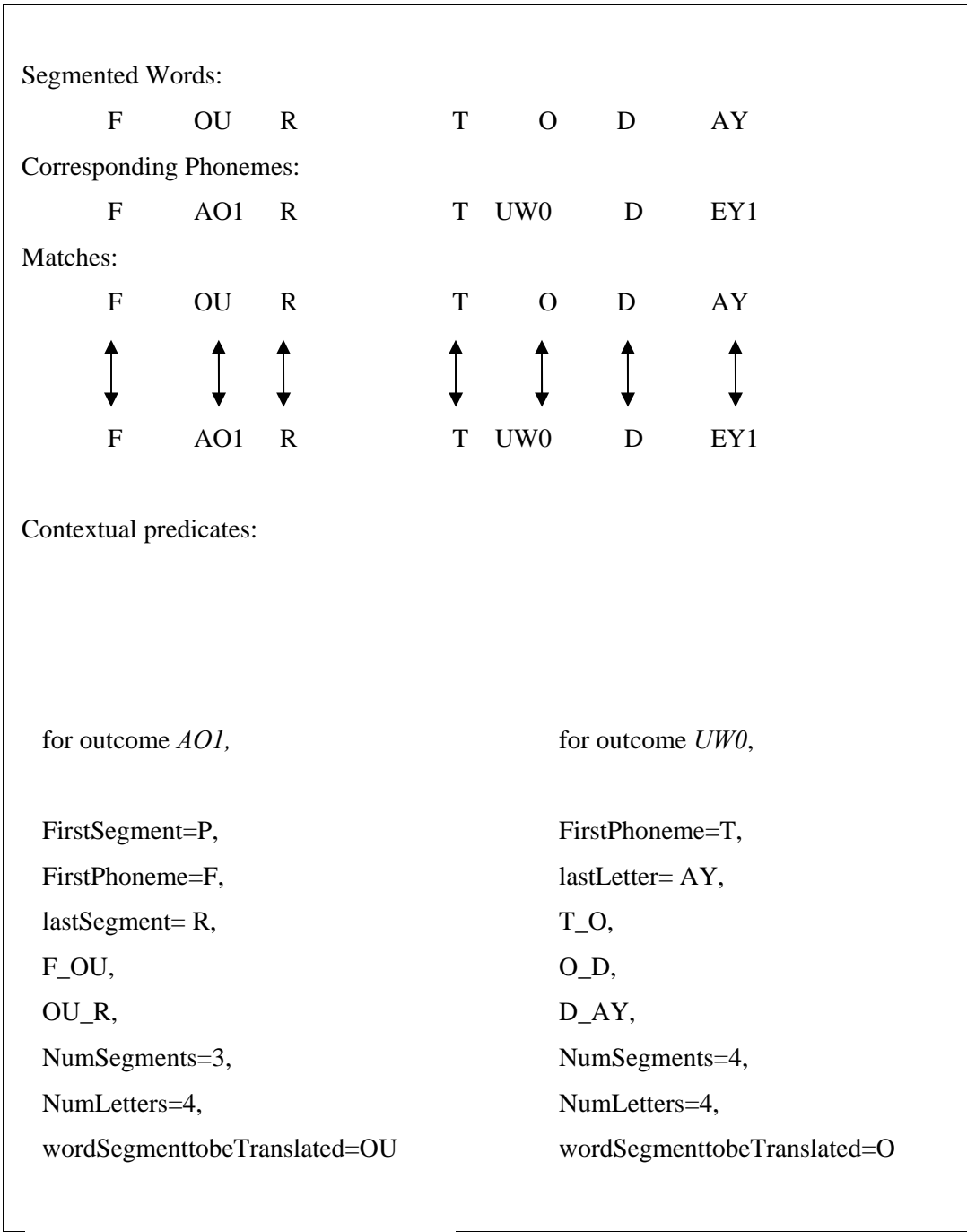


Figure 5: Example of context predicates for text to phonetic translation

4.3 Phonetic Model

We use the training examples of which contextual predicates and their outcomes are generated (Examples in Figure 5:) to train a text to translation model for the purpose of translating SMS text to phonemes. Using the same examples, we train another model for translating contextual predicates involving phonemes to text.

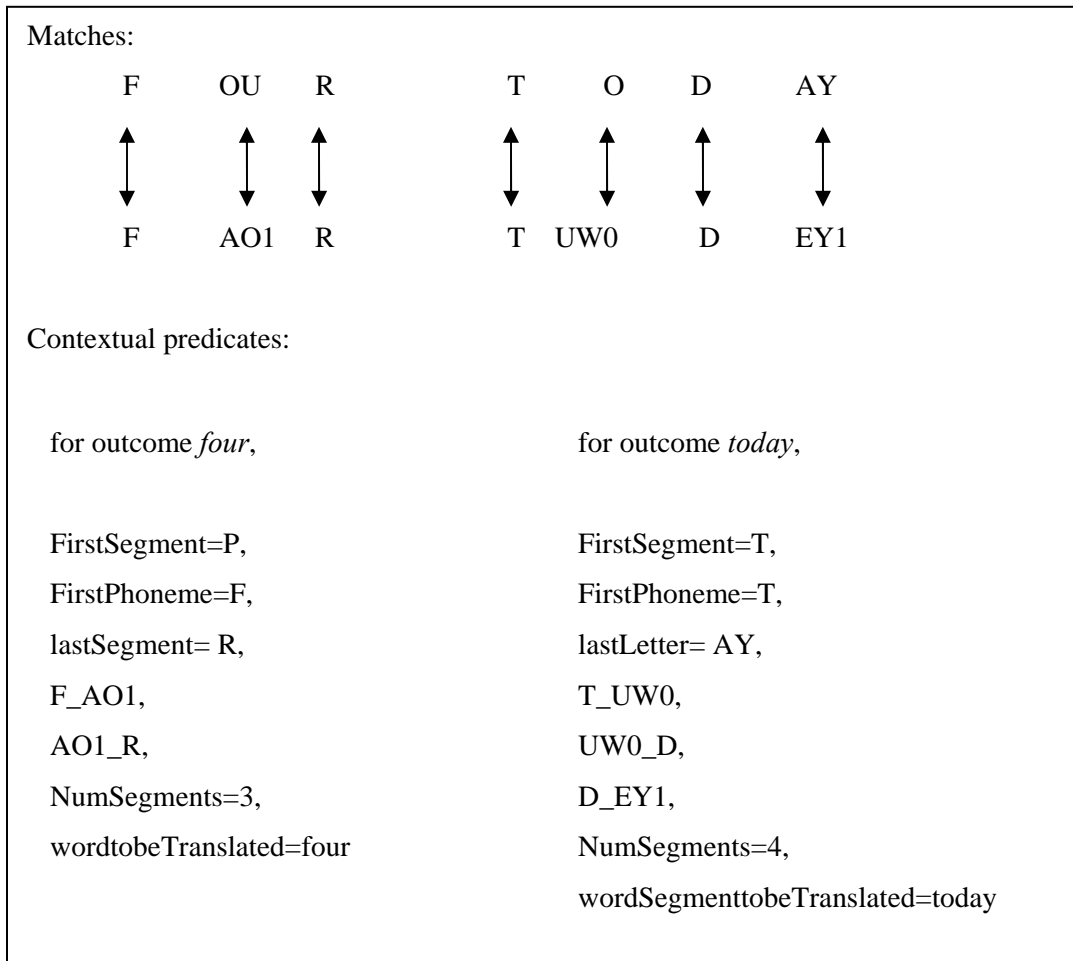


Figure 6: Example of context predicates for phonetic to word translation

Thus the process of matching a shortform involves translating it first into a phonetic representation and then matching the phonetic representation to the actual word using maximum entropy models.

An overview of the process is shown below:

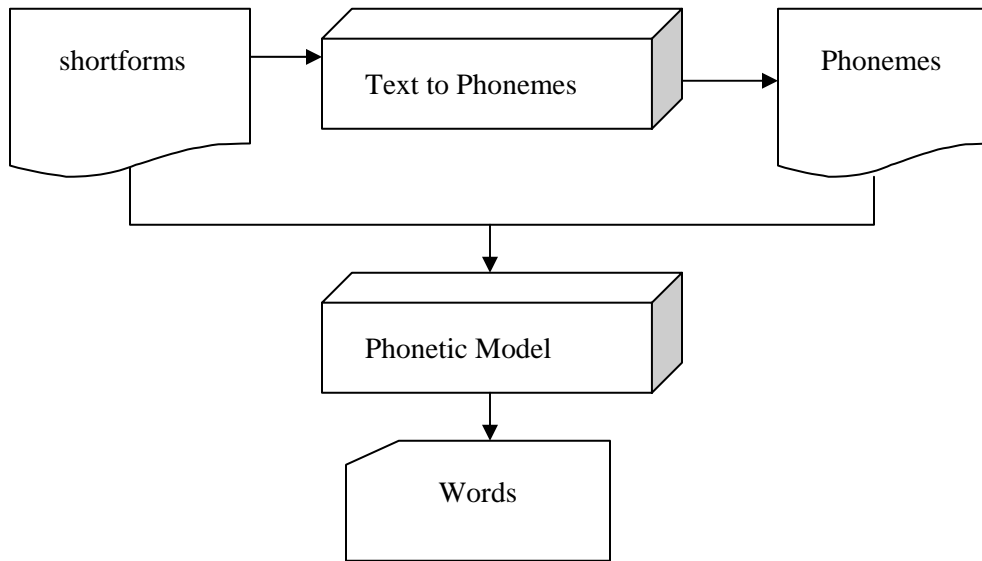


Figure 7: Overview of Process of Translating shortform(phonetic substitution)

4.4 Lexical Similarity Measurement

In order to infer the rules for omission, we need to find the difference between the original word and the transformed word that has some of its letters omitted. In other words, we have to find the orthographic similarity between the original word and the transformed word. For this purpose, we obtain a corpus of actual words and their corresponding shortforms as seen in SMS text messages.

Text	-	txt
What	-	wat
Should	-	shld
From	-	frm
Have	-	hv
Your	-	yr

Figure 8: Examples from corpus of actual words and corresponding shortforms

The Letter Omission Model implements an algorithm to find the longest common subsequence (LCS) between two words using dynamic programming. In this model, we used the LCS algorithm on the corpus discussed above. By getting the longest common subsequences for each pair, we can trace out the differences between the pair.

For example,

Actual word		Shortform
<u>f</u> <u>o</u> <u>r</u> <u>w</u> <u>a</u> <u>r</u> <u>d</u>	-	<u>f</u> <u>r</u> <u>w</u> <u>r</u> <u>d</u>
<u>l</u> <u>o</u> <u>v</u> <u>e</u>	-	<u>l</u> <u>u</u> <u>v</u>

The longest common subsequences are frwrd and lve.

So the transformations are:

forward:		love:
o → ε		o → u
a → ε		e → ε

Like the Shortform Identification and Classification Model, training the Letter Omission Model requires contextual predicates encoded from the input shortforms and its surrounding context.

For each space in between segments and for each segment, we extract contextual predicates from the space/segments and its surrounding segments. Contextual predicates extracted include the immediate surrounding segments of the space and the first and last letters of the shortforms. In the case where surrounding word tokens are present, the classification of word tokens are incorporated as contextual predicates as well.

This gives an overview of the shortform and enables us to model the constraints based on human interpretation and translation using the clues in the shortform itself.

The extraction of contextual predicates from the context of individual shortforms is illustrated below in *Figure 10: Contextual Predicates Extraction examples for segmented shortforms*.

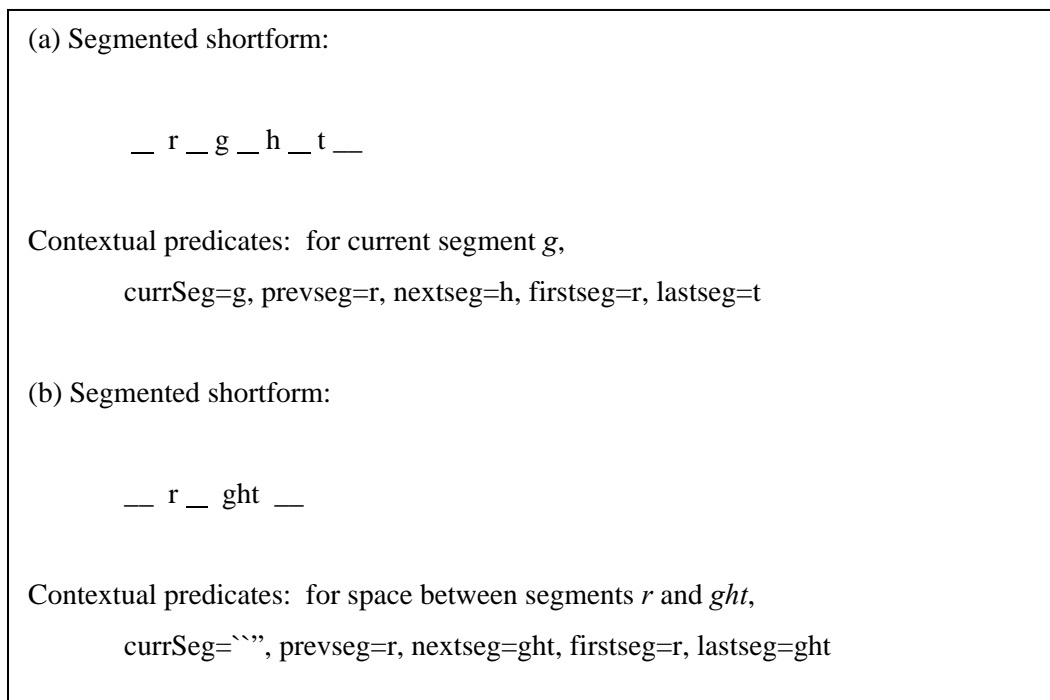


Figure 10: Contextual Predicates Extraction examples for segmented shortforms

A set of contextual predicates along with the resulting outcome are used to train the letter omission model. For example, from *Figure 7*, the right segmentation is (b). The correct transformation in this example is *right* \rightarrow *right* . The extraction of contextual predicates for the correct training example is shown below.

Segmented shortform:

r ght

Contextual predicates for *right* \rightarrow *right* ,

currSeg="", prevseg=r, nextseg=ght, firstseg=r, lastseg=ght i

where the last token *i* is the outcome. In other words, the above represents:

Figure 11: Contextual Predicates example for training example for Letter Omission

5 Shortform Encoding

In this section, we discuss the methodology behind our implementation of shortform encoding. As opposed to shortform decoding, shortform encoding refers to encoding proper English with correct grammar structure and actual words into a compacted shortened form where the grammatical structure and spelling might be compromised.

Text summarization and text reduction are inherently lossy processes since both necessarily involve decisions about what elements of a document can safely be omitted. (Simon Corston-Oliver, 2001) In this section, we look at how we can omit letters in words such that we can retrieve the word from its compacted form.

To decide whether or not to drop letters, we can use the machine learning framework to train an encoding model that learns when and where to drop letters.

Using essentially the same implementation for detecting the possible letter additions to shortforms in shortform decoding, we can use the same maximum entropy framework in shortform decoding for shortform encoding by viewing the omission process as a classification problem where we view each letter as possible candidates for omission and classify them as being either suitable for omission or essential in ensuring the original word can be obtained from the resulting shortform.

In this case, however, we swap the input files for actual words and shortforms so that machine learner is actually learning the rules on when and how to transform actual words into shortforms.

An example is shown below.

Contextual predicates for $right \rightarrow rght$,

$currSeg=i, prevseg=r, nextseg=ght, firstseg=r, lastseg=ght$ ""

where the last token "" is the outcome. In other words, the above represents:

$i \rightarrow \epsilon$

Figure 12: Contextual Predicates example for training example for shortform encoding

A set of the input training examples for this encoder can be obtained by swapping the order of the actual words and corresponding shortforms from the training examples from the Letter Omission Model (see Section 4.5).

In addition, we can further increase the flexibility of the system by incorporating the phonetic transformations from the Phonetic Model to form a combined superset of all the actual words and shortforms from the letter omission and the phonetic models. For example,

Contextual predicates for $b4 \rightarrow before$,

$currSeg=b, prevseg=""$, $nextseg=4, firstseg=b, lastseg=4$ be
 $currSeg=4, prevseg=b, nextseg=""$, $firstseg=b, lastseg=4$ fore

where the last tokens, *be* and *fore*, are the outcomes. In other words, the above represents:

$b \rightarrow be$ and $4 \rightarrow fore$

Figure 13: Contextual Predicates example adapted from Phonetic Model

6 Experiments

In this section, we focus on the evaluation and performance of the system. We first define the evaluation metrics to be used, followed by the evaluation of the performance of each individual model and then conclude with an evaluation on the integrated system.

For purposes of evaluation, we extract a subset of 2500 SMS text messages from the corpus of 25000 SMS text messages collection we have collected and use it as our standard input for each evaluation. Using human manual labour, we produce a classification-tagged and corpus in long form to produce a pair of SMS parallel corpora of which we set the translated corpus as the gold standard.

6.1 Definition of Evaluation Metrics

In comparing the results of the models presented in this paper, we use a common metric in machine translation. We define precision, recall and the F measure as follows:

$$\textit{Precision} = \frac{A}{A+B}$$

$$\textit{Recall} = \frac{A}{A+C}$$

$$F = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

Where

In shortform identification and categorization,

With a category type t ,

A is the number of type t tokens correctly tagged t .

B is the number of tokens not of type t incorrectly tagged t .

C is the number of type t tokens incorrectly tagged as other types.

In addition to precision, recall and F measure, we define the following regarding token accuracy:

$$OverallTokenAccuracy = \frac{\sum correctTokens}{\sum alltokens}$$

$$UniqueTokenAccuracy = \frac{\sum correctUniqueTokens}{\sum alluniquetokens}$$

Where

OverallTokenAccuracy is the percentage of tokens categorized/translated correctly.

UniqueTokenAccuracy is the percentage of unique tokens categorized/translated correctly.

6.2 Individual Model and Overall Decoder Evaluation

Evaluation is done on each individual model, namely, Identification and Categorization, Letter Omission and Phonetic, to examine their performance for their specific tasks.

For evaluation of the Identification and Categorization Model, we pass the standard input of 500 SMS text untagged messages into the model and extract the resulting tags for comparison with the corresponding tagged and translated corpus of the parallel corpora.

The results are tabulated as follows:

To get some comparison of the results, we define a baseline model where it predicts the category “actual words” for all word tokens. This gives the baseline model a precision of 43.1%, a recall of 55% and F-measure of 49.05%.

In comparison, with reference to *Figure 14: Identification and categorization performance evaluation* below, the results are indicative of the following: Though, the identification and categorization model is not able to categorize correctly all word tokens, it gives significantly better categorization performance compared to the baseline. The difference of 29.15% could have been created if there were a greater percentage of word tokens that are not actual words. The unique token accuracy is lesser than the overall token accuracy. This is somewhat expected since there are less much common variants of shortforms, words, and acronyms around.

RESULTS	Identification & Categorization Model
<i>OverallTokenAccuracy(actual words)</i>	89.7%
<i>UniqueTokenAccuracy(actual words)</i>	63.0%
<i>OverallTokenAccuracy(acronyms)</i>	82.1%
<i>UniqueTokenAccuracy(acronyms)</i>	78.4%
<i>OverallTokenAccuracy(shortforms:phonetic)</i>	64.4%
<i>UniqueTokenAccuracy(shortforms:phonetic)</i>	45.3%
<i>OverallTokenAccuracy(shortforms:o t)</i>	74.4%
<i>UniqueTokenAccuracy(shortforms:o t)</i>	65.7%
Average weighted Precision	74.2%
Average weighted Recall	82.1%
Average weighted F-Measure	78.2%

Figure 14: Identification and categorization performance evaluation

For evaluation of both the Letter Omission and Phonetic models, we pass the untagged corpus of the pair of gold standard parallel corpora, through the two models. We then compared the results with the tagged and translated half of the corpora.

RESULTS	Phonetic Model	Letter Omission Model
<i>OverallTokenAccuracy</i>	73.7%	81.7%
<i>UniqueTokenAccuracy</i>	44.6%	68.3%

Figure 15: Phonetic and Letter Omission Model evaluation results

From the results, we can see that the two token accuracy metrics for the letter omission model is better than those for the phonetic Model. This is probably because the shortforms given phonetic substitutions have much wilder variants than their counterparts formed through letter omission.

RESULTS	Baseline Model	SMS Codec System
<i>OverallTokenAccuracy</i>	55%	74.6%
<i>UniqueTokenAccuracy</i>	14%	56.7%

Figure 16: Comparison between the baseline model and SMS Codec System

Referring to *Figure 16: Comparison between the baseline model and SMS Codec System* above, the SMS Decoder of the Codec System performs close to 20% better than the baseline model. This is especially so in an SMS corpus with greater frequency of shortforms.

With initial results, I find the main exceptions to the translation process of the Letter Omission Model are the shortforms that require double insertions. For example, the shortform *lv* that translates to *love* requires the insertion of letters *o* and *e*. Therefore, to drive the translation as close as possible to its corresponding actual word, I introduce an additional loop to the letter omission model where the translated word loops back to the categorization model which checks whether it can be classified as an actual word and passes to the letter omission model for an additional insertion if not. The additional condition is that this loop stops at the point where the probability of the translation being an actual word is highest.

With respect to the results of the final evaluation, it seems that system does well on certain shortforms given that the overall word accuracy is higher than the unique token accuracy. This came as no surprise as there can be very wild variations on the words involved. Certainly, a word can be shortened in many ways as it is.

7 Conclusion and Future Work

In conclusion, I have proposed the use of maximum entropy machine learning framework for SMS shortform identification that can categorize text into actual words, shortforms and acronym. This model does the latter without requiring the presence of acronyms-definitions pairs that is common in today's research. I have also proposed a model for decoding SMS shortform using a series of transformation and translation models based on maximum entropy. As far as I know, no research has yet to be done on such application using maximum entropy for such purposes.

In addition, in my knowledge, very little research has been done on SMS shortform identification and codec. Thus, future work could be devoted to fine-tuning the capture of rules for human shortform creation and in turn, infer the rules for shortform decoding. The present corpus of SMS parallel corpora can also be extended to facilitate future research.

8 References

- Serguei Pakhomov and Mayo Foundation, Rochester, MN (2002). Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 160-167. Microsoft
- Simon Corston-Oliver, Microsoft Research (2001). Text Compaction for display on very small screens. NAACL. 2001
- Shao, Li & Ng, Hwee Tou (2004). Mining New Word Translations from Comparable Corpora. Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004). (pp. 618-624). University of Geneva, Geneva, Switzerland.
- Chieu, Hai Leong, and Ng, Hwee Tou (2003). Named Entity Recognition with a Maximum Entropy Approach. Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003). (Shared Task Paper). (pp. 160-163). Edmonton, Alberta, Canada.
- Li Chieu, Hai Leong, and Ng, Hwee Tou (2002). Teaching a Weaker Classifier: Named Entity Recognition on Upper Case Text. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02). (pp. 481-488). Philadelphia, Pennsylvania, USA.
- Faulkner, X and Culwin, F (2001) SMS: Users and usage. People and Computers XV- Interaction without frontiers. Joint proceedings of HCI 2001 and IHM 2001. Vol. 2nd (Adjunct), pp. 73-79
- LS Larkey, P Ogilvie, MA Price, B Tamilio (2000). Acrophile: an automated acronym extractor and server. Proceedings of the fifth ACM conference on Digital libraries, 2000

- Yeates, Stuart (1999). Automatic extraction of acronyms from text. In Proceedings of the Third New Zealand Computer Science Research Students' Conference. Hamilton, New Zealand, April 1999, University of Waikato, pages 117-124.
- Yeates, Stuart, Bainbridge, David, and Witten, Ian. (2000) Using Compression to identify acronyms in text. Submitted to Data Compression Conference, DCC2000.
- A Berger, S Della Pietra and V Della Pietra (1996). A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics(22-1), 1996
- K Nigam, J Lafferty, A McCallum (1999). Using maximum entropy for text classification. IJCAI-99 Workshop on Machine Learning for Information, 1999
- Knuth, D.E. (1973). The art of computer programming; Volume 3: Sorting and searching. Addison-Wesley Publishing Company: Reading, Mass. Page 392.
- W. H. Lin and H. H. Chen (2002). Backward machine transliteration by learning phonetic similarity. In Proceedings of the Sixth Conference on Natural Language Learning (CoNLL), pages 139--145, 2002.
- ID Melamed, R Green, JP Turian (2003). Precision and Recall of Machine Translation. In Proceedings of the Human Language Technology and North American Chapter of the Association for Computational Linguistics Conference 2003 (HLT-NAACL 2003) page 1
- K Papineni, S Roukos, T Ward, WJ Zhu (2002). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association, 2002
- Y Yang, X Liu (1999). A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR, Conference on Research and Development in Information Retrieval, 1999
- The American Heritage College Dictionary, Third Edition. Boston: Houghton Mifflin Company, 1993.

Adwait Ratnaparkhi (1998). MAXIMUM ENTROPY MODELS FOR NATURAL LANGUAGE AMBIGUITY RESOLUTION. A DISSERTATION in Computer and Information Science. Ph. D. thesis, University of Pennsylvania, 1998

Appendix A – List of CMUDICT Phonemes

LIST OF CMUDICT PHONEMES		
PHONEME	EXAMPLES	TRANSLATION
AA	odd	AA D
AE	At	AE T
AH	Hut	HH AH T
AO	Ought	AO T
AW	Cow	K AW
AY	Hide	HH AY D
B	Be	B IY
CH	Cheese	CH IY Z
D	Dee	D IY
DH	Thee	DH IY
EH	Ed	EH D
ER	Hurt	HH ER T
EY	Ate	EY T
F	Fee	F IY
G	Green	G R IY N
HH	He	HH IY
IH	It	IH T
IY	Eat	IY T
JH	Gee	JH IY
K	Key	K IY
L	Lee	L IY
M	Me	M IY
N	Knee	N IY
NG	Ping	P IH NG
OW	Oat	OW T
OY	Toy	T OY
P	Pee	P IY
R	Read	R IY D
S	Sea	S IY

SH	She	SH IY
T	Tea	T IY
TH	Theta	TH EY T AH
UH	Hood	HH UH D
UW	Two	T UW
V	Vee	V IY
W	We	W IY
Y	Yield	Y IY L D
Z	Zee	Z IY
ZH	seizure	S IY ZH ER

Appendix B – Extract from CMUDICT

ABACHA AE1 B AH0 K AH0
ABACK AH0 B AE1 K
ABACO AE1 B AH0 K OW2
ABACUS AE1 B AH0 K AH0 S
ABAD AH0 B AA1 D
ABADAKA AH0 B AE1 D AH0 K AH0
ABADI AH0 B AE1 D IY0
ABADIE AH0 B AE1 D IY0
ABAIR AH0 B EH1 R
ABALKIN AH0 B AA1 L K IH0 N
ABALONE AE2 B AH0 L OW1 N IY0
ABALOS AA0 B AA1 L OW0 Z
ABANDON AH0 B AE1 N D AH0 N
ABANDONED AH0 B AE1 N D AH0 N D
ABANDONING AH0 B AE1 N D AH0 N IH0 NG
ABANDONMENT AH0 B AE1 N D AH0 N M AH0 N T
ABANDONMENTS AH0 B AE1 N D AH0 N M AH0 N T S
ABANDONS AH0 B AE1 N D AH0 N Z
ABANTO AH0 B AE1 N T OW0
ABARCA AH0 B AA1 R K AH0
ABARE AA0 B AA1 R IY0
ABASCAL AE1 B AH0 S K AH0 L
ABASH AH0 B AE1 SH
ABASHED AH0 B AE1 SH T
ABATE AH0 B EY1 T
ABATED AH0 B EY1 T IH0 D
ABATEMENT AH0 B EY1 T M AH0 N T
ABATEMENTS AH0 B EY1 T M AH0 N T S

Appendix C– List of Ranges for Features

List of Ranges for Categorization Features			
Bigram Statistic-Related Features		Features related to case, digits and symbols	
<i>BSr</i>	1-20	<i>PcUr</i>	1-10
<i>BPr</i>	1-20	<i>PcDr</i>	1-10
<i>BMxr</i>	1-20	<i>Pos1stD</i>	1-10
<i>BMnr</i>	1-20	<i>PoslastD</i>	1-10
<i>BgSr</i>	1-20	<i>PcSyr</i>	1-10
<i>BgEr</i>	1-20	<i>Pos1stSy</i>	1-10
<i>PosBMx</i>	1-10	<i>PoslastSy</i>	1-10
<i>PosBMn</i>	1-10		
Dictionary-based Features			
<i>Wx</i>	Binary feature. Either turned on or off.		
<i>!Wx</i>	Binary feature. Either turned on or off.		
Other Features			
<i>TkLen</i>	Range of actual length of word Tokens		