http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (7): 954-965, 2007
ISSN 1812-5638

Analyzing Statistical and Syntactical English Text for Word Prediction and Text Generation

Taher S.K. Homeed and Mansoor Al-A'ali
Department of Computer Science, College of Information Technology, University of Bahrain,
P.O. Box 32038, Kingdom of Bahrain

Abstract: This research describes a technique for word and phrase prediction to help the writer of an English text in text generation and to speed up the typing process. The technique consists of a learning phase and a generation or prediction phase. The learning phase learns the English phrase and sentence syntax as well as keeps all the corpora for reference against the syntax and the actual text. The prediction of the next word or phrase is based on the preceding one or more words and the history and frequency of occurrences of phrase mappings. This research establishes that word and phrase prediction based on the preceding word or words depending on statistical or frequency use of the words is enhanced by coupling it with syntax structures to ensure that a form of semantics and grammar correctness is maintained. The approach shows the way to learn both the words strings and the syntax structures which are later used for the prediction or generation purposes. We present results to demonstrate that our approach gives enhancements to key stroke reduction and a more accurate and grammatically correct prediction.

Key words: Word prediction, key stroke reduction, syntax structure, English language structure

INTRODUCTION

Word prediction background: The problem of word prediction boils down to finding a linguistically relevant mapping between the user input and system output (Foster *et al*., 2002; Tam *et al*., 2002; Potelle and Rouet, 2003; Gibler and Childress, 1982; Alvar *et al*., 2006; Diana and Jo-Anne, 2007). As user input, one or more word-initial characters, for instance, can be used in word completion, possibly matching a certain number of word tokens in the lexicon employed in the prediction system. Prediction methods aim to achieve accurate enough predictions. Prediction can be for single word prediction, phrases consisting of two or more word tokens prediction. That kind of approach to word prediction is justified, because the token frequency of phrases appears to be sufficiently high in English texts in addition to being capable of modeling text structure as an alternation between single words and phrases (Erman and Warren, 2000). The German FASTY system developed by Matiasek *et al*. (2002) is an example of a prediction system with multiple prediction methods.

Several prediction schemes such as those by Lois and Drawer (1998), Shieber and Baker (2003) and Ian and Stephen (2006), were presented since the nineties, for systems intended to facilitate text production for handicapped individuals. These schemes were based on single-subject language models, where the system is self-adapting to the past language use of the subject. Sentence position, the immediately preceding one or two words and initial letters of the desired word are cues which may be used by the systems.

Word prediction systems are normally based on phrase prediction and single word prediction. Word prediction performance is measured by savings on keystroke or character savings percentages. Various parameters of the prediction process affect the operation of a prediction system as a whole. Analysis of the performance of prediction techniques show that prediction methods may be improved.

Word prediction performance from the application user's point of view was discussed in Garay-Vitoria and Abascal (2005) and Lesher *et al*. (2002). They recommend the following three parameters to be defined in order to measure the performance of the prediction process:

- Savings = cost of using predictions_cost of entering standard events;
- Cost of using predictions = visual scan time for searching for predictions + cognitive time for deciding on The correct prediction + physical movement time required to select a prediction;
- Cost of entering standard events = cognitive time required to formulate the next event + physical movement required to enter the next event.

Corresponding Author: Taher S.K. Homeed, Department of Computer Science, College of Information Technology, University of Bahrain, P.O. Box 32038, Kingdom of Bahrain

All the measures above assume that only one prediction is offered on the list, i.e., the size of the prediction list is one token. Further, word prediction performance is usually quantified in terms of the number of keystrokes or characters which can be saved in the typing of text. However, the percentage of keystroke or character savings does not necessarily correlate very closely with other measures of performance, such as the time saved in the composition of the text by means of word prediction (Tina and Hunnicutt, 2002).

The performance of prediction systems with a phrase prediction are normally measured as:

- The selection of a suitable prediction technique for single word prediction and phrase prediction.
- The frequency of phrases in the predicted text.
- Prediction accuracy, which is the number of word and phrase predictions which are actually selected by the user.

Most word prediction approaches tend to focus on one or all of the above three parameters, although most reported results tend to be based on a small corpora of text and thus cannot be considered as a final solution to such a difficult problem. To our knowledge there is no report on the best prediction technique.

Another prediction issue which has not been considered in the literature is word or phrase filling, where the user will leave in the middle of the text a blank for the system to decide which should be predicted based on the previous and post phrases in relation to the blank left.

Word prediction systems can be seen as a form of text production. Word or phrase prediction is an attempt to reduce the number of keystrokes or characters needed to type the text. A good prediction approach is one that learns by updating the database of phrases each time a new phrase is generated as per the user selection.

Word prediction research: First word prediction systems were an aid for people suffering from various kinds of disabilities or difficulties in typing (Gibler and Childress, 1982; Raskind and Higgins, 1998; Lin, 2003, 2004). Nowadays, however, this is no longer the main field of use for word prediction systems (Boissie`re and Dours, 2002). Writing assistance systems based on word prediction have already moved from very small disabled population to the large and wide ordinary people (Boissie`re and Dours, 2002).

The main feature of word prediction is the fact that only the left part of the sentence completed so far by the user is available for the prediction process which should decide on what is the most fitting next word or phrase should be. The suggested or predicted word or phrase or groups of words or phrases must be grammatically fit in the newly to be formed sentence or phrase. Therefore for prediction to be correct and useful it must fit in the syntactic or grammatical context of the newly formed sentence. To our knowledge the reported prediction techniques did not consider the issue of grammar, syntax and semantics in the prediction process but are on the whole based on stored sequences which are selected regardless of their semantic, syntactic or grammar fitting.

Word prediction technology was initially designed to help individuals with text entry mobility disabilities. Word prediction is also useful for slow typists, pen or probe users and individuals with minor visual impairments or learning disabilities. These programs provided functionality to interpret the typist's grammar and guess what words would follow based on syntax, the first letter typed and frequency of use. Word prediction technology is particularly useful in assisting with text entry and provides a method for rate enhancement by reducing the number of keystrokes the user has to input. Recent studies have shown that word prediction technology can reduce keystroke by 50 to 60% (Dick, 2001).

Word prediction systems have a number of benefits: 1) help the users to improve their writing skills, 2) Help users with physical disabilities to write with less effort as the prediction can substantially reduce the number of keystrokes needed to write, 3) help users with learning difficulties to concentrate on the content of their work rather than struggling with the spelling correctness, 4) enable users to enhance the system by increasing the system's knowledge in words and sentences related to the subjects that they are interested in and 5) save users time and effort in thinking about the most appropriate word (phrase) that should come next.

While word prediction has been shown to have advantages for some users, it can also make the composition process more complex, which may slow the typist. First, the word list may distract the user, as the ever changing presentation draws the attention away from the sentence being constructed. The cognitive process of selecting words from a prediction list may disrupt the creative process and interfere with the flow of composition, especially for people who have significant difficulty in word recognition (Boissie`re and Dours, 2002; Diana and Jo-Anne, 2007; Lesher *et al*., 2002; Lin, 2003, 2004; Raskind and Higgins, 1998; Tam *et al*., 2002). Word prediction can provide too much of a good thing. Hunnicutt (2002) state that, as the number of predictions grow, the increase in keystroke savings diminish (i.e., the

keystroke savings plateau). As more predictions are listed, the user must inspect a longer word list and the chance of missing the desired word on the list increases. Some users may only look at the predictions at the top of the list and ignore the words lower on the list.

Use of a word prediction feature requires additional cognitive and perceptual processes and these are the major contributors to the increase in selection time (Horstmann and Levine, 1991). Visual search of the prediction list and the user's decision about whether the word is on the list are two of the processes researchers attributed to increased selection time. Cognitive and perceptual loads imposed on the user are associated with a time cost that can offset and even overwhelm the keystroke savings provided by word prediction.

According to Tam *et al.* (2002) individuals with lower intelligence scores and low scores in memory and reading may have trouble keeping track of their place on the copy material. Participants who reported no impairment in intelligence and memory also had difficulty keeping track of their place on the source document. The participants in this study reported that, because they had to look up from the source copy to scan the word prediction list, they often lost their place. The time saved due to keystroke reductions was outweighed by the loss of time to search for their place. In fact, for seven out of ten participants in this study, word prediction reduced the typing speed in direct proportion to the number of times it was used.

Eyas (2004) conducted a study which suggested using a cluster of computers to build an Optimal Binary Search Tree (OBST) that will be used for the statistical approach in word prediction. In his research he explains the difficulty of expressing the statistical model without losing any accuracy. Unfortunately, their research focused more on how to use the optimal binary searching rather than give any results or technique for word prediction.

Most word prediction systems employ statistical analysis for their word prediction. The choice of words for placement in the prediction list is based upon the probability that they will appear in the text. In the simplest systems, the probabilities only relate to isolated words and their likelihood of use. The easiest way of performing word prediction is to use a fixed lexicon, as is used by the software designed in this project. Each word in the lexicon will have a frequency associated with it which shows how often it is used in the text.

Two different methods are used to perform predictions. The first method sorts the whole lexicon based on the frequency order and offers the user a few words from the prediction list with the highest frequencies. The second method tags the words with the frequencies with which they appear after other given words. In this way when a word is entered, the most frequently used word which follows it can be extracted to produce a prediction list. This methods has an advantage over the first method that the prediction list will be much more on the current status of the sentence and therefore much likely to contain correct predictions.

N-gram models have been among the most successful approaches used for language modeling. These refer to finite state of analysis of a series of 1, 2 or 3 word sequences and are called unigram, bi-gram and trigram. Using a single computer would not offer the capability of building the trigram. Consider a vocabulary size V, then there will be V3 trigrams, for which 20,000 words translates to 8 trillion trigrams (Eyas, 2004).

Eyas (2004) used an Optimal Binary search tree, N-Gram, trigram to determine keystroke saving and the results show improved KSR (Keystroke saving rate). No indication of how practical methods of word prediction can be used, e.g., is it feasible on one computer.

FORMAL REPRESENTATION OF PREDICTING PROCESS

Predicting the next one, two, or three words depend totally on the database stored, what the user is writing and under what topic it is stored.

Here is a formal representation of predicting process:

Let W be the set of all words: $W = \{w_i\}$ where $I = 1$ to n
Let S be the set of all sentences: $S = \{s_i\}$ where $j = 1$ to m
And any sentence is composed of many words: $S_k \{w_i, w_j, w_m, \ldots\}$ where k, I, j, m = 1 to n and

$Sk \in S$ and $\forall$ wi $\in$ Sk, wi $\in$ W

Let R be the set of all paragraphs,

$\therefore R = \{ri\}$ where I = 1 to x

Let To be the set of all topics,

$\therefore To = \{ti\}$ where I = 1 to n

Wi $\in$ {tj…k} I≠k and j…k > = 1
That is:

$Sk \subseteq S \leftrightarrow \forall k \in W$ then $k \in S$; $W \subseteq S$; $S \subseteq R$; $W \subseteq R$

That is:

$W \in si$ and $W \in sj$ and $si \neq sj$

That is

$W \in Ti$ and $W \in Tj$ and $Ti \neq Tj$

Let Ow = {wi} be the set of one word
Tw = {wi, wk} be the set of two words
Hw = {wi, wk, wm} be the set of three words
Where I, k, m = 1 to n
Then $Ow \subseteq Tw$ and $Tw \subseteq Hw$ and $Hw \subseteq S$,
Then

$Ow \subseteq S$ and $Tw \subseteq S$ (Transitivity Relationship)

1. Let $P(Si)^{1w}$ be the function of predicting next one word in a sentence

Then

$P(Si)^{1w} = \{Si\} \quad \{Pw\}$, where I = 1 to x and Pw is the set of predicted one word

That means the predicted one word will be appended to the previous sentence written by the user.

2. Let $P(Si)^{2w}$ be the function of predicting next two words in a sentence

Then

$P(Si)^{2w} = \{Si\} \quad \{P2w\}$, where I = 1 to x and P2w is the set of predicted two words

That means the predicted two words will be appended to the previous sentence written by the user.

3. Let $P(Si)^{3w}$ be the function of predicting next three words in a sentence

Then

$P(Si)^{3w} = \{Si\} \quad \{P3w\}$, where I = 1 to x and P3w is the set of predicted three words

That means the predicted three words will be appended to the previous sentence written by the user.

LEARNING AND PREDICTING WORDS AND PHRASES

The system consists of two main phases: the Learning process and the Prediction process. The system has four main processes as shown in Fig. 1, the user can use the prediction subsystem (which is the main part of the system) to predict the next word or phrase or the user can enhance the words, syntax and sentences knowledge.

As shown in Fig. 1, the system consists of four main components: 1) Learn Words, 2) Learn Sentences, 3) Learn Syntax and 4) Prediction component or module. The prediction module is further detailed in Fig. 2 and explained below. The Learn Word process will analyze the incoming text for any new words and will add any new word to the database as shown in Table 1-3. The Learn Sentence module will analyze the incoming text for any new sentences which are not already stored in the database and will add new sentences to the database. The Learn Syntax module will analyze the incoming text for any new sentence structures which are not already in the database and will add them. Currently the system deals with word structures from sentences in the form of two to six words combinations, i.e., one word followed by one word, one word followed by two words, one word followed by three words, two words followed by one word, two words followed by two words, two words followed by three words, three words followed by one word, three words followed by two words and three words followed by three words. Therefore a sentence of six words will produce many combinations because the combinations will start from word one to word ten. For example, the sentence: The man went with his daughter to buy a cat will produce ten combinations (Table 1-3) totaling 68 different combinational structures. It would seem that for learning and storing very large texts we would have to keep very huge number of combinations; although this may appear to be so but in fact these combinations would start to repeat and we would reverse the exponential growth, i.e., the number of new combinations would not increase proportionally to the increase in the number of sentences or phrases especially in the same subject domain.

Table 1: Combinations of 1 word followed by 1 word, 2 words and 3 words

Previous word	Next word	Next two words	Next three words
The	Man	Man went	Man went with
Man	Went	Went with	Went with his
Went	With	With his	With his daughter
With	His	His daughter	His daughter to
His	Daughter	Daughter to	Daughter to buy
Daughter	To	To buy	To buy a
To	Buy	Buy a	Buy a cat
Buy	A	A cat	
A	Cat		

Table 2: Combinations of 2 words followed by 1 word, 2 words and 3 words

Previous two words	Next word	Next two words	Next three words
The man	Went	Went with	Went with his
Man went	With	With his	With his daughter
Went with	His	His daughter	His daughter to
With his	Daughter	Daughter to	Daughter to buy
His daughter	To	To buy	To buy a
Daughter to	Buy	Buy a	Buy a cat
To buy	A	A cat	
Buy a	Cat		

Table 3: Combinations of 3 words followed by 1 word, 2 words and 3 words

Previous three words	Next word	Next two words	Next three words
The man went	With	With his	With his daughter
Man went with	His	His daughter	His daughter to
Went with his	Daughter	Daughter to	Daughter to buy
With his daughter	To	to buy	To buy a
His daughter to	Buy	Buy a	Buy a cat
Daughter to buy	A	A cat	
To buy a	Cat		

Figure 2 shows the details of the prediction process, whenever the user calls the prediction process, he has to open an MS-Word document and the system will read the text as the user types and the system will calculate the code for the text then access the sentences database to find all the possible sentences for this code and display them in one word list, two word list and three word list, then the system will sort them in a descending order

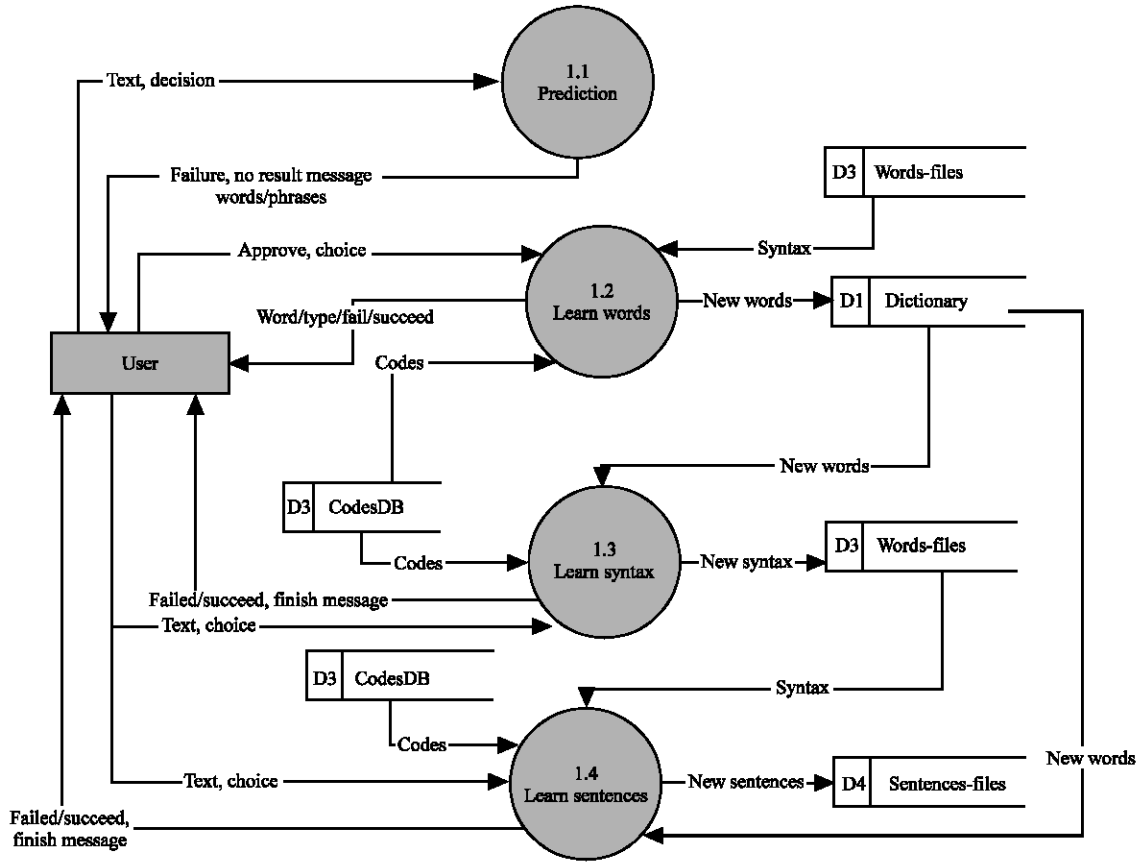

Fig. 1: Sentence and syntax learning and prediction components and their associated data structures

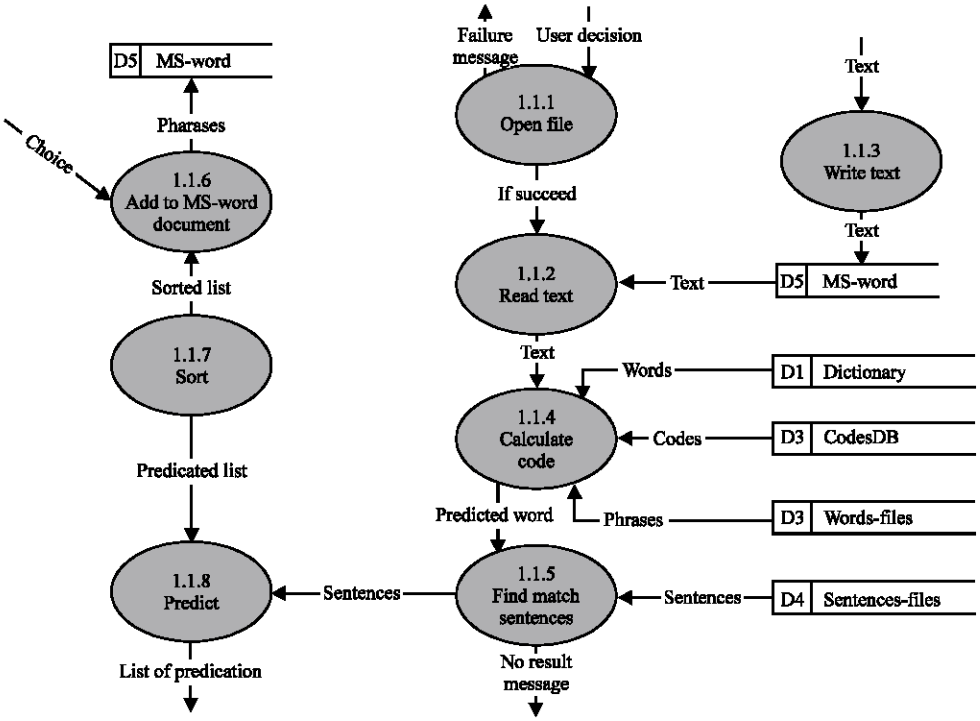

Fig. 2: Sentence and syntax learning and prediction components and their associated data structures

according to their fit status, i.e., most appropriate fitment first, then the next lower fit. This is done in accordance with the syntax and grammar structure stored for the previous phrase and the predicted phrase.

Opening the prediction file process goes into more than one sub process because there are many options either to open a new file or to browse for an existing one, then the user can choose between auto prediction mode (that is when the user types a word, then the next word (phrase) will automatically be predicted) or between prediction on demand (that is whenever the user needs prediction he has to ask for it). Furthermore, if the user chooses auto prediction mode, there is also an option for the prediction speeds to match the time needed for typing one word. If the user chooses the prediction on demand mode then the system will disable the timers, otherwise it will set the timer to the beginner or professional interval and if the user chooses to open an existing document, it must be MS-Word document file. After processing all the options the system will open the prediction document.

Figure 2 showed the five processes 1.1.1 to 1.1.5, after opening a file, the system has to read the user's text, that is if the prediction is in auto mode, then the system has to wait for reaching the timer interval, then it will check the document if there is any change (either write or delete words), then if it is the first time the system predicts in this document, it will scan it to find the last end of sentence mark and save its index then read maximum three previous words from the new index, but if it is not the first time then scan from the last end of sentence mark and update the index. On the other hand, if the prediction is done on a demand from the user, then it will just read a maximum of three words from the last word in the document.

Calculate the code for the sentence process will find the type and code for each word and the sentence length, then check the existence in the database according to the length, if it exists, it will take all the possible next predicted codes. Then after finding the sentences that match all the possible next predicted code which is combined of the next one word, next two words and next three words, the predicting process will display the predicted words (phrases) in one word list, two words list and three words list according to their length. Sorting the prediction lists is the next process to be done; the lists will be sorted according to their frequencies using the selection sort process.

The third part of the system is to enhance the syntax knowledge. First the user has to open a file and then the system will read the sentences from the file and find each sentence's word type. After that it will save the new syntax rule in the syntax database. That is, the system will divide the syntax rule of the sentence into 1 word_1 word, 1 word_2 word, 1 word_3 word, 2 word_1 word, 2 word_2 word, 2 word_3 word, 3 word_1 word, 3 word_2 word, 3 word_3 word partitions and it will check its existence in the corresponding database, if it is new syntax rule then it will read the next sentence and process it.

The last part of the system is to enhance the sentences knowledge. First the user opens a file, then the system will read the sentence and find the type for each sentence After that it will save the new sentence in the sentences database. The system will divide the sentence into partitions (sub-phrases) and check their existence in the corresponding database, if the sentence's partition already exists then it increases its frequency and it will save the new sentence and any of its sub-phrases.

The second subsystem is enhancing the word knowledge. The user has to open a file and the system will scan the sentences in it, then finds each sentence's words types, but if a word is new; the system doesn't have its type, then it will save it's index in the sentence. After that it will divide the code to compare the words before and after the unknown with the stored syntaxes to find a match for it and the unknown word will have the type from the syntax rule that is corresponding to its position in the sentence.

Learning words, this is browsing for just MS-Word document and opens it then start reading the sentences from it and finding the types for each word. We have to deal with this problem because most of the words in the English language have more than one type. If the word has more than one type and it has one word before it then the system will take the word before it and two words after it and find the type for each for the words that come after it, if their types exist (not new words) then check syntax rule for this partition with the stored syntax rules and compare the words before and after the word, if they match then the word (that has more than one type) will take the type from the syntax rule that is corresponding to it's index in the sentence. But if they do not match, the syntax rule will then check other possibilities if the words after also has more than one type; that is check all syntax rules for each type till finding the match one.

If the word has more than one type and has three or two words before and one word after it, then it will find the type for the word that comes after and check the match syntax then compare the syntax rule with the words before and after, if they are matching then the word with the multiple type will take the type of the corresponding one in the syntax rule.

After finding the new word index and finding the code for the whole sentence, the system will divide the sentence. first the system will count the number of unknown words in the sentence, if one or more unknown words than one but there are at least three words between them (in order to be able to find the syntax rule from the database). After that it will count number of words before and after the new word, then it will find the match syntax's database according to each division length and find the match syntax rule from that database, then it will extract the unknown word's type from the syntax rule according to its position in the division and in the syntax rule, but if there is no match, it will take the next division.

After that the system will check if the word's type is found or not, if it is still unknown, it will be displayed for the user to find it's type and add it to the database, but if found then it will be displayed for the user to approve saving it in the dictionary.

English Prediction system deals with many entities, it has the dictionary entity which has all the words and their types and the code entity which has for each word type a corresponding code, so one code represents many words. There are also the syntaxes entities which are in the form of 1word that may followed by 1 word or 2 words or 3 words of the syntaxes types, the syntax is also formed as 2words that may be followed by 1 word or 2 words or 3 words of the syntax types and they also formed as 3 words that may be followed by 1 word or 2 words or 3 words of the syntax types. One type in the dictionary is relating to many syntax 1words. The system has also the sentences entities which are in the form of sentences of length 1 that may be followed by sentences of length 1 or 2 or 3, they can be formed also as sentences of length 2 that may be followed by a sentences of length 1 or 2 or 3 and they can be formed as a sentences of length 3 that may be followed by sentences of length 1 or 2 or 3. All the entities that the system consists of and the relationship between them are shown in Fig. 1 and 2.

The process in this level is used to analyze text. The input comes from a text file. The analyzed phrase is sorted in the database under the phrase data store. The predicting process is used to predict text. The input comes from the word data store. It also receives input from the phrases data store. It also provides statistical information to the words data store. Reporting process receives input from the phrases table. It provides statistical reports to the user. It receives statistical information from the phrase data store.

LEARNING AND PREDICTION ALGORITHMS

As earlier explained the general method for learning the words (phrase) sequences, the syntactical structures of these sequences and for the prediction procedure. These different learning and prediction procedures are further elaborated in the form of pseudo code algorithms which are self explanatory and reflect our programs which produced the testing results. Figure 3 shows the Learning Syntax algorithm. This algorithm identifies (learns) all the different syntax structures which are read from the new

```
Browse for MS-Word document
Open MS-Word document
Scan Document sentence by sentence.
For each sentence in the document Do
Begin
        Calculate type for each word
        If the word has more than one type then
                Find type for words after it
                Compare codes before and after the word with those stored in syntax database
                If (match found) then
                        The word takes the type in the match code
                End if
End if
        Concatenate the sentence code
        While (not end of sentence) then
        Begin
        For I:=1 to m
              For j:= 1 to n
                    Divide the sentence and save into syntax- table [I word, j word]
        End for
          End while
End For
```

Fig. 3: Learning syntax algorithm

```
Browse for MS-Word document
Open MS-Word document
Scan Document sentence by sentence.
For each sentence in the document Do
Begin
        Calculate type for each word
        If the word has more than one type then
            Find type for words after it
            Compare codes before and after the word with those stored in syntax table
            If (match found) then
                The word takes the type in the match code
            End if
End if
        Concatenate the sentence code
        While (not end of sentence) then
        Begin
        Call Divide the sentence and save it into 9 tables
        For I:=1 to m
            For j:= 1 to n
                Divide the sentence and save into sentence -table [I word, j word]
            End for
        End while
End For
```

Fig. 4: Learning sentences algorithm

```
Browse for MS-Word document
Open MS-Word document
        Scan Document sentence by sentence.
For each sentence in the document Do
        Begin
                Calculate type for each word
                If the word has more than one type then
                    Find type for words after it
                    Compare codes before and after the word with those stored in syntax table
                    If (match found) then
                        The word takes the type in the match code
                    End if
                End if
        If the word is a new word then
                Save its index in the sentence
        End if
                Concatenate the sentence code
                If (there is new word in the sentence) then
                    Divide the sentence according to no. of words before and after the new
                    Compare with syntaxes from the syntax table
                    If (match found) then
                        The new word takes the type from the syntax mle
                        Save the new word in the dictionary table
        End if
                End if
End For
```

Fig. 5: Learning words algorithm

input text and saves any new syntax structure in the designated table. Figure 4 shows the algorithm responsible for identifying (learning) any new words sequences from the input text and saves it in the designated table. Figure 5 shows the algorithm responsible fro identifying (learning) the type of each word based on its position in the sentence and based on its position in other existing structures. Figure 6 is the main algorithm of the system which uses all other existing information from the syntax, sentence and word type structures and the statistics calculated in order to predict the next one word, the next two words or the next three words depending on the previous one word, two words or three words.

```
Choose file for prediction
If new then
        Open new MS-Word document.
Else if existing then
        Browse for MS-Word document
        Open file
End if
Choose type of prediction
If Auto prediction then
        Switch to auto prediction mode
Else if Prediction on demand then
        Switch to demand mode
End if
If auto prediction then
        Choose speed of typing
        If speed for beginner then
                Set the auto interval to beginner
        Else if speed for professional then
                Set the auto interval to professional
        End if
End if
If (user choose to predict in Prediction on demand) OR (time interval exceed in Auto prediction mode) then
        Read the current sentence length
        If length =1 then
                Read the last word
                Calculate its code
                Read predicted code from 1word_1word syntax table
                Read predicted code from 1word_2word syntax table
                Read predicted code from 1word_3word syntax table
                Find all matches from 1sentence_1sentence table based on code
                Find all matches from 1sentence_2sentence table based on code
                Find all matches from 1sentence_3sentence table based on code
        Else if length =2 then
                Read the last two words
                Calculate their combination code
                Read predicted code from 2word_1word syntax table
                Read predicted code from 2word_2word syntax table
                Read predicted code from 2word_3word syntax table
                Find all matches from 2sentence_1sentence table based on code
                Find all matches from 2sentence_2sentence table based on code
                Find all matches from 2sentence_3sentence table based on code
        Else if length >= 3 then
                Read the last three words
                Calculate their combination code
                Read predicted code from 3word_1word syntax table
                Read predicted code from 3word_2word syntax table
                Read predicted code from 3word_3word syntax table
                Find all matches from 3sentence_1sentence table based on code
                Find all matches from 3sentence_2sentence table based on code
                Find all matches from 3sentence_3sentence table based on code
        End if
        Sort Predicted words (phrases) according to their frequencies using selection sort
        Display Predicted words (phrases) in prediction list
End if
        If (user chooses word (phrase) from the list) then
        Insert word (phrase) in MS-Word document
End if
```

Fig. 6: The prediction algorithm

RESULTS AND TESTING

An experiment was conducted in order to evaluate our approach. The users were divided into 2 groups based on their experience in typing. The users were asked to type different paragraphs made up of an average of 7 lines with an average of 120 words in each paragraph. The beginners group was considered to be slow typists (less than 20 words per min). The advanced group had a lot of experience in typing long documents (above 50 wpm). The beginners, without the aid of the word prediction tool, took an average time of 620 sec. The same groups of beginners were then asked to type the same paragraph, but this time they were asked to use the word

prediction tool to assist them in typing the document. They took an average time of 334.3 sec. Similarly the same group was asked to type the paragraph using the word prediction tool. This time the software was fed with text documents related to the subject being typed. This increased the prediction capabilities of the software and they took an average time = 223 sec.

The results obtained above were used to calculate the efficiency increase/decrease while using the software.
Average time taken without prediction tool = 620 sec
Average time taken with prediction tool = 334 sec
Average time take with prediction tool with increased vocabulary = 223 sec

Efficiency increase between time taken with and without tool:
=100 - (334/620) × 100
= 46% efficiency increase

Efficiency increase between time taken without tool and with tool containing subject related words and phrases:
= 100 - (223/620) × 100
= 65% efficiency increase

The following results were obtained for the advanced group without the help of the prediction tool:
Average time taken = 116 sec

When the advanced users were asked to use the prediction tool to assist them in typing the text, they took an average time of 88 sec.

Similarly the advanced groups were asked to type the paragraph using the word prediction tool. This time the software was fed with text documents related to the subject being typed. This increased the prediction capabilities of the software. The following results were observed:
Average time = 44 sec

Average time taken without prediction tool = 116 sec
Average time taken with prediction tool = 88 sec
Average time taken with prediction tool with increased vocabulary and subject related phrases = 44 sec

Efficiency increase between time taken with and without tool
= 100 - (88/116) × 100
= 25% efficiency increase

Efficiency increase between time taken without tool and with the tool containing increased vocabulary
= 100 - (44/116) × 100
= 63% efficiency increase

All the users were comfortable with using computers in general and they were given time to familiarize themselves with the prediction tool before the experiment. Users were divided into groups based on their typing speed only. The beginners experienced a higher efficiency increase than the advanced users. The reason for this may be that advanced users usually would not want to stop and use the prediction features as they can already type at a fast rate, they would only use the feature for some words, or if the word prediction tool predicted a whole phrase as shown in the results for advanced users using the software with an increased vocabulary set (63% efficiency increase). The beginners experienced a 65% efficiency increase when using the software with an increased vocabulary set. This greatly enhanced the prediction capabilities of the software.

Predicting the next one word, the next two words or the next three words can be based on the previous one word, previous two words or previous three words (Table 1-3). Figure 7-9 present the results realized from the prediction process depending on the previous one, two or three words. All the results are based on the data entered in the system for a computer subject and the users were from a computing background. Open ended prediction for any subject requires more research and data and it is outside the scope of this study.

Figure 7 presents the results realized by predicting only the next word based on the previous one to five words. It is observed that as we increase the number of previous words in the sentence upon which the program makes the prediction, we increase the possibility of the user selecting one of the predicted single words. In addition, the more previous words in the sentence used to

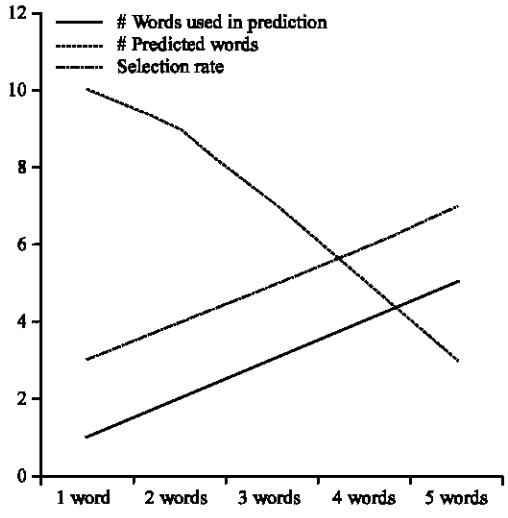

Fig. 7: Next 1 word prediction based on a phrase of 1, 2, 3, 4 and 5 words

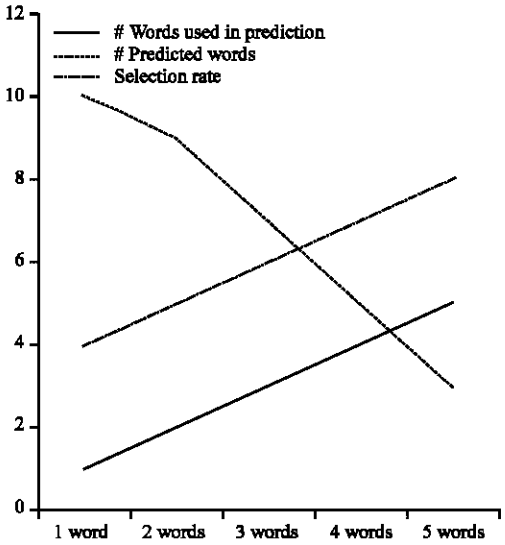

Fig. 8: Next 2 words prediction based on a phrase of 1, 2, 3, 4 and 5 words

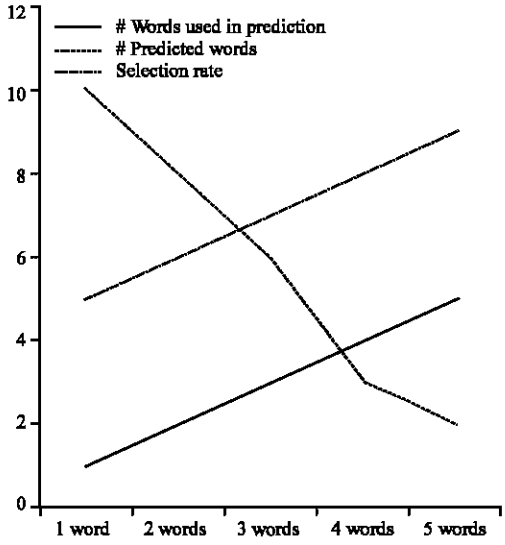

Fig. 9: Next 3 words prediction based on a phrase of 1, 2, 3, 4 and 5 words

decide the prediction, the less words are predicted and thus producing a shorter list of words for the user to select from which appears more helpful and hence the higher selection rate. For example, when five previous words were used to make the prediction, the number of predicted words was only 3 and the selection rate was 70% whilst when only one previous word was used to make the prediction of the next possible word, the number of predicted words reached the maximum, but the selection rate was very low not exceeding 30%. From the results, we notice that predicting only the next word is beneficial to some extent but since the list of predicted words tends to be high the user finds it difficult to make a good selection.

Figure 8 presents the results of predicting the next two words based on the previous one, two, three, four and five words. Predicting more than one word is more of predicting a phrase rather than a word. We observe from Fig. 8 that the selection rate is relatively higher than in Fig. 7 and the number of predicted two word phrases is less. For example, when the previous five words were used to make the prediction of the next two words, the number of predicted two word phrases is down to three but the selection rate is as high as 70%.

The results of predicting the next three word phrases is presented in Fig. 9. Clearly, predicting the next three word phrases has a better impact on the user as can be seen from the results. For example, when predicting the next three words based on the previous five words, the number of predicted three word phrases is down to two and the selection rate is as high as 90%. This makes a lot of sense since the combination of the previous five words and the next three words adds up to eight words and within the same field for eight words to be together is not very high.

The results presented in Fig. 7-9 are by no means fixed results and would change if larger amounts of data is fed to the learning modules but the trend would be the same. Of course, our next task is to base predictions based on more than five words and to make predictions of more than three words. But in conclusion, the more previous words used for the predictions process the less predicted lists of words and the higher the selection rate.

In this research, we have not investigated longer combinations than the current lengths ranging from one to three words phrases which gives a maximum of one word to three word predictions at any moment in time, but in future research we will be doing just that to see the benefit in word prediction and on the number of combinations we would have to keep.

CONCLUSIONS

This study presented an approach and a number of algorithms for word prediction and sentence generation. The results demonstrate that this approach reduces keystrokes by the user and help the writer to select better words and phrases for the text. Word prediction based on statistical uses of words and phrases is acceptable but does not guarantee correct grammar of the generated or predicted text. We demonstrate that word prediction and text generation is greatly enhanced when the learning process is complemented with learning the syntax structures of the text. All the corpora of the learnt text and

the learnt syntax are kept to help in correct prediction and text generation. Currently the system was tested on prediction based on the previous one, two, three, four and five words but we aim to expand the testing based on any number of words. The results clearly show that the more words used in the phrase for the predictions process the less list of word sequences are predicted and the more likely the user would make a selection from the predicted list. The research demonstrates that the selection by the user of the predicted text is improved if it is based on vocabulary and phrases learnt for the same subject domain. Further, the research demonstrates that the user is more likely to choose from longer predicted phrases than single words. We aim to continue working on utilizing the syntax in helping to make a more accurate prediction.

REFERENCES

Alvar, V.P., N. Kai and S. Tapio, 2006. Analysing performance in a word prediction system with multiple prediction methods. Computer Speech and Language, doi:10.1016/j.csl.2006.09.002.

Boissie`re, P. and D. Dours, 2002. An overview of existing writing assistance systems. In: Modeling, Measurement and Control, Se´rie C, (bioenginaring), Association for the Advancement of Modeling and Simulation Techniques in Enterprise, (AMSE), pp: 119-128.

Diana, D. and L. Jo-Anne, 2007. Cognitive load in hypertext reading: A review. Comput. Hum. Behav., 23: 1616-1641.

Dick, K., 2001. Most U.S. workers comfy with technology-study. Newsbites, from Computer Almanac: Interesting Information and Numbers About Computers. http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/bam/www/numbers.html.

Erman, B. and B. Warren, 2000. The idiom principle and the open choice principle. Text, 20: 29-62.

Eyas, E., 2004. Word prediction using a clustered optimal binary search tree. Inform. Process. Lett., 92: 257-265.

Foster, G., P. Langlais and G. Lapalme, 2002. User-friendly text prediction for translators. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Philadelphia, pp: 148-155.

Garay-Vitoria, N. and J. Abascal, 2005. Text prediction systems: A survey. An internal report. EHU-KAT-IK-05-05, Computer Architecture and Technology Department, University of the Basque Country.

Gibler, C.D. and D.S. Childress, 1982. Language anticipation with a computer-based scanning communication aid. Proceedings of the IEEE Computer Society Workshop on Computing to the Handicapped, pp: 11-15.

Horstmann, H.M. and S.P. Levine, 1991. The Effectiveness of Word Prediction. Presperin, J. (Ed.), Proceedings of the 14th Annual RESNA Conference, pp: 100-102.

Hunnicutt, S., 2002. Improving word prediction using markov models and heuristic methods. Augmentive Alternative Commun., 17: 255-264.

Ian, R. and C. Stephen, 2006. Stochastic and syntactic techniques for predicting phrase breaks. Comput. Speech Language, doi:10.1016/j.csl.2006.09.004.

Lesher, G.W., B.J. Moulton, D.J. Higginbotham and B. Alsofrom, 2002. Limits of human word prediction performance CSUN 2002. California State University, Northridge.

Lin, D.M., 2003. Hypertext for the aged: Effects of text topologies. Comput. Hum. Behav., 19: 201-209.

Lin, D.M., 2004. Evaluating older adults_retention in hypertext perusal: Impacts of presentation media as a function of text topology. Comput. Hum. Behav., 20: 491-503.

Lois, B. and P.O. Drawer, 1998. Two simple prediction algorithms to facilitate text production. Proceedings of the 2nd conference on applied natural language processing, pp: 33-40.

Matiasek, J., M. Baroni and H. Trost, 2002. FASTY. A Multi-Lingual Approach to Text Prediction. Miesenberger, K. and K.J. Zagler (Eds.), Proceedings of the Eight Intentional Conference Computers Helping People with Special Needs, ICCHP 2002, pp: 243-250.

Potelle, H. and J.F. Rouet, 2003. Effects of content representation and readers_prior knowledge on the comprehension of hypertext. Int. J. Hum. Comput. Studies, 58: 327-345.

Raskind, M.H. and E.L. Higgins, 1998. Assistive technology for postsecondary students with learning disabilities: An overview. J. Learning Disabilities, 31: 27-40.

Shieber, M.S. and E. Baker, 2003. Abbreviated text input. In: International Conference on Intelligent User Interfaces, Miami, Florida USA., pp: 293-296.

Tam, C., D. Reid, S. Naumann and B. O'Keefe, 2002. Perceived benefits of word prediction intervention on written productivity in children with spina bifida and hydrocephalus. Occup. Therapy Int., 9: 237-255.

Tina, M. and Hunnicutt, 2002. Sheri Measuring the effectiveness of word prediction: The advantage of long-term use. Speech Music Hearing, 43: 57-67.