A Learning-Classification Based Approach for Word Prediction

Hisham Al-Mubaid
Computer Science Department, University of Houston-Clear Lake, USA

Abstract: *Word prediction is an important NLP problem in which we want to predict the correct word in a given context. Word completion utilities, predictive text entry systems, writing aids, and language translation are some of common word prediction applications. This paper presents a new word prediction approach based on context features and machine learning. The proposed method casts the problem as a learning-classification task by training word predictors with highly discriminating features selected by various feature selection techniques. The contribution of this work lies in the new way of presenting this problem, and the unique combination of a top performer in machine learning, svm, with various feature selection techniques MI, X^2, and more. The method is implemented and evaluated using several datasets. The experimental results show clearly that the method is effective in predicting the correct words by utilizing small contexts. The system achieved impressive results, compared with similar work; the accuracy in some experiments approaches 91% correct predictions.*

Keywords: *Word prediction, word completion, machine learning, natural language processing.*

Received January 7, 2006; accepted June 6, 2006

1. Introduction

Word Prediction (WP) is an important Natural Language Processing (NLP) task in which we want to predict (*determine*) the correct word in a given context. Word prediction task can be employed in many applications, for example, predictive text entry systems, word completion utilities, and writing aids [9, 13]. Statistical and similarity based approaches have done quite well in tackling this problem just like other similar problems such as *word sense disambiguation* [4, 12, 21, 22, 23]. A common approach to handle such disambiguation-like problems is to train and apply word *bigram* or *n-gram* models.

This paper presents an effective method for word prediction using machine learning and new feature extraction and selection techniques. We use feature selection techniques adapted from Mutual Information (MI) and Chi-square (X^2). These feature extraction and selection techniques, *MI* and X^2, have been used successfully in Information Retrieval (IR) and Text Categorization (TC) [10, 11, 26]. Thus, the *WP* problem here is casted as a word classification task in which multiple candidate words are classified to determine the most correct one in the given context. For example, in this word prediction instance:

$$[w_n \ldots w_3\ w_2\ w_1\ \text{-?-}\ ]$$

we wish to predict and determine the word that follows the sequence … $w_3\ w_2\ w_1$ (i. e., the word in place of the "*-?-*").

The proposed method has a unique way of learning the representations of words in a given corpus:

1. For a given occurrence of a word *w*, the representation of *w* involves recording the occurrence of certain word features extracted from the training corpus using new feature extraction techniques adapted from *MI* and X^2.
2. The encoding of (1) is used in the training phase to train word classifiers using the *SVM* learner.
3. The word classifiers of (2) are then employed by word predictors in a new way to determine the correct word given its context. One of the properties of this method is that it performs WP by utilizing very small contexts (only preceding three words).

The method has been implemented and evaluated extensively; the experiments and results are reported in this paper. The results clearly demonstrate that the method is effective in predicting correct words by utilizing very small contexts. The system achieved accuracy approaching *91%* in some experiments, and outperforming most of the published methods on this task.

The rest of the paper is organized as follows. Section 2 presents a brief overview of the related work. The proposed methods including feature selection, learning, and prediction are explained in section 3. Section 4 describes the baseline method. The evaluation process and experimental results are discussed in section 5. Finally, section 6 presents the conclusion.

2. Related Work

A number of methods and systems have been proposed for word prediction in the past few decades. These methods can be classified as statistical methods that are based on statistical (and probabilistic) language models; and syntactic methods in which syntactic information is extracted and exploited in word prediction task. In [9], Fazly presents a comprehensive review of prior related work in word prediction. Fazly also presents a collection of experiments on word prediction applied to word completion utilities. The implemented and evaluated algorithms [9] were based on word unigrams and bigrams, and based on syntactic features like POS tags in the syntactic predictors, and combination. The training and testing are done on texts taken from British National Corpus (BNC). Roughly speaking, tags-and-words predictors achieved the best overall performance with *hit rate* approaching *37%*, and keystroke savings around *53% —hit rate* is defined to be the percentage of the times that the correct word appears in the prediction list. Among the other related interesting work is the approach presented in [7]. That approach attempts to learn the contexts in which a word tends to appear, using expressive and rich set of features. The features are introduced in a language as information sources. It also attempts to augment local context information by global sentence information. The evaluation of the method in this paper is very similar to that presented in [7].

One of the related problems to word prediction is the context-sensitive spelling error correction, or *malapropisms* [2, 14]. In this problem, the misspelled variant of the original word is a correct word and belongs to the language [2, 14, 15]. For example, the misspelling of the word *quite* as *quiet* is a context-sensitive spelling error. Since *quiet* is a valid word in *English*, the traditional spell-checkers will not discover this spelling error. Thus, the function of the context-sensitive spelling correction is to choose, for an instance for a word in text (e. g., *quite)*, its correct spelling from its confusion set (e. g., *quite, quiet*). It is worth mentioning at this point that *word prediction* can be harder than *context-sensitive spelling problem* such that, in the latter problem the size of the given context is double the size of the given context in word prediction. That is, in word prediction, only the preceding words are available as context to the prediction task, whereas in the context-sensitive spelling correction task, the words before and after the target word are available as a context. Of course the context of prediction or classification task is critically an important resource for such a task.

3. The Proposed Method

The proposed method is based on representing each word as a feature vector, and then using machine learning to train word classifiers during the training phase. The word classifiers are then employed, in the perdition phase, to determine from a confusion set the correct word in a given context. Thus, the task is casted as a word classification task. For example, let the confusion set be {*weak, week*} then when a user types the letter '*w*' the word prediction task triggers and tries to determine whether the user wants to type '*weak*' or '*week*'.

In a given context (e. g., [w_n, …, w_3 w_2 w_1 $\underline{w_x}$]), we want to predict the word w_x, such that the context of the word to be predicted (e. g., {w_1, w_2, w_3, ….w_n}) is given along with the confusion set. The confusion set is the set of the alternative (*candidate*) words in this context, e. g., {w_x, w_y}. We want to determine/predict which of the two *candidate* words {w_x, w_y} should be in this context. In word completion utilities, the word prediction task can start after typing the first letter of the target word, so that, the prediction task can be limited to alternative words that start with that entered letter. In this research, we follow the majority of researchers and assume that the confusion sets are predetermined [7, 8, 14, 15]. Each confusion set contains two or more of the mostly confused words in the language. For example, MS Word [19] utilizes a list of confusion sets, called *commonly confused words*, for grammar checking. Such a list, shown in Table 1, can be used as a basis for a *WP* task.

Table 1. A part from the commonly confused words list of MS Word.

Commonly Confused Words
Abut-About, Adept-Adapt, Adepts-Adopts, Ads-Adds, Advice-Advise, Aid-Aide, Ail-Ale, Alters-Altars, Assess-Asses, Augur-Auger, Bare-Bear, Beet-Beat, Bettor-Better, Border-Boarder, Breath-Breathe, Bridal-Bridle, Broach-Brooch, …
…
…
Theirs-Their's, Tide-Tied, Undo-Undue, Upwards-Upward, Urn-Earn, Vein-Vain, Who's-Whose, Wile-While, Wither-Whither, Won't-Wont, Yolk-Yoke, You're-Your

Examples of confusion sets used in this research include: {*quite-quiet, peace-piece, passed-past, being-begin, than-then, raise-rise, site-sight*} (Table 5). Now we can summarize the problem as follow. Let c = {w_1, w_2, …, w_n} be the context of the prediction task where n is an integer number represents the size of context window (*in this research we tested for n values of 3, 5, or 10*). The words w_1, w_2, …, w_n are the words that appear immediately before the word to be predicted. Also let f = {w_x, w_y} be the confusion set for this case. Our proposed method relies on machine learning to train word classifiers to classify (predict) whether w_x or w_y is the predicted correct word in that context. Each word in the confusion set is represented as a projection on the feature vector that is composed from the training data. One of the contributions of this work is in the way we extract and compute the features from the training data. We describe next the feature

extraction process and then we talk about the learning and the prediction steps.

3.1. Feature Selection and Extraction

Let a training text T be given. We extract from T all the occurrences of the confusion set words w_x and w_y. Each occurrence is extracted along with its context (preceding n words) to make one *training* example of the form [w_n … w_3 w_2 w_1 $\underline{w_x}$] or [w_n …w_3 w_2 w_1 $\underline{w_y}$]. Thus, we have now two sets of training examples; the training examples of w_x and the training examples of w_y both extracted from T. We convert each example into a feature vector as follows. The given context words are used as features in some of the related work [14, 22, 23]. In this research, however, we do not use word features directly from the contexts; instead we select, as features, only certain words with high "*discriminating*" capabilities between the two confused words (w_x and w_y). These features are used to represent each example in the training and prediction. We use the confusion words occurrences extracted from the training text T as labeled training examples. Feature selection is a key issue in the effectiveness and efficiency of the learning and classification performance of such methods as the one presented here.

Before delving into the details of feature selection, let us mention that there has been a lot of research devoted to feature selection in machine learning and data mining, particularly in *text categorization* research, see for example [10, 11, 26]. Assume that we have two classes C_1 and C_2 of labeled examples extracted from the training text T. Let C_1 contains examples of w_x and their contexts, and C_2 includes examples of w_y with their contexts. We extract all the context words $W = \{w_1, w_2, \ldots, w_m\}$ from the sets C_1 and C_2. Now, each such context word $w_i \in W$ may occur in contexts from C_1 or C_2 or both with different frequency distributions. Now, if a context word $w_i \in W$ appears in a context of a prediction example, we would like to be able to determine to what extent the existence of w_i suggests that this example belongs to C_1 or C_2. Thus, we select those words w_i from W which are highly associated with either C_1 or C_2 (*the highly discriminating words*) as features. We utilize feature selection techniques like *MI* and X^2 [11, 26] to select the highly discriminating context words from W. MI and X^2 were used effectively for feature selection in text categorization and information retrieval [10, 11, 26] but never been utilized for language prediction or classification problems. In the rest of this section, we explain how *MI* and X^2 are applied to determine which context words from W will be selected as features.

Let us first define the notions of a, b, c, and d as follows. From the training examples, we calculate four numeric values a, b, c, and d for each context word $w_i \in W$ as follows:

a = Number of occurrences of w_i in C_1.
b = Number of occurrences of w_i in C_2.
c = Number of examples of C_1 that do not contain w_i.
d = Number of examples of C_2 that do not contain w_i.

Then, MI is defined as:

$$MI = \frac{N*a}{(a+b)*(a+c)} \quad (1)$$

Where N is the total number of examples in C_1 and C_2. Chi-square (X^2) is computed as:

$$X^2 = \frac{N*(ad-cb)^2}{(a+c)*(b+d)*(a+b)*(c+d)} \quad (2)$$

Again, N is the total number of examples in C_1 and C_2.

Illustrating the proposed WP method by an example when using the *MI* technique for feature selection, we calculate the *MI* value for each $w_i \in W$. Then we choose the k top $w_i \in W$ words with the highest *MI* values as features. In our experiments, we tested on k values of 10, 20, and 30. For example, if $k = 10$, then each training example is represented by a vector of 10 entries, such that, the first entry represents the word with the highest *MI* value, the second entry represents the word with the second highest *MI* value, and so on. Then for a given training example, the feature vector entry is set to 1 if the corresponding feature word occurs/appears in that training example, and set to 0 otherwise. Thus, if we want to utilize the 20 most discriminating words as features to represent each example, then feature vector size will be 20. Consider the following example, let $W = \{w_1, w_2, \ldots, w_m\}$ be the set of all context words. We compute *MI* for each $w_i \in W$ and sort the words W according to their *MI* values in descending order as in Table 2.

Table 2. Words $w_i \in W$ with the highest MI values.

Context Words w_i	***MI***
Person	1.92
Nice	1.90
Found	1.87
Still	1.86
Place	1.68
Generate	1.56
Went	1.48
Clear	1.33
Deliver	1.33
Small	1.27
…	…

Table 2 shows the top 10 context words having the highest 10 MI values. These 10 words will be used to compose the feature vectors for training and prediction examples. For example, the following feature vector:

0	*1*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*

represents an example containing the 2nd, 3rd and 7th feature words (*viz., nice, found,* and *went)* in the given

context. Additionally, if the window size is 5, then that example may look like:

$$went \_\_\_ \; nice \_\_\_ \; found \; < w_x \; or \; w_y >$$

That is, three of the 10 feature words are occurring within the preceding 5 words of the word to be predicted. In this case, window size is 5 and the vector size is 10. For example, the word '*nice'*, occurred as third preceding word in the context but it is translated to a '1' in the seventh entry of the feature vector.

Let us look into the MI feature selection technique in little more detail. The objective of MI is to select from two classes C_1 and C_2 of examples the most discriminating features (*words*). A good such feature is the one that is highly associated with C_1 but not with C_2 or vice versa. MI uses the co-occurrence counts *a*, *b*, *c*, and *d* with equation (1) to compute MI value for each feature, such that the feature with highest MI value will be the best in discriminating C_1 from C_2. The MI's formula gives most weight to *a* (the numerator in equation (1)) which represent the association between the word/ feature and class C_1. We would like to update this formula by multiplying MI by the difference ($a - b$) between *a* and *b*. Recall that, for a given word w_i, the value *b* represents the association between w_i and class C_2 (how many times w_i occurs in C_2). In this, we subtract from *a* the number of times the word is associated with C_2. That is, if a word w_i is associated *q* times with C_1 and *q* times with C2 then the formula yields zero, which is what we want, since in this case, the feature w_i is not really a discriminating feature. Thus, we applied the formula:

$$MI\_1 = MI * (a - b) \quad (3)$$

for feature selection. Furthermore, to give more weight to *a*, we also applied the formula:

$$MI\_2 = MI * a * (a - b) \quad (4)$$

Notice that equations (3) and (4) can also be written as:

$$MI\_1 = \frac{N*a}{(a+b)*(a+c)} * (a - b)$$

$$MI\_2 = \frac{N*a}{(a+b)*(a+c)} * a * (a - b)$$

respectively.

We found out after extensive experimentation, with multiple datasets, that using *MI_2* for feature selection gives, in most cases, better results than *MI* and *MI_1*, see Table 3. The results in Table 3 demonstrate clearly that our proposed feature selection technique *MI_2* which is adapted from *MI* outperforms *MI* across the three confusion sets using *Reuters* dataset. These experiments as shown in Table 3 are done on more than 3,000 prediction instances (*Table 6 gives numbers of testing instances in Reuters and other datasets*). Thus, in our experiments we used only *MI_2* (instead of *MI* or *MI_1*) and X^2 for feature selection.

Table 3. Accuracy results of four feature selection techniques on three confusion sets using *Reuters* dataset.

Confusion Sets	MI	MI*A	MI_1	MI_2
Conf. set 1	72.77	79.71	78.94	81.64
Conf. set 2	86.32	88.87	88.91	89.80
Conf. set 3	92.77	94.79	95.11	95.21

3.2. Learning and Prediction

Thus, from the training text we generate feature vectors using the top words selected using *MI_2* or X^2. Then, we use the well-established learning technique *Support Vector Machines* (SVM) [3, 25] to train classifiers with the training vectors. SVM is an inductive learning technique for two-class classification. A significant elaboration of theoretical and empirical justification has been presented in the literature to support SVM [3, 6]. Moreover, SVM was extensively applied in various areas and achieved remarkable results.

For example, in text categorization, SVM was investigated extensively and proved to be one of the best learning algorithms [6, 10, 16]. In the present method, for a given confusion set $\{w_x, w_y\}$, we construct one feature vector for each w_x and each w_y instance in the training text. Thus, these vectors will be the training examples, and we divide them into two classes, one for w_x vectors and one for w_y vectors. Then SVM trains on these two classes and produces a classifier (*model*). Thus, we construct with SVM a classifier for each confusion set. The created classifier is then used in the prediction phase to predict the word in the given context. Of course, in the prediction process, we construct a feature vector in the same way as in the training process. We use a linear SVM in all our experiments as most of related work. The implementation of SVM we used is the linear SVM-*light*, available at: http://svmlight.joachims.og with the default parameters.

4. The Baseline Method: *Naive Bayes*

We applied *Naïve Bayes* (N. Bays) for the prediction task to compare with our method. In applying *N. Bayes* for *WP*, we followed the general procedure by assuming the probabilistic model of the training examples [8]. *Naïve Bayes* was applied into many disambiguation-like NLP problems, for example, word sense disambiguation [4, 12, 21, 22, 23]. We briefly introduce *Naïve Bayes* here and describe the experimental settings with it, for more details you can refer to [12, 17]. Let $W = \{w_1, w_2, ..., w_n\}$ be the context. Let further $C = \{c_1, c_2, ..., c_m\}$ be the confusion set that contains the alternative (*candidate*) words for the prediction task. The decision rule of the Naïve Bayes is as follows:

$$c^* = \underset{k}{\text{argmax}}\ P(c_k|W) = \underset{k}{\text{argmax}}\ (P(c_k) \cdot \prod_{i=1}^{n} P(w_i \mid c_k)) \quad (5)$$

Such that $P(c_k|W)$ is the conditional probability of the confusion set word c_k appears in the context W. This decision rule selects $c^* \in C$ as the predicted word in the given context W. The probabilities P (c_k) and P ($w_i|c_k$) are computed from the training text T. Notice here that *Naïve Bayes* assumes that the context words $w_1, w_2, \ldots, w_n$ are conditionally independent. There is one issue with the *Naïve Bayes* is that the probability P ($w_i|c_k$) may, very well, be a very small value or zero, so we use a smoothing technique to avoid this problem. There are a number of smoothing techniques proposed in the literature, for example, *add-1*, *Ng's smoothing*, and *Kneser-Ney and Katz smoothing*. For more details on smoothing see [5, 14]. Chen *et al.* (1998) [5] presents a comprehensive review about the smoothing techniques.

5. Evaluation and Experimental Results

In this section, we describe the datasets used in experiments and the experimental settings, then we discuss the results.

5.1. Datasets

We used four different text datasets to evaluate our method. The details of the datasets are in Table 4. We select the testing text size to be little less than the training text size as the case in the actual prediction. The testing text size is not important and will not affect the performance as we only utilize the preceding 3 words for each prediction case. The datasets are as follows:

- The ACL dataset were obtained from Linguistic Data Consortium (LDC) (www.ldc.upenn.edu) and include news stories 1987-1991 taken from the Wall Street Journal (WSJ).
- The Reuters is taken from the Reuters-21578 benchmark dataset. Reuters-21578 contains 21578 news articles from the Reuters newswire [24].
- The BioMed text is a corpus of biomedical articles taken from Medline [18]. The Medline database is considered to be the largest and most comprehensive data resource in bioinformatics. We use this text to evaluate the performance of our method on specialized texts.
- The 10-K dataset contains financial text of 10-K filings of US corporate, taken from U.S. Securities and Exchanges Commissions (SEC) at (www.sec.gov). 10-K filing is an annual financial and transactional report required by SEC from all public companies, and it gives the most comprehensive information on financial information of a public company. At SEC website, 10-K filings of around 10,000 public companies in the last few years are available (and totally there are around 50,000 filings. The size of these documents is around 30 GB.). This dataset is another specialized text (financial text) used to evaluate our method.

Table 4. Details of the four datasets used in experiments.

Dataset (Source)	Training Text Size Words	Testing Text Size Words
Reuters (Reuters-21578)	977,418	167,835
ACL (LDC www.ldc.upenn.edu)	761,730	451,407
Biomed Text (*Medline)*	774,206	466,254
10-K (SEC at www.sec.gov)	527,390	152,069

Table 5. The three confusion sets used in the experiments.

Confusion Set 1	Accept-except, affect-effect, begin-being, country-county, …
Confusion Set 2	Site-sight, than-then, further-farther, raise-rise, …
Confusion Set 3	Advice-advise, weak-week, sea-see, lose-loose, …

5.2. Confusion Sets

We used three confusion sets in the experiments, shown in Table 5. These confusion sets were commonly used in word prediction and context-sensitive spelling research; see [2, 14, 15].

5.3. Evaluation and Discussion

Several experiments have been conducted to evaluate the method. We used *MI*, *MI_2*, and X^2 for feature selection, and *SVM* for learning and prediction; we also used the *N. Bayes* algorithm [17] as baseline to compare our results. For context size, we used preceding 3, 5, or 10 words. We found out the context of size 3, using only preceding 3 words, produces the best performance. Furthermore, we experimented on how many features to include in the feature vectors. For that, we tried 10, 20, and 30 features and found that the best performance resulted when using 20 features (i. e., using the top 20 words having the highest 20 *MI_2*, or X^2). Thus, the results reported here are generated using the preceding 3 words (context size = 3) and the top 20 *MI_2*, or X^2 words. We initially tested our method using 3 datasets; *Reuters*, *ACL*, and *BioMed* (Table 4), and the three confusion sets (Table 5). The results are presented in Table 6 when using *MI_2* for feature selection, and in Table 7 when the X^2 feature selection technique was used. With a total of 19,438 word prediction instances were tested in each experiment (Tables 6 and 7), we notice that *MI_2* (Table 6) produces slightly better accuracy than X^2 (Table 7).

Moreover, to compare our method against the baseline method we ran all the testing prediction

instances on the *Bayesian* method and the results are in Table 8. The Bayesian method produced slightly better accuracy than MI_2 only in the *Reuters* dataset, but with the other two datasets, both MI_2 and X^2 outperform Bayesian significantly (Table 8). Furthermore, the micro-average accuracy on the three datasets demonstrates that *MI_2* and X^2 outperform *Bayesian* (Table 8). Finally, since the *10-K* dataset is very specialized dataset and is not as commonly used in NLP research as the other datasets, we tested our method on it in a separate experiment using *MI_2* and X^2 with the three confusion sets, and the results are in Table 9. In this experiment too, *MI_2* with 91.42% accuracy outperforms X^2 with 87.09% accuracy. This experiment also proves that our method can achieve impressive accuracies exceeding 91% correct predictions (Table 9). Overall, our method of learning-classification-based word prediction is capable of achieving accuracy in the range of 87% – 88% correct predictions using only the three preceding words as context, which emphasizes the robustness of the feature selection techniques and the learning method. Furthermore, the experimental results proved that the method can achieve really high accuracies; for example, the method produced accuracy of ~90% using confusion set 2 and *Reuter* (Table 7), and the average accuracies on *BioMed* and *Reuters* are approaching ~89% and ~90%, respectively (Table 6). In addition, the method achieved accuracy of 95.2% on *Reuters* using confusion set 3 (Table 6) and 93.1% on the *BioMed* dataset using confusion set 3 (Table 7).

Table 6. Accuracy results with the 3 datasets and 3 confusion sets using *MI_2* for feature selection, preceding 3 words for contexts, and top 20 features.

Dataset	Confusion Set 1		Confusion Set 2		Confusion Set 3		Average Accuracy
	No. of Tested Instances	Accuracy	No. of Tested Instances	Accuracy	No. of Tested Instances	Accuracy	
Reuters	615	81.46	1481	89.80	941	95.21	89.79
ACL	2658	86.68	3149	83.39	2369	87.08	85.53
BioMed	2725	86.93	4313	88.73	1187	93.09	88.76
Total	5998		8943		4497		

Table 7. Accuracy results with the 3 datasets and 3 confusion sets using X2 for feature selection, preceding 3 words for contexts, and top 20 features.

Dataset	Confusion set 1		Confusion set 2		Confusion set 3		Average Accuracy
	No. of Tested Instances	Accuracy	No. of Tested Instances	Accuracy	No. of Tested Instances	Accuracy	
Reuters	615	81.46	1481	89.80	941	86.96	87.23
ACL	2658	85.94	3149	82.85	2369	87.21	85.12
BioMed	2725	85.13	4313	87.22	1187	93.09	87.37
Total	5998		8943		7853		

6. Contribution and Conclusion

We presented a learning-classification based method for word prediction. The method uses very small context (*the preceding three words*) to predict the following word in that context with high accuracy. The method was evaluated extensively and compared with the *Bayesian* algorithm as a baseline. The experimental results showed that our approach can achieve impressive accuracy in percentages of correct predictions, which validates its efficiency. The contribution of this work can be viewed in a number new aspects: Casting the *wp* task as a learning-classification task by using machine learning to train word predictors using highly discriminating features selected by various techniques. The presented method also includes a new feature selection technique *MI_2* adapted from *MI* and outperforms *MI* and X^2 in most experiments. Furthermore, the unique combination of one of the top performers in machine learning (*svm*) with feature selection techniques, *MI* and X^2, which are used in TC and IR, makes a good contribution into *WP*. These aspects can contribute in solving other similar NLP problems as mentioned earlier in this paper.

Table 8. Average accuracy on each method with each dataset, accuracy here is the average of testing on all confusion sets.

Dataset	No. of Tested Instances	Accuracy		
		N.Bayes	MI_2	X^2
Reuters	3037	90.67	89.79	87.23
ACL	8176	80.12	85.53	85.12
BioMed	8225	81.28	88.76	87.37
Total	19,438			
Micro. Avg		82.26	87.56	86.40

Table 9. Accuracy results for the 10-K dataset.

Dataset	No. of Tested Instances	Accuracy	
		MI_2	X^2
10-K	2,610	91.42	87.09

Word prediction is a very important task and has many significant applications. A robust word prediction system can benefit users, by allowing higher text entry rates, and minimizing number of typographical errors and misspellings. This aspect has been observed by the developers of the open-source word processor *OpenOffice* [20], which provides, along with standard word processing features, word completion (*www.openoffice.org*) [20]. In the future directions of this research, we would like to try a few new aspects to further improve the prediction accuracy. For example, we will investigate increasing the context size without affecting the computation complexity of the method. Also, we plan to explore the possibility of involving positional information about the context features in the learning process.

Acknowledgements

This work was supported by the Institute for Space Systems Operations (ISSO), February 2005.

References

[1] Al-Mubaid H., "Context-Based Word Prediction and Classification," *in Proceedings of the 18th International Conference on Computers and their Applications CATA'2003*, Hawaii, USA, pp. 384-388, March 2003.

[2] Al-Mubaid H. and Truemper K., "Learning to Find Context-Based Spelling Errors," in Triantaphyllou E. and Felici G. (Eds), *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, Kluwer Academic Publishers, Forthcoming 2006.

[3] Bose B. E. R., Guyon I., and Vapnik V., "A Training Algorithm for Optimal Margin Classifiers," *in Proceedings of COLT*, USA, pp. 144-152, 1992.

[4] Bruce R. and Wiebe J., "Word-Sense Disambiguation Using Decomposable Models," *in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*, USA, pp. 139-145, 1994.

[5] Chen S. and Goodman J., "An Empirical Study of Smoothing Techniques for Language Modeling," *Technical Report TR-10-98*, Centre for Research in Computing Technology, Harvard University, Cambridge, Massachusetts, 1998.

[6] Dumais S. T., Platt J., Heckerman D., and Sahami M., "Inductive Learning Algorithms and Representations for Text Categorization," *in Proceedings of the 7th International Conference on Information and Knowledge Management*, USA, pp. 148-155, 1998.

[7] Even-Zohar Y. and Roth D., "A Classification Approach to Word Prediction," *in Proceedings of the NAACL'00*, USA, pp. 124-131, May 2000.

[8] Even-Zohar Y., Roth D., and Zelenko D., "Word Prediction and Clustering," *The Bar-Ilan Symposium on the Foundations of Artificial Intelligence*, June 1999.

[9] Fazly A., "The Use of Syntax in Word Completion Utilities," *Master Thesis*, University of Toronto, Canada, 2002.

[10] Forman G., "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *JMLR*, vol. 3, no. 1, pp. 1289-1305, 2003.

[11] Galavotti L., Sebastiani F., and Simi M., "Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization," *in Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'00)*, Portugal, pp. 59-68, 2000.

[12] Gale W. A., Church K. W., and Yarowsky D., "A Method for Disambiguating Word Senses in a Large Corpus," *Computers and the Humanities*, vol. 26, no. 1, pp. 415-439, 1992.

[13] Garay-Vitoria N. and González-Abascal J., "Intelligent Word-Prediction to Enhance Text Input Rate," *in Proceedings of the 2nd International Conference on Intelligent User Interfaces*, pp. 241-244, January 1997.

[14] Ginter F., Boberg J., Jarvinen J., and Salakoski T., "New Techniques for Disambiguation in Natural Language and Their Application to Biological Text," *JMLR*, vol. 5, no. 1, pp. 605-621, 2004.

[15] Golding A. R. and Roth D., "A Window-Based Approach to Context-Sensitive Spelling Correction, in Machine Learning," *Special Issue on Natural Language Learning*, vol. 34, no. 1, pp. 107-130, 1999.

[16] Joachims T., "Text Categorization with Support Vector Machines: Learning With Many Relevant Features," *in Proceedings of the 10th European Conference on Machine Learning*, pp. 137-142, 1998.

[17] Manning C. D. and Sch¨utze H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Massachusetts, 1999.

[18] Medline, Accessed Using Entrez PubMed Interface, available at: http://www.ncbi.nlm.nih.gov/entrez/ query.fcgi, 2005.

[19] MS WORD: Microsoft Corporation, available at: http://www.microsoft.com/Word/, 2005.

[20] OpenOffice, *A Multiplatform and Multilingual Office Suite and Open-Source Project*, Founded and Sponsored by Sun Microsystems, available at: http://www.openoffice.org, 2000.

[21] Pedersen T., Bruce R., and Wiebe J., "Sequential Model Selection for Word Sense Disambiguation," *in Proceedings of The Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC, pp. 388-395, 1997.

[22] Pedersen T. and Bruce R., "Knowledge Lean Word Sense Disambiguation," *in Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, Madison, WI, 1998.

[23] Pedersen T., "Search Techniques for Learning Probabilistic Models for Word Sense Disambiguation," *in Proceedings of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information*, Palo Alto, CA, 1999.

[24] Reuters-21578, available at: http://www.daviddlewis.com/resources/testcollections/reuters21578/, 2005.

[25] Vapnik V., *The Nature of Statistical Learning Theory*, Springer, New York, USA, 1995.

[26] Yang Y. and Pedersen J. P., "A Comparative Study on Feature Selection in Text Categorization," *in Proceedings of the 4th International Conference on Machine Learning*, Nashville, USA, pp. 412-420, 1997.

Hisham Al-Mubaid obtained his PhD degree in computer science from the University of Texas at Dallas, USA, in 2000. He worked one year as an assistant professor at State University of New York (SUNY), USA. He joined the University of Houston-Clear Lake, USA, in 2001 as an assistant professor of computer science. His research interests and publications have been primarily centered around natural language processing, and include text categorization, machine learning, text mining, semantics and ontology. He also has interests and publications in bioinformatics and teaching-learning research. He serves in the technical and program committees of several journals and conferences.