# Extract Baseline Information Using Support Vector Machine

Walaa ALY[1], Seiichi UCHIDA[1*], Masakazu SUZUKI[2]

[1] Department of Intelligent Systems, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka-shi, 819-0395 Japan
walaa@human.is.kyushu-u.ac.jp, uchida@human.is.kyushu-u.ac.jp

[2] Department of Mathematics, Kyushu University,
6-10-1, Hakozaki, Higashi-ku, Fukuoka-shi, 812-8581 Japan
suzuki@math.kyushu-u.ac.jp

**Abstract**

Printed mathematical documents have many features which differ from text documents. These features include two-dimensional structure, such as subscripts, superscripts, and scalable symbols, such as minus, summation. Due to these features, the recognition of printed mathematical documents becomes very challenging. In this paper, a support vector machine (SVM) classifier is used to extract baseline information, which provides a basis for finding the two-dimensional structure from printed mathematical documents. Experiments using very large databases show high efficiency of the proposed method with the extraction accuracy reaching 99.25 %.

## 1 Introduction

Mathematical expressions are considered as an essential part in any scientific and technical document. Recognition of mathematical expressions have two major steps. (i) Recognition of symbols: each symbol is recognized using an OCR approach. (ii) Structure analysis: from the set of recognized symbols and their bounding boxes, the expression is reconstructed by analyzing the two-dimensional structure of the expression. A survey of past attempts on mathematical expression recognition can be found in [1].

In this paper, we will mainly focus on the structure analysis step of mathematical expressions. Many researchers have discussed the structure analysis step, starting with Anderson [2], where a purely syntactic approach was introduced for parsing mathematical expressions. Okamoto and his colleagues [3] used geometric information to find the structure of the expression. Twaakyondo et al. [4] used two strategies, namely, top-down and bottom-up, to determine the structure of the expression.

In past attempts, the evaluations have been carried out with respect to the total performance of the system and, therefore, they have not evaluated the individual steps of the structure analysis. That is, those past attempts gave neither qualitative nor quantitative analysis of the discrimination task [‡]. In contrast, this paper gives a solid ground to the structure analysis step through several evaluations with very large databases.

In this paper, baseline information is extracted as a classification task for mathematical expressions. Figure 1 illustrates the classification task. *Baseline* is a line containing only symbols which are not vertically offset from other symbols. Hereafter, these symbols will be called baseline symbols. For example, the expression "$A_2 + B_1$" has baseline symbols "$A$,""+," and "$B$." In contrast, *non-baseline* is a line containing symbols which vertically offset from baseline symbols. Hereafter, these symbols will be called non-baseline symbols. For example, the expression "$X^2 + Y_5$" has

*Correspondence to: 744 Motooka, Nishi-ku, Fukuoka-shi, 819-0395 Japan and Tel:092-802-3574, Fax:092-802-3600.

‡Moreover, in other attempts [5, 6, 7, 8, 9, 10, 11, 12, 13, 14], the detail of the structure part is completely concealed as a black box of a large math OCR system.

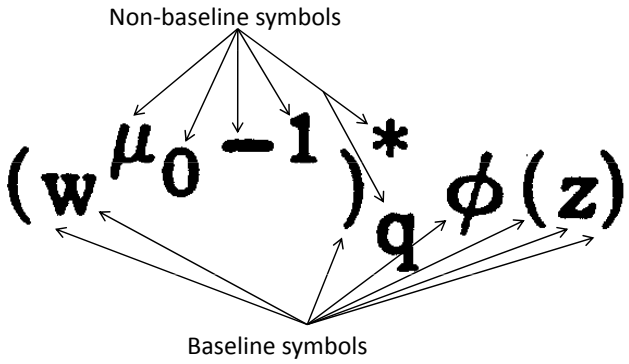


Figure 1: Baseline/non-baseline classification problem.

$$(h^{\mu}_{ik}h^{\mu}_{jl} - h^{\mu}_{il}h^{\mu}_{jk})$$

$$\sum c_{ij} g_k^{q_k - i} g_{k-1}^{q_{k-1} - j}$$

$$\mathrm{H}^r(\mathrm{K}, \mathrm{A}) \times \mathrm{H}^{1-r}(\mathrm{K}, \hat{\mathrm{A}})$$

Figure 2: Examples of mathematical expressions.

baseline symbols "$X$," "+," and "$Y$," and non-baseline symbols "2," and "5." We will classify each symbol into baseline and non-baseline as the most important step of structure analysis. In the following, all mathematical symbols, such as "$*$," "$\cup$," and "$\int$," and alphanumeric characters, such as "$A$," "2," and "$\eta$," will be called symbols.

The classification task is not trivial. Figure 2 depicts several superscripts and subscripts in various mathematical expressions. In the bottom example, it is difficult to distinguish non-baseline characters "$r$" and "$1 - r$" from the baseline characters (e.g., "$H$") by using only the offset. They clearly show how difficult it is to extract baseline information.

Although baseline information is very important, there is only little related work, to the authors' best knowledge. Zanibbi et al [15] is related to our work. They proposed a system to recognize mathematical expressions by using tree transformation. In their system, they used a layout analysis technique for discovering baseline information, but, unfortunately, they gave neither a quantitative nor a qualitative analysis thereof.

The remainder of this paper is organized as follows: Section 2 describes the classification task. Section 3 introduces the features used in the classification task. Section 4 shows experimental results with very large databases. Finally, Section 5 derives a conclusion.

# 2 The Classification Task

Our task is the classification of each symbol into baseline and non-baseline class. In this task, several features were used. These features include (i) basic features based on the height, width, location, symbol type, category name, and entity name of the target symbol, and (ii) context features based on basic features of the symbols around the target symbol. A SVM was used to determine the class of each symbol.

The main contributions of this paper are two-fold: (i) showing high classification accuracy through a very large scale experiment, and (ii) showing the importance of the context features and symbol type for better classification.

# 3 Feature Extraction

## 3.1 Basic feature

For each symbol, we extract basic features to classify it into baseline and non-baseline. The basic features include width, height, location, category name, entity name and symbol type. Each symbol has a category name and an entity name. For example, symbol "$\gamma$" has category name "Greek" and entity name "gamma." The details about the category name and entity name can be found in [16]. The details for symbol types will be discussed below.

Note that we assume that the category of the symbol is given by some preceding symbol recognition method. This assumption is often acceptable because most mathematical OCRs are comprised of two steps as noted in Section 1.

### 3.1.1 Symbol types

A type is estimated for each symbol according to its occupation of three regions called the $X$, $Y$ and $Z$ regions. Figure 3 shows these regions. Region $X$ is equivalent to the ascender part and region $Z$ is equivalent to the descender part. Note that the specification of those regions are detailed in Section 3.1.2.

Estimation of type is very important to describe the variation in symbol sizes. For example, character "$A$" has type $X$ and $Y$ and character "$a$" has type $Y$. The estimation of types for mathematical symbols has no previous published literature, to the authors' best knowledge. All the past attempts have estimated only the types of alphanumeric characters. Characters are classified into 4 types according to ascender and descender parts.

To cover all the variation in symbol sizes, each of the $X$, $Y$ and $Z$ regions is divided into 4 subregions and, therefore, we have 12 regions. For example, in Fig 3, the symbol " $-$ " occupies 0.25 of the $Y$ region, the symbol " $\in$ " occupies the full $Y$ region, the symbol " $\cap$ " occupies the full $Y$ region and 0.5 of the $X$ region, and the symbol " $\int$ " occupies all of the $X$, $Y$ and $Z$ regions. In the proposed method, we have 24 symbol types as a different combination of 12 regions.

To estimate the type of symbol " $\cup$ " in Fig 4, the top and base of the $Y$ region of the mathematical expression which included this symbol are first calculated using all the characters in this expressions; it is equal to the average of top and base of each individual character in the expression. For example, the base for character "$A$" is equal to its leftmost bounding box and its top can be calculated by subtracting this base value from the $Y$ height. Then, the type is estimated according to these top and base values. Symbol " $\cup$ " in this expression has type "$X + Y + 0.5Z$."

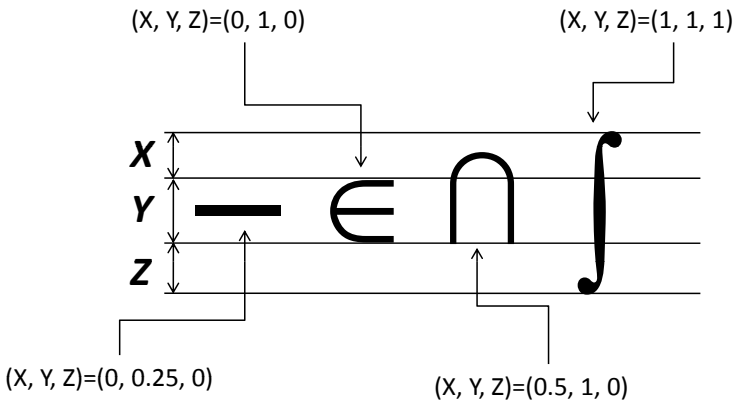


Figure 3: $X$, $Y$, and $Z$ regions.


Figure 4: Example of the type for symbol " $\cup$ ."

### 3.1.2 Specifying $X$, $Y$, and $Z$ regions

On setting the types for each symbol, we must know the height of the $X$, $Y$, and $Z$ regions. In order to calculate the $X$, $Y$ and $Z$ heights of a document, the heights of the $X$, $Y$ and $Z$ for each baseline character of a document are first measured. At the measurement, we need to refer to its entity (i.e., recognition result). For example, if the entity name of a character is "A," the height of the $X$ and $Y$ regions are measured. Then the measured heights are averaged to calculate the $X$, $Y$ and $Z$ heights for all the baseline characters of the document. Note that the $X$, $Y$ and $Z$ heights are different in each document.

The $X$, $Y$ and $Z$ heights for non-baseline characters is also estimated in the same way. This is because they often have their own $X$, $Y$ and $Z$ heights (which is slightly different from the $X$, $Y$ and $Z$ heights of the baseline characters) due to their own font shapes [§].

## 3.2 Context features

Although basic features are important in the classification task, they are insufficient for some cases. For example, the symbols in Fig. 2 can not be classified directly using only basic features; the non-baseline symbols are closer in their height and position to baseline symbols. The features of the

[§]Readers may be confused by the fact that we need to discriminate between baseline characters and non-baseline characters for estimating their own $X$, $Y$ and $Z$ heights during the process toward our final goal, i.e., the classification task. For this discrimination we used a predetermined $X$, $Y$ and $Z$ heights which is calculated from all characters contained in the database. Of course, the result from this discrimination includes some errors. These errors do not affect the estimation seriously because we use the average of the heights.

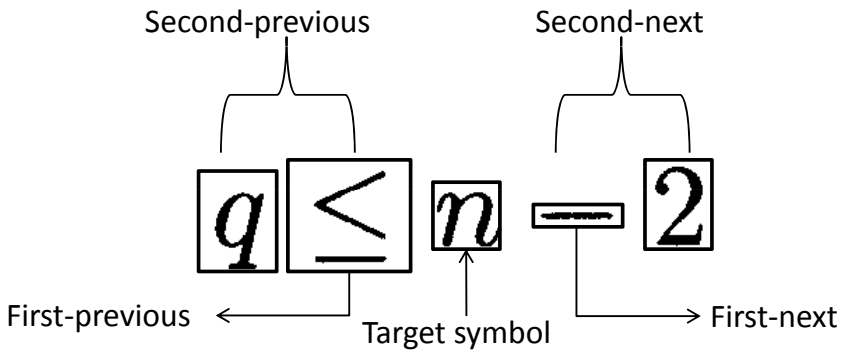


Figure 5: Example of context symbols for the target symbol "$n$."

symbols around the target symbol is very important in these cases. These features are called context features. Figure 5 shows the context symbols for symbol "$n$."

Context features are very important in the classification task; especially for mathematical symbols such as "$-$,""$+$," and "$*$." The basic features cannot classify these mathematical symbols exactly; their basic features overlap in baseline and non-baseline classes and, therefore, we used context features. These features include basic features for the symbols around the target symbol. As will be shown in the experimental results, these context features affect the classification accuracy.

# 4 Experimental Results

## 4.1 Database

The classification task was conducted on 229,898 target symbols. This huge number of symbols was extracted from two large databases, InftyCDB-1 [16, 17] and InftyCDB-2 [18], which together consist of 65 English articles (published between 1949 and 2000), 4 French articles (published between 1974 and 1988), and 7 German articles (published between 1956 and 1987) on pure mathematics. The total number of pages in the databases is 908.

To the authors' best knowledge, these databases are the largest of those used in past attempts on the classification task. For example, they are larger than the database used in [19], which consists of 297 pages. Such large databases are well suited to derive general properties of mathematical expressions and thus also suited to design the classifier for the baseline information.

All the mathematical expressions in the database were used, except matrices and fraction expressions. In these two types of expressions, the size of font is often very irregular and thus they will distort our results. Matrices and fraction expressions can be detected easily and treated separately from other mathematical expressions.

## 4.2 Classification accuracy

We used a binary SVM classifier to classify symbols into baseline and non-baseline symbol. To classify symbols, SVM determines a hyperplane which separate the two classes with maximum margin between the the vectors of the two classes. More details about SVM can be found in [20]. We use the LibSVM software [21] to train SVM classifiers. We construct SVM with linear kernel.

Table 1: Classification rate(%).

| Category name | Without context features | | With context features |
|---|---|---|---|
| | without type | with type | |
| Accent | 93.39 | 90.80 | 96.23 |
| Arrow | 95.84 | 96.00 | 99.20 |
| Big-symbols | 99.93 | 99.93 | 99.70 |
| Characters | 97.80 | 99.29 | 99.42 |
| Numeric | 97.71 | 98.30 | 99.42 |
| Operator | 93.43 | 93.51 | 98.57 |
| Others | 89.66 | 91.56 | 96.92 |
| Parentheses | 96.69 | 99.72 | 99.60 |
| Point | 92.69 | 98.42 | 99.65 |
| Average of accuracy | 94.98 | 98.04 | **99.25** |

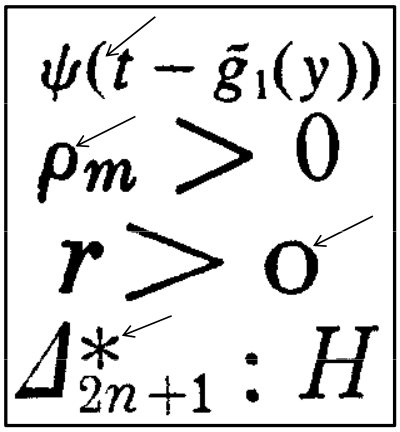


Figure 6: Example of misclassified symbols.

In the following table, we used 5-fold cross validation for the evaluation of the accuracy. The accuracy rate is evaluated for each category. We have 10 categories in the proposed method.

Table 1 shows the accuracy rate without using the context features. The accuracy rate is low in this case. From this table, we notice the effect of using symbol types in basic features as the accuracy rate is increased from 94.98 % to 98.04%.

Table 1 shows also the accuracy rate using the context features. The accuracy rate is increased in this case. From this table we notice that the effect of context features varies according to the category type. For example, the operator category is affected very much by the context features; its accuracy rate increased from 93.51 to 98.57 % when using the context features; its baseline classification depend on its preceding and succeeding symbols. In contrast, the parentheses category was not effected by the context features; its baseline classification does not depend on its preceding and succeeding symbols. This table shows the importance of using context features as the accuracy rate reached 99.25%. This accuracy rate was achieved with context of one symbol before and one after the target symbol.

Figure 6 shows an example of misclassified symbols in Table 1. A closer investigation of the misclassified symbols revealed that, (i) most of them occur in only two documents; these documents have a very special format. For example, the baseline parentheses in the first expression and the baseline symbol "$\rho$" in the second expression have very small height and they will be classified as non-baseline class. (ii) Some errors are due to the irregularity of symbol and characters. The details of irregular symbols and characters are found in [22]. For example, the baseline "0" in the third expression has very small height and it occupied $Y$ region which differs from other files; normally, numeric symbols occupy $X$ and $Y$ regions. (iii) Some errors due to some symbols such as "$-$, " and "$*$; " these symbols have often difficult characteristics. For example, the non-base symbol " $*$ " in the last expression has large height and low position and will be classified as baseline symbol.

# 5 Conclusion

In this paper, each symbol is classified into baseline and non-baseline classes. To deal with the large variation of symbol sizes, the symbol type and the context features were used. A SVM classifier was used to classify each symbol into baseline and non-baseline class.

The performance of the classification task over very large databases is very high as it reached 99.25%. In this task, we emphasize the importance of both the symbol type and the context features to improve the performance.


## Acknowledgment

The authors deeply thank the anonymous reviewers for their kind comments.


# References and Notes


[1] K. Chan, and D. Yeung, "Mathematical expression recognition: A survey," Int. Journal Document Analysis and Recognition, vol. 3, pp. 3–15, 2000.

[2] R.H. Anderson, "Syntax-directed recognition of handprinted 2-D mathematics," pp. 436-459, Ph.D. Dissertation, Harvard University, Cambridge, 1968.

[3] M. Okamoto, and B. Miao, "Recognition of mathematical expressions by using the layout structure of symbols," Proc. 1st Int. Conf. Document Analysis and Recognition, pp. 242–250, 1991.

[4] H. Twaakyondo, and M. Okamoto, "Structure analysis and recognition of mathematical expressions," Proc. 3rd Int. Conf. Document Analysis and Recognition, pp. 430–437, 1995.

[5] H. J. Lee, and J. S.Wang, "Design of a mathematical expression system," Proc. 3rd Int. Conf. Document Analysis and Recognition, pp. 1084–1087, 1995.

[6] J. Ha, R. M. Haraalick, and I. T. Phillips, "Understanding mathematical expressions from document image," Proc. 3rd Int. Conf. Document Analysis and Recognition, pp. 956–959, 1995.

[7] H. J. Lee, and M.c. Lee, "Understanding mathematical expressions using procedure-oriented transformation," Int. Journal Pattern Recognition, vol. 24, pp. 447–457, 1994.

[8] A. Grbavec, and L. Pottier, "Mathematical formula recognition using graph grammar," Proc. Society of Photo-Optical Instrumentation Engineers, pp. 44–52, 1998.

[9] S. Lavirotte and L. Pottier, "Optical formula recognition," Proc. 4th Int.Conf. Document Analysis and Recognition, pp. 357–361, 1997.

[10] D. Blostein, and A. Grbavec, "Recognition of mathematical notation," In Handbook of Character Recognition and Document Image Analysis, World Scientific, pp. 557–582, 1997.

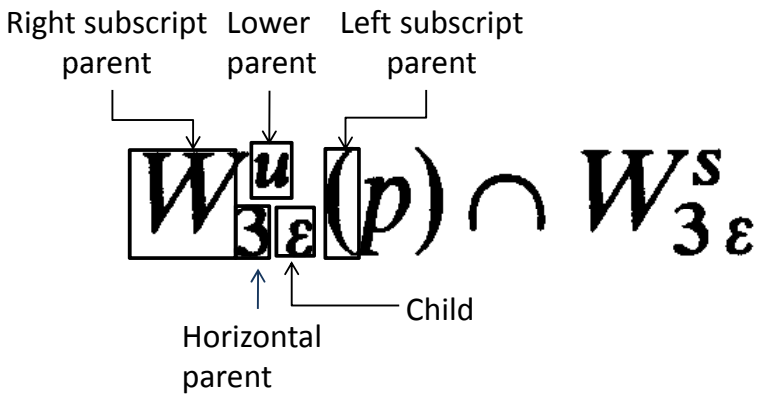


Figure 7: Example of the candidate parents for child "$\varepsilon$."

[11] U. Garain, and B. B. Chaudhuri, "A syntactic approach for processing mathematical expressions in printed documents," Proc. Int. Conf. Pattern Recognition, pp. 523–526, 2000.

[12] Y. Guo, L. Huang, C. Liu , and X. Jiang, "An automatic mathematical expression understanding system," Proc. 9th Int. Conf. Document Analysis and Recognition, pp. 719–723, 2007.

[13] R. Zanibbi, D. Blostein, and J.R. Cordy, "Baseline structure analysis of handwritten mathematics notation," Proc. 6th Int. Conf. Document Analysis and Recognition, pp. 768–773, 2001.

[14] J. -Y. Toumit, S. Garcia-Salicetti, and H. Emptoz, "A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents," Proc. 5th Int. Conf. Document Analysis and Recognition, pp. 119–122, 1999.

[15] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation," Int. Journal Pattern analysis and machine intelligence, vol. 24, pp. 1455–1467, 2002.

[16] M. Suzuki, S. Uchida, and A. Nomura, "A Ground-truthed mathematical character and symbol image database," Proc. 8th Int. Conf. Document Analysis and Recognition, pp. 675–679, 2005.

[17] S. Uchida, A. Nomura, and M. Suzuki, "Quantitative analysis of mathematical documents," Int. Journal Document Analysis and Recognition, vol. 7, pp. 211–218, 2005.

[18] M. Suzuki, C. Malon, and S. Uchida, "Databases of mathematical documents," Research Reports on Information Science and Electrical Engineering of Kyushu University, vol. 12, pp. 7–14, 2007.

[19] U. Garain and B. B. Chaudhuri, "A corpus for OCR research on mathematical expressions," Int. Journal Document Analysis and Recognition, vol. 7, pp. 241–259, 2005.

[20] V. N. Vapnik, "Statistical Learning Theory: A Primer," Int. Journal of Computer Vision, vol. 38, pp.9–13, 2000.

[21] C. C. Chang, C. J. Lin, LIBSVM: A library for support vector machines. ¡http://www.csie.ntu.edu.tw, 2001.

[22] A. Walaa , S. Uchida, M. Suzuki, "Automatic Classification of Spatial Relationships among Mathematical Symbols Using Geometric Features," Proc. 10th Int. Conf. Document Analysis and Recognition, Int. Journal the Institute of Electronics, Informations, and Communication Engineers, vol. E92-D, 2009

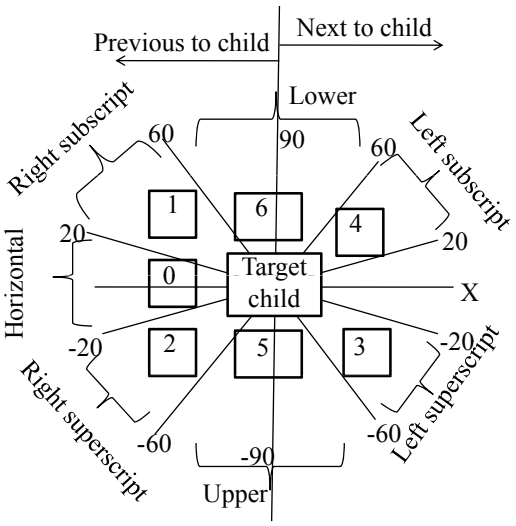


Figure 8: Angle between the child and its candidate parents.

# Appendix

# A Determine the parent of each child using SVM and baseline information

## A.1 Outline

An SVM classifier also can be used with the baseline information to determine the parent of each child. For example, in Fig 7 symbol "$\varepsilon$" will have symbol "3" as parent from candidate parents. Selecting the parent for each symbol and the baseline information are important to specify the special relationship among symbols, the details about the special relationship can be found in [22].

The selection of the parent for each child from the candidate parents is done by the following steps. The symbols of mathematical expression are first ordered according to the leftmost bounding box. Then, the type of each symbols is estimated; the details about symbol types can be found in Section 3.1.1. After that, a normalized bounding box is evaluated for each symbol. The details about the normalized bounding box can be found in [22]. After that, the angle between the center of the normalized bounding box of the child and the center of the normalized bounding box for each symbol in the expression is calculated. Then, the first symbols, which satisfy the angle condition as illustrated in Fig 8, are selected as candidate parents. After that, the features are extracted from the child and its candidate parents. Finally, the SVM with quadratic kernel is used to select the parent for the child among 7 candidates (at most). Experiments using very large databases show high efficiency of the proposed method with the extraction accuracy reaching 98.54%.

Table 2: Classification rate(%) .

| | |
|---|---|
| basic features only | 98.47 |
| with baseline | 98.48 |
| with angle | 98.47 |
| with $H, D$ | 98.49 |
| with baseline and $H, D$ | 98.54 |

## A.2 Feature extraction

We will divide features into basic features and additional features. Basic features include width, height for child and height for each of the candidate parent, and the horizontal distance between the child and each of the candidate parent. Additional features include baseline information for child and candidate parents, the angle between the child and the candidate parents, and relative size $H$ and relative position $D$ features. The details about $H, D$ features can be found in [22].

## A.3 Evaluation by cross-validation

An SVM classifier is used to select the parent of each child. In the experiment we have 187,415 children. We construct SVM with polynomial kernel with degree "2."

In Table 2, we used 5-fold cross validation for the evaluation of the accuracy. The accuracy rate is evaluated for each category. We have 10 categories in the proposed method. In these experiments we exclude all accent symbols because they have different characteristics which will confuse the classification task.

Table 2 shows the effect of using additional features in the proposed method. This table proves the importance of using baseline information and $H, D$ features as the accuracy improved to 98.54%.