データベースシステム 124−15
情 報 学 基 礎 62−15
（2001. 5. 22）

意味付き文字画像を用いた文献の電子化

石川 正敏† 波多野 賢治‡ 天笠 俊之‡ 吉川 正俊‡ 植村 俊亮‡ 勝村 哲也†

†島根県立大学 総合政策学部
‡奈良先端科学技術大学院大学 情報科学研究科

†{m-ishikawa, t-katsumura}@u-shimane.ac.jp
‡{hatano, amagasa, yosikawa, uemura}@is.aist-nara.ac.jp

Unicodeなどの既存の符号化文字集合は，文書の記述に必要な文字種の不足があるため文献の記述を正確に表現できないことがある．そこで本研究では，符号化文字集合の代わりに文字を表現する画像で文献を電子化する．しかし文字の表現に画像を用いた場合，計算機による文字の識別が困難になるため，本稿では画像に文字に関する情報を付加する．本稿では，このような画像を意味付き文字画像と呼ぶ．意味付き文字画像を用いた文書の記述には，文書の論理構造の記述に適したXMLを利用する．さらに本稿では，提案形式に従った文書に変換するシステムを試作し提案形式の改良点などについて考察する．

Digitizing Documents Using Imaged Characters with Semantical Information

Masatoshi Ishikawa† Kenji Hatano‡ Toshiyuki Amagasa‡ Masatosi Yosikawa‡
Shunsuke Uemura‡ Tetsuya Katsumura†

†Faculty of Policy Studies, The University of Shimane
‡Graduate School of Information Science, Nara Institute of Science and Technology

†{m-ishikawa, t-katsumura}@u-shimane.ac.jp
‡{hatano, amagasa, yosikawa, uemura}@is.aist-nara.ac.jp

Coded Character sets, such as Unicode and so on, have a problem of a lack of character types for digitizing documents. We propose a method to use imaged characters for digitizing documents, instead of coded character sets. Imaged characters having their semantical information themselves are a useful tool, because they are recognizable by a computer. We use XML for describing electronic documents using imaged characters, because XML is suitable for describing a logical structure of a document. We tested a prototype system of converting plain texts into imaged characters, and discussed problems of the prototype for the future improvement.

1 はじめに

電子図書館のサービスの一つに既存文献の電子化が挙げられる．また，Unicode[1] の普及により計算機上で様々な言語で記述された文献の閲覧が可能になりつつある．特に近年，東アジア各国で，文献の電子化が盛んである．しかし，既存の符号化文字集合は，文献の電子化に必要な文字がすべて収録されていないため，文献の記述を正確に表現できないことがある．特に漢字は，新しい字の誕生や，過去の文献で使用された文字種の総数の調査が困難であるので，文字種の不足を解消できないと考えられる．

文字種の不足に対する対策としては以下のものが挙げられる．

(1) 代替文字列の利用
この方法は，文書の記述の正確さに欠けるが，符号化文字集合だけで文書を記述できる．しかし，文書を読むには，参照資料や記述規則について予め理解していなければならないので，利用者への負担が大きい．

(2) 文献全体を一画像として表示
この方法は，画像であるので文献の記述やレイアウトを正確に計算機上に再現できる．しかし，計算機による文字識別が困難である．また，一般に符号化文字集合を用いた文書よりファイルサイズが大きいので転送効率が悪い．

(3) 文字集合の拡張
この方法には，符号化文字集合を拡張する方法と，文字画像を文書に埋め込む方法がある．前者は，計算機処理に適した方法であるが，利用者ごとに異なる符号化文字集合ができるため，文書によっては，利用者ごとに文書の記述が異なることがある．後者は，文書交換と文字画像交換を同時にすることで利用者間で表示される文書の記述を一致させることができるが，文字画像による文字識別は，符号化文字集合より効率が悪い．

本稿では，使用する文字種が多い漢文を対象に，符号化文字集合を使用せずに文献を電子化する方法を提案する．その方法として，WWW 環境で利用ができると考えられる文字画像を用いて文献の電子化を行う．しかし，文字画像だけで文書を記述した場合，計算機による文字識別の効率が悪いので，各文字画像に対し明示的に読みなどの情報を付加する．本稿では，情報を併せもつ文字画像を意味付き文字画像と呼ぶ．意味付き文字画像を用いた文書の記述には，文書の論理構造の記述に適した XML[2] を用いる．意味付き文字画像によって，既存の符号化文字集合に依存しない文書記述と符号化文字集合と同様の計算機処理とを実現できると考えられる．また，本稿では，1 字以上の意味付き文字画像で構成する単語について情報を付加する．単語に情報を付加することで，意味付き文字画像の情報だけでは表せない詳細な情報を文書に付加できる．本稿の提案形式で記述される文書は XML 文書であるので，他の XML 関連技術が利用できる．例えば，XSL スタイルシートを用いた XML 文書から HTML 文書への変換が挙げられる [3]．さらに，本稿では，既存の電子文書を本稿で提案する文書形式に変換するシステムを試作し提案形式の利用や改良について考察する．

2 関連研究

2.1 電子文書形式

文書交換に適した形式として，米 Adobe 社は PDF (Portable Document Format) を提案している [4]．また，同社は PDF ファイルの閲覧のために Acrobat Reader を無償で提供している．PDF には注釈機能もあり，文書の製作者や閲覧者が自由に注釈をつけることができる．しかし，PDF は印刷を目的とした形式であり．必ずしもデータベースの格納に向いているわけではない．本稿の提案形式は XML 文書であるので，XML 関連技術を利用することでデータベースへの格納などを比較的容易に行うことができると考えられる．

Sakaguchi らは，クライアント側でフォントがなくとも多言語文書を表示するための形式として HTML を拡張した MHTML を提案している [5]．MHTML は文書中の個々の文字をビットマップ画像で表現している．MHTML で記述された文献は，専用の閲覧システムが必要であり利用者の環境によっては文献を閲覧できない場合があるが，本稿の提案形式は，一般的な WWW 環境での文献表示が可能である．

2.2 外字処理

WindowsNT 漢字処理技術協議会は，インターネットを介した外字交換手法を提案し仕様を公開している [6]．提案手法は，外字交換の書式にはXMLを利用しており，文書中に埋め込む文字画像の指定に関する書式を定義しているが，文字画像に対する付加情報についての制約は特にない．対して，本稿の提案形式は，文書中の文字に意味情報を積極的に追加するので，意味付き文字画像単体で文字識別が可能である．

2.3 XML

XML(Extensible Markup Language) は，1998年にWWWコンソーシアムが仕様を公開したマークアップ言語である [2]．XMLは，文書の論理構造を記述するための言語である．また，XML文書の処理系の実装が進んでいるため，XMLは様々な分野で利用されている．

3 文書形式

3.1 定義

まず，意味付き文字画像について定義する．意味付き文字画像は，文書を表示するための文字画像と，文字識別のため情報の組で表す．文字画像に付加する情報は，読みなどの辞書に掲載されている情報を用いる．意味付き文字画像は，Unicodeなどの符号化文字集合に依存しないので，文書の記述者が自由に文字を追加できる．

[定義 1] 意味付き文字画像
意味付き文字画像 *letter* は，文書表示に用いる画像 *img* と，計算機処理等で利用する情報 *letter_info* の組である．

$$letter = (img, letter\_info)$$

単語は，1字以上の文字列で一文字だけでは表せない複雑な意味を表す．例えば，“一期一会” は個々の文字を個別に読んでも意味をなさないが，全体を通して読むことで “生涯にただ一度しかない出会い” という意味を表す．従って，意味付き文字画像の情報だけの文書より詳細な情報を文書に付加できると考えられる．単語は以下のような形式で記述する．

[定義 2] 単語
単語は一文字以上の意味付き文字画像の列と，単語の意味の組で記述する．

$$word = (\{l_1, \cdots, l_n\}, word\_info)$$

$\{l_1, \cdots, l_n\}$ は，意味付き文字画像の列であり，$word\_info$ は，単語の意味である．

先に述べた通り，単語は意味付き文字画像の1次元列として表現する．また，単語をある決まった順序で並べることで文書が表現される．そこで，本稿では，文書構造を以下のように定義する．

[定義 3] 文書
文書は一つのルートを持ち，中間ノードが単語 *word*，葉が意味付き文字画像 *letter* である，高さ2の木として表現する (図1)．

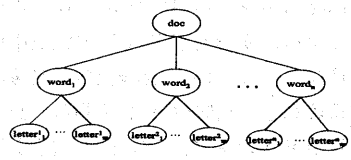

図 1: 文書構造

3.2 XMLによる記述

3.1で示した通り，本稿で提案する文書形式は木構造である．XMLは文書の論理構造を木構造で表現するので，本稿の提案形式を記述するのに適している．そこで，先に定義した意味付き文字画像，単語，文書のそれぞれに対応したDTDの例を以下に示す．

```
<!ELEMENT doc (word+)>
<!ELENEMT word (letter+, word_info)>
<!ELEMENT word_info (#PCDATA)>
<!ELEMENT letter (img,letter_info)>
<!ELEMENT img (#PCDATA)>
<!ELEMENT letter_info (#PCDATA)>
```

例では，文書のルートを要素 doc で示す．要素 doc は一つ以上の要素 word を子として持つ．要素 word は，単語を表し，一つ以上の要素 letter と，一つの要素 word_info を持つ．要素 letter は，意味付き文字画像を記述する要素であり，文字画像を表す要素 img と，文字の情報を記述する要素 letter_info を一つずつ持つ．本稿では，要素 img に文字画像ファイルの URL を記述する．また，URL などの文書中に複数回登場すると考えられる情報は，実態参照を用いて記述を簡単にできる．

4 試作システム

4.1 システム概要

試作システムは，主に文書形式変換，文書管理，文書整形の機能を持つ．試作システムの主な構成は以下の通りである (図 2)．また，利用者による文献閲覧には，WWW ブラウザの利用を想定している．

(1) 変換機
変換機は，与えられた文書の提案形式に従った文書への変換及び単語や文字に関する情報の追加をする．また，変換対象の文書が XML 文書の場合，意味付き文字画像などの追加する要素を他の XML 処理系でも使用可能にするために DTD や XSL スタイルシートの書き換えもする．

(2) XML 文書データベース
データベースは，変換機によって生成された XML 文書を管理する．データベース技術を利用すれば文献検索などのサービスを提供できるが，本稿のシステムでは XML 文書をファイル形式で保存する．

(3) 表示形式変換機
表示形式変換機は，利用者から閲覧要求のあった XML 文書を HTML 文書に変換し結果を WWW サーバに渡す．クライアントである WWW ブラウザには，同様の処理ができるものもあるが，本稿では利用者の環境に依存しない表示環境を提供するために，サーバ側で処理をする．

(4) 辞書データベース，意味付き文字画像データベース
これらのデータベースは，単語や文字に関する情報を提供する．また，意味付き文字画像データベースは，変換後の文書の表示に必要な文字画像を提供する．

(5) WWW サーバ
文書閲覧者との通信を処理する．

(6) テンプレート
DTD もしくは XSL スタイルシートを書き換えるための雛型である．

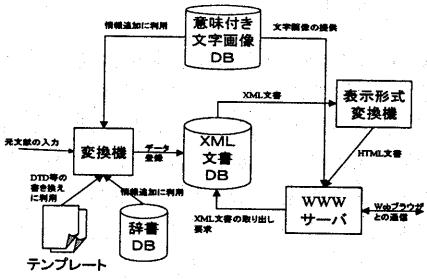

図 2: 試作システム構成

4.2 文書変換処理

本稿の提案形式を用いた文書は，XML 文書であるので，文書の表示には，既存の XSL 処理系が利用できる．本稿では，Cocoon [7] を利用する．従って，本節では，与えられた文書を本稿の提案形式に従った文書への変換処理について述べる．本稿では，与えられる文書を XML 文書であると仮定する．文書形式の変換には，以下のような処理が必要である．

(1) DTD，XSL スタイルシートの書き換え

(3) 意味付き文字画像などの要素の挿入処理

(2) 意味付き文字画像及び単語に関する情報の検索

4.2.1 DTD, XSL スタイルシートの書き換え

変換対象の文書が XML 文書である場合，DTD などの定義ファイルの書き換えは，本稿の提案形式で記述した XML 文書を他の処理系でも利用可能にするための処理である．

これら定義ファイルの書き換えでは，まず意味付き文字画像や単語の要素を挿入する親要素を指定する．このような指定は，人手によって処理する．要素指定後の書き換え処理は機械的にできると考えられる．

以下に，要素の書き換え処理について述べる．

(1) それぞれのファイルに対して予め作成した挿入要素に関するテンプレートを追加する．

(2) 挿入対象である親要素に追加したテンプレートの参照情報を書き加える．例えば，DTD の場合，挿入元の要素に定義された “# PCDATA” 部分を “word+” に書き換える．

(3) 該当する要素に対して書き換えが終われば処理を終了する．

4.2.2 意味付き文字画像などの要素の挿入処理

本処理は，4.1 に指定した挿入元の要素に対して，単語要素や意味付き文字画像要素及びそれぞれの要素に関する情報を挿入する．変換対象の文書は XML 文書を仮定しており，変換処理によって出力される文書も XML 文書である．

(1) 変換対象の XML 文書の読み込み及び DOM 木への変換をする．

(2) DOM 木の各ノードを巡回し，テキストノードがあれば，内容を抜き出す．

(3) 次節で述べる意味付き文字画像及び単語に関する情報の検索を行い，その結果に従った要素の生成とテキストを抜き出した親要素に生成した要素を挿入をする．

(4) 該当するすべてのテキストノードに対して処理が終了するまで，(3) を繰り返す．

4.2.3 意味付き文字画像及び単語に関する情報の検索

与えられた文字列に対して，意味付き文字画像の要素及び，単語要素の生成に必要な意味情報の検索をする．まず，意味付き文字画像の検索の手順を以下に示す．

(1) 与えられた文字列の先頭から順に 1 文字ずつ，取り出す．

(2) 取り出した文字をキーに意味付き文字画像データベースに対し問い合せる．

(3) 意味付き文字画像データの検索結果から，文字の意味と，文字画像の URL を取り出し意味付き文字画像要素を生成する．

(4) 与えられた文字列のすべての文字に関して (1) から (3) の処理を繰り返し，未処理の文字がなくなれば終了．

次に，単語に関する情報の検索の処理の手順を以下に示す．

(1) 与えられた文字列の先頭にポインタを設定する．

(2) ポインタから n 文字 (n : 1 から与えられた文字列の長さまで) の部分文字列を取り出す．

(3) 取り出した文字列をキーに辞書データベースに問い合せる．

(4) (3) の検索キーと一致するものがあれば，検索キーを単語として判定し，検索キーである部分文字列と検索結果を組とした単語要素を生成する．また，ポインタの位置を部分文字列の最後尾の次の位置に移動させ，処理 (2) を繰り返す．

(5) 検索結果が空であれば，文字列から取り出す部分文字列の長さを一つ長くし，処理 (2) から繰り返す．

(6) 部分文字列の長さが文字列の長さを越えても検索結果が得られなければ，部分文字列の先頭を表すポインタの位置を一つずらす．その際，元のポインタの位置に示されていた文字を単語として要素を生成する．

(7) ポインタが文字列の最後尾までくれば処理を終了する.

4.3 変換例

例として，以下のようなXML文書を本稿の提案形式に従った文書に変換することを考える.

<例文>花が咲く．</例文>

この時，例えば辞書に“花”と“咲く”が登録されていたとすると変換結果は以下のようになる.

```
<例文>
  <word>
    <letter>花</letter>
    <word_info>...</word_info>
  </word>
  <word>
    <letter>が</letter>
    <word_info/>
  </word>
  <word>
    <letter>咲</letter>
    <letter>く</letter>
    <word_info>...</word_info>
  </word>
  <word>
    <letter>．</letter>
    <word_info/>
  </word>
</例文>
```

“花”と“咲く”以外の“が”や“．”のような辞書に登録されていない文字も，文書を構成しているので単語として扱う．しかし，“が”などに併せて挿入される要素word_infoは，空である．また，各文字は要素letterの子になっている．例文では記述を簡単にするため要素letterの子の要素の記述は省略した.

5 実験

5.1 実験環境

本稿で試作したシステムは，CPUがPentiumIII (500MHz)，メモリが256Mbyte, OSがWindows NT4.0であるPC上で実装した．また，システムの作成には，Java1.3を使用し，辞書データの管理にはOracle 8.1.6を使用した.

本稿の実験で用いたデータは以下の通りである.

(1) ekanji[8]
ekanjiは，出典ごとにまとめられた約87000字の漢字フォント集合と漢字に関する情報を公開している．本稿では，文字画像に諸橋大漢和辞典及びUnicode2.0に従った漢字集合を用いた.

(2) 大正大藏經テキストデータベース[9]
大正大藏經テキストデータベース研究会が独自の書式に従ったテキストとして電子化している仏典集である．本稿では，変換対象の元データとして阿含部vol.1-2をページごとに分割しXML文書に変換したものを用いた.

(3) 漢韓日・英・辭典[10]，Digital Dictionary of Buddhism[11]
A. C. Mullerが公開している文学や仏教などに関する辞書データである．本稿ではこれら辞書データから機械的に抽出できた28772語をデータベースに登録した.

5.2 文書形式変換処理時間

文書形式変換処理時間の大半は，データベースへの問い合わせ処理である．特に単語についての問合せは，文字列の長さを動的に変えて繰り返し処理するので多くの処理時間を必要とする．従って，文字列から取り出す単語の長さを制限することでに処理時間を短くできる．そこで，本節では，与えられた文字列から取り出す単語の長さ制限と処理時間の関係を調べる(図3)．また，生成されるXML文書のファイルサイズを計測し(図4)，両方の結果から妥当な単語の長さの制限を調べる．前者は，100個のファイルの処理にかかった時間を測定し，後者は100個のファイルの平均を測定した.

図3と図4から文書変換処理は，単語の長さ制限の大きさに比例して処理時間が長くなっているが，生成されたファイルのサイズは単語の長さ制限が6文字から変化がなくなる．これは，本稿で登録した辞書データにおいて6字以内の長さの単語が全体の98%を占めていることから，7文字以上の長

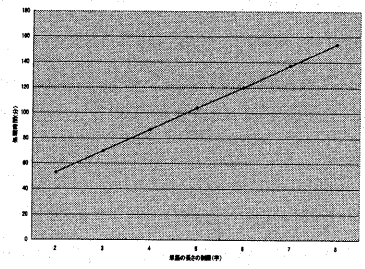

図 3: 単語の長さ制限と処理時間

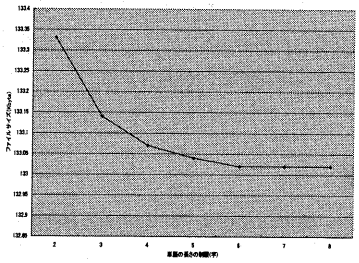

図 4: 単語の長さ制限とファイルサイズ

さ単語を文書中から発見する頻度が非常に低いためだと考えられる．また，本稿のシステムにおけるファイルサイズの変化は，文書に追加される情報より適切な単語が取り出されないため追加される不適切なタグによる影響の方が大きいためだと考えられる．従って，本稿のシステムでは，単語の長さ制限は 6 文字が適当であると考えられる．

5.3 表示

図 5 は変換前の XML 文書と XSL スタイルシートを用いてブラウザ上で表示した結果である．図 6 は本研究の提案形式に変換した文書と XSL スタイルシートを組み合わせてブラウザで表示した結果である．元文書では，外字を表すのに実体参照を表す記号が用いられているが，変換後は外字部分が対応する漢字に置き換わっているため，可読性が良くなっていることがわかる．また，図 6 では，JavaScript を利用して文中の単語 “阿含” についての意味情報も表示している．

このような処理は，文献を読むことに対する負担を減らすことができるので，図書館などの広く文献を公開する場所での利用が有効であると考えられる．

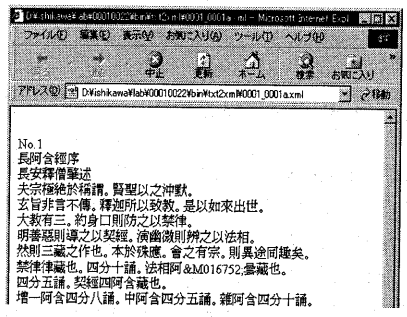

図 5: 元文書

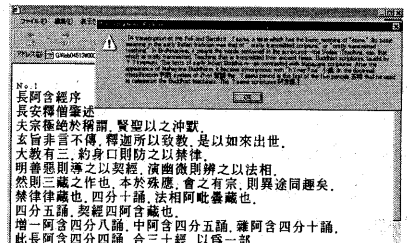

図 6: 変換後の文書

6 考察

6.1 文書形式変換処理について

本稿のシステムにおいて，与えられた文書から単語を取り出す方法は，機械的に取り出した文字列と辞書データとの照合をしているにだけである．

一般に古文書は，現代の語と用法が異なる部分もあるので，既存の構文解析器を利用しても十分な結果が得られないことがあると考えられる．このような場合，本稿で用いた手順による単語の取り出し処理をした後に，専門家による単語区分の修正処理をしなければならない．そのためには，本

稿で提案した形式で記述された文書を修正するためのオーサリングシステムが必要である．

その他，本システムでは，辞書データベースを変換システムと同一サーバ内で作成したが，すべての辞書データを管理することは困難である．従って，ネットワーク上に公開されているオンライン辞書の利用を考慮しなければならない．

6.2 文書形式の改良

本研究の提案形式では，元の文書に要素を追加するだけでなく文書の表示に画像を用いるので，文書の転送に元文書より多くの帯域幅が必要である．様々な利用者の環境に対応するには，できるだけ狭い帯域幅での文書転送を実現する必要がある．従って，提案形式で記述した文書の最適化や圧縮等を検討する．本節では，文書の最適化の方針について述べる．

本稿で提案した形式では，文書中の文字を逐一，意味付き文字画像に置き換えている．一般に，文書中には同じ文字が複数回登場するので，意味付き文字画像も複数回登場する．従って，提案形式は冗長であるといえる．そこで，先に述べた冗長な記述を実態参照を用いて削減する．さらに文書中の個々の意味付き文字画像をXML文書として個別に扱い．本文を操作するときにそれらのXML文書を自動的に取り込むようにすれば文書の記述が実態参照を用いたものより冗長性を削減できると考えられる．冗長性の削減によって文書のファイルサイズが小さくなり，文書の転送コストが削減できる．また，冗長性を減らすには，文書中に用いられている文字や単語の情報を予め抽出する必要があるため，索引の作成や文字調査などに利用できると考えられる．

7 まとめ

本稿では，文献の記述を正確に電子化するために，意味付き文字画像を用いる文書形式を提案した．意味付き文字画像は，文字画像に意味情報を追加したものである．従って，文字画像だけでは困難な文字識別を計算機で処理できると考えられる．その他に，文書中の単語についても意味情報を追加することで文書の読解の支援や文書の内容に基づいた柔軟な検索に応用できると考えられる．

本稿で提案した文書形式の記述にはXMLを用いた．XMLを利用することで様々な既存技術の利用できるため，文書形式の変換システムや文書閲覧システムの実装が比較的容易できる．他に，本稿では，XML文書を本稿の提案形式に従った文書へ変換する方法について述べた．また，与えられた文書を本稿の提案形式に変換するシステムを試作し，処理の改善と提案形式の改良について考察した．

今後は，変換対象文書に対応した構文解析機を用いることで，より精度の高い意味情報の付加を実現する必要がある．また，文書に追加した情報を利用した検索方法を考察し，提案形式の実用性について評価する．他に紙媒体などで記述された文献を対象に，スキャナ等でそれらを取り込んだ上で，文字画像の抽出と，本稿の提案形式に従った電子文献の変換システムの構築を目指す．

参考文献

[1] The Unicode Consortium: *The Unicode Standard 3.0*, Addison Wesley, Apr. 2000

[2] WWW Consortium:"Extensible Markup Language (XML)", http://www.w3.org/XML/

[3] WWW Consortium:"Extensible Stylesheet Language (XSL)", http://www.w3.org/Style/XSL/

[4] アドビシステムズ社:"Adobe PDF", http://www.adobe.co.jp/products/acrobat/adobepdf.html

[5] Tetsuo Sakaguchi, Akira Maeda, Takehisa Fujita et.al:"A browsing tool of multi-lingual documents for users without multi-lingual fonts",*Proceedings of the 1st ACM international conference on Digital libraries*,pp63 - 71, 1996

[6] WindowsNT 漢字処理技術協議会:"XKP GAIJI 交換仕様", http://www.xkp.or.jp/next/XKPGAIJI100.htm

[7] Apache XML Project:"Cocoon", http://xml.apache.org/cocoon/index.html

[8] 勝村哲也:"ekanji", http://www.zinbun.kyoto-u.ac.jp/ ekanji/ http://nohara.u-shimane.ac.jp/ekanji/

[9] 大蔵経テキストデータベース研究会:大正新脩大藏經テキストデータベース, http://www.l.u-tokyo.ac.jp/ sat/japan/index.html

[10] A. Charles Muller:" 漢韓日・英・辭典", http://www.human.toyogakuen-u.ac.jp/ãcmuller/dicts/dealt.htm

[11] A. Charles Muller:"Digital Dictionary of Buddhism", http://www.human.toyogakuen-u.ac.jp/ãcmuller/dicts/ddb-intro.htm