System for Markup and Retrieval of Texts (SMART): An Update

Christian Wittern
Chung Hwa Institute of Buddhist Studies
Taipei, Taiwan

At the EBTI Kyoto meeting in October 1997, a plan for the System for Markup And Retrieval of Texts (SMART) has been first introduced. In the meantime, with a grant from the German Science Council (DFG), a fulltime pursuit of research and development will be possible for the period of two years.

In this presentation, I report on the progress achieved so far. Many of the practical assumptions have been challenged since the inception of the project, but the most basic aim is still as much needed as ever: A convenient tool to make the fruits of TEI based markup available to the researcher: Complex queries based on conditions derived from structural or content markup, interactive semi-automatic markup by writing the results back to the texts in some form of markup, and many more fancy things. I will look at some of the more technical issues and then continue to some practical considerations and demonstrations.

Format of the Index Files

Overview

As the name suggests, retrieval plays a very important part in SMART. Retrieval is based on a new type of index, that places a layer of abstraction between the actual locations in electronic documents and the positions in the texts referenced.

The SMART project tries to develop a suite of general tools for work on East Asian texts. Some assumptions are made about the texts:

- They should be encoded in XML according to the TEI guidelines.
- The milestone tags (<pb>, <lb>) down to the line level.

Retrieval in SMART is based on index files. These files serve two separate purposes:

- Allow for fast access to any portion of the text and provide some basic context information independently of the format.
- Reconstruct a flat version for each of the witnesses, that are recorded as variant readings in the XML files.

While the first point might be self-explanatory, I will give some explanation of the second point. Figure 1 shows a short extract of a text in XML format. Textual variants are surrounded by the <app> element, with the text version of the base text (Taisho) marked with the <lem> element, other versions are marked with the <rdg> element, the attribute wit indicates the name of the witness.

```
<pb ed="T" id="T10.0280.0445a" n="0445a"/>
<lb n="0445a01"/><head type="no">No. 280 [Nos. 278(3, 5), 279(7, 9)]</head>
<div1 type="jing">
<lb n="0445a02"/><head><title>佛說兜沙經</title><app n="044501"><lem>一
</lem><rdg wit="【三】【宮】">&lac;</rdg></app></head>
<lb n="0445a03"/>
<lb n="0445a04"/>
  <byline>後漢
    <app n="044502">
      <lem>月氏</lem>
      <rdg wit="【宋】【宮】">&lac;</rdg>
      <rdg wit="【元】【明】">月支</rdg>
    </app>三藏
    <app n="044503">
      <lem>&lac;</lem>
      <rdg wit="【三】【宮】">法師</rdg>
    </app>
  支婁迦讖譯</byline>
<lb n="0445a05"/><p>一切諸佛威神恩。諸過去當來今現在亦爾。
<lb n="0445a06"/>佛在摩竭提國時。法清淨處。其處號曰在所
<lb n="0445a07"/>問清淨。始作佛時。光<app n="044504"><lem>景</lem>
<rdg wit="【明】【宮】">影</rdg></app>甚明。自然金剛蓮
```

Figure 1 Part of a TEI encoded text

As can be seen from this example, some characters (marked with the <rdg> element) fall out of the sequential structure of the text, they can be seen as stacked on top of the primary version of the text (marked with the <rdg> element). For the building of the index, additional lines will be inserted into the index, to account for these additional characters. An example is shown in Figure 2.

In this example, an index with two additional characters has been buildt, the meaning of the numbers to the left of the characters is explained below.

The index consists of three or four different data files:

- A dictionary file.
- A binary index to the dictionary file.
- A file that records mappings between the text identifiers and filenames.

- One or more files that record mappings between the logical and physical location of lines. This is not always necessary.

Figure 2 Some lines from the generated index table

Format of the dictionary file

The dictionary file simply contains the entries to be indexed one per line. The encoding is in UTF-16 (which is identical to UCS-2, the most popular Unicode encoding for most practical purposes). Mojikyo characters (that is, characters defined by the Mojikyo Font Institute[1]) are using the Shift-Mojikyo[2] encoding developed by Mr. Masahiko Maedera. The dictionary entries are separated by a tab character from the character addresses (locations), which are then following up to the end of line.

[1] See the homepage of the Mojikyo Font Institute at http://www.mojikyo.gr.jp
[2] A detailed explanation of this encoding method and the reason it become necessary is available at http://hp.vector.co.jp/authors/VA002891/TMED_EN/TMFMT3.TXT

The file begins with the byte order mark FFFE followed by DIC and a digit indicating the format type (eg. 1) and a line-end (0D000A00) sequence.

Format 1:
The index length of the index term is variable, it is terminated by a tab character (0900). Following is a sequence of 6 byte location addresses, up to the end of the line.

The last adress is followed by a newline character. The adress is always given for the first Character; if preceding characters are given, for e.g. KWIC they should be appended at the end,separated from the other characters by a comma (2C00) character.
Example:
[U4e00][U4e00][U4e01] 00b4e8c6e63300b4c6e933e7 2 instances
...
Format of the character address:
The logical position of each character in the text, along with some other information, is expressed with the character address (I will call this location from now on). This location consists of a 15 digit decimal number, which encodes this information in its positions. Lets look at an example first:
The first character in the line T10n0279_p0001a01 is expressed as 000100279000000110101, which breaks down to:

T10n0279_p0001a01 =	010 0001 10101 000	
	010	Volume = 010 (maximum: 280 volumes)
	0001	Page = p0001
	1	Cataster = a
	01	Line = 01
	01	Character = n/a

0	RDG = 0=T, 1=S, 2=Y, 3=M 4=3, 5=G,9=others
0	BaseTag = 0=P, 1=L, 2=Z, 3=J, 4=Byline, 5=Comm, 6=Note
0	ExtTag = user defined... Name, Yin, etc.

As can be seen, the digit breaks down to different units, which have different semantics assigned to it. The first 12 digits are used to encode the character location in the text, expressed in the units of volume (3 digits), page (4 digits), cataster (1 digit), line (2 digits) and character number on the line (2 digits). These units are of course completely arbitrary and they can be redefined as needed.

The last three digits, are defined below the horizontal bar, they provided some information about the context of the character in question, derived from the XML markup. Provided is information about whether the character is

Alternatively, the cataster number can be used as a flag. I am using it now to indicate whether the character is the last in the line[3]: For this purpose, the value of the cataster indicator is increasing in increments of 3. Adding 1 to the original value will indicate the last character of a line. This is important to be able to calculate the distance in the text of two characters for word-clustering and approximity searches.

Format of the index file

This is a binary (non-text) file consisting of long (i.e. four-byte) integers. These are offsets to the starting byte of the entities contained in the dictionary file. The offsets are sorted according to the lexical value (i.e. collating sequence) of the tokens to which they point.

This index format is modelled after a format used by Jim Breen for his EDICT dictionary.

List of texts

Gives first (and last?) lines of the texts, followed by text title and any other information. A hierarchy of lower level textual elements can be constructed by indenting the file with one space/tab for every hierarchical level.

Mappings to file-offsets

This gives the position of the beginning of each line and the filename, where this is found. The receiving application might have to strip some markup, the beginning is the beginning of the physical line in the file, not the logical line of the text.

[3] For searches with defined ranges of proximity, it might be desirable to calculate the distance in character positions. For this purpose, a flag indicating the line end is needed.

Other considerations

Determining the optimal size of the index

The size of the index depends largely on the number of characters included besides the lookup character. To estimate the size, a set of dictionary files of 1 to 7 characters has been calculated for one volume. Additionaly, complete indices have been created for a dictionary of 3 and 7 characters length. A larger size means faster response time and an immediate KWIC-like view. If this is not done and a KWIC is needed, it would have to be constructed on the fly, which is more time-consuming.

One volume has approx. 1 500 000 characters.
In strings of different length, they are approximately distributed as follows:

Length of entry	Lines in index	Size of dictionary (1 Vol)	Size of index (28 Vols)
1	3814	16 487	
2	174208	1 049 942	
3	575164	4 610 411	313 499378
4	838242	8 396 379	
5	970633	11 666 525	
6	1034526	14 507 096	
7	1074590	17 221 803	916 231744
Total	1504517	--	--

Table 1 Length of the text entry and size of index

Normalization of characters

The index is based on Unicode, while the source files might be in Big5, JIS or some other encoding. While the conversion to Unicode is no problem for most characters, there is a small number of characters, that, although identical for most practical purposes, map to different Unicode points, depending on the source[4]. Either during creation of the index, or in the retrieval engine, appropriate attention has to be paid to this problem.

A retrieval engine prototype

As a proof of concept a prototype retrieval engine has been built. This engine works on top of MS-Word. It accesses the files in Word format released by the Chinese Buddhist

[4] A list of the characters with this problem is given in the appendix. It should be noted, that this is still ongoing research, and the inclusion or exclusion of some characters will depend on the type of text.

Electronic Text Association[5] (CBETA) on their most recent CD-ROM[6]. The index in the format outlined above has been built from the original XML files. Two versions have been used, one using two characters following the indexed characters (i.e. the same format as above in Figure 2), the other with four characters following the indexed character and two preceding characters, bringing the total length of the text entry to 7. The size of these two indices is given in Table 1.

With this retrieval engine, any string of up to three (in the first index) or up to five (in the second) index can be immediately accessed. For longer strings, repeated lookups would have to be done, which slow down the retrieval somehow. Search behavior on Chinese Internet sites suggest howeve, that most search terms used for retrieval are two or three characters in length. The search term can either be entered in a dialog box or, if the text is already open in Word, simply highlighted before invoking the search engine.

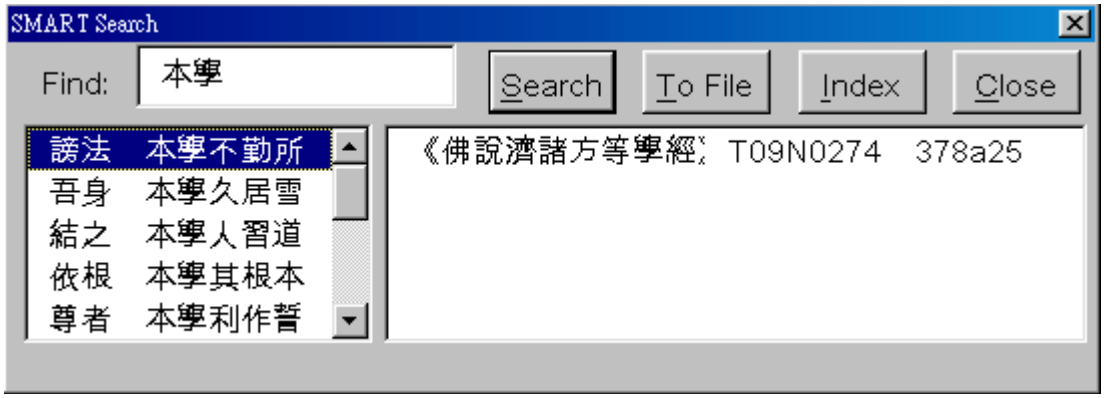

Figure 3 SMART prototype search dialog

In Figure 3, the search box of this prototype is shown. The characters 本學 have been typed in the find box and the search button has been pressed. The list in the lower left gives strings that begin with the search term and show some context to the left and right. This list is available almost immediately. In the lower left is a list of locations of the string highlighted to the left. Doubleclicking on a location opens up the file on the desired location. The highlight in the left list can be moved up and down, thus providing a way to browse the index. Click on the "Index" button allows the selection of another index, in this case only the Taisho index with a text-entry length of three characters is available, although any index in the format outlined above could be used.

[5] CBETA's homepage is at http://ccbs.ntu.edu.tw/cbeta. See also CBETA's presentation at this conference.
[6] This CD-ROM has been distributed to participants at this conference. It is also available from CBETA for the cost of mailing it. More information can be found on CBETA's homepage.

The button “To File” produces the following output in a new file:

Searchterm: 本學 Occurrences: 25

-2	-1	Index	Title	Location
謗	法	本學不勤所	《佛說濟諸方等學經》	T09N0274, p378a25
吾	身	本學久居雪	《修行道地經》	T15N0606, p210a29
結	之	本學人習道	《出曜經》	T04N0212, p736a20
依	根	本學其根本	《攝大乘論釋》	T31N1598, p429b23
尊	者	本學利作誓	《鞞婆沙論》	T28N1547, p416b09
學	如	本學善學無	《差摩婆帝授記經》	T14N0573, p946b07
聖	實	本學地聽聞	《菩薩本生鬘論》	T03N0160, p351b23
姓	子	本學大乘爲	《慧上菩薩問大善權經》	T12N0345, p162a16
言	我	本學婆羅門	《文殊師利問菩薩署經》	T14N0458, p438b05
耶	二	本學師之所	《大寶積經》	T11N0310, p266c10
不	變	本學明了在	《佛說無量壽經》	T12N0360, p266a27
菩	提	本學是勤修	《大方廣佛華嚴經》	T10N0279, p72c05
說	我	本學是法能	《佛說華手經》	T16N0657, p199a15
畏	佛	本學時在佛	《佛說濟諸方等學經》	T09N0274, p377b18
如	吾	本學此三昧	《賢劫經》	T14N0425, p64a02
諸	佛	本學皆由精	《持人菩薩經》	T14N0481, p627b16
佛	子	本學真諦解	《度世品經》	T10N0292, p653b29
眾	生	本學習世出	《菩薩念佛三昧經》	T13N0414, p815b06
聽	者	本學聲聞尋	《佛說文殊悔過經》	T14N0459, p441c15
性	之	本學般若波	《放光般若經》	T08N0221, p100c17
此	字	本學通諸法	《佛說大般泥洹經》	T12N0376, p887c22
時	人	本學道不值	《佛說三品弟子經》	T17N0767, p701a06
曰	汝	本學道二十	《菩薩從兜術天降神母胎說廣普經》	T12N0384, p1052c05
心	爲	本學道人眾	《那先比丘經》	T32N1670B, p708b23
其	佛	本學道時至	《持人菩薩經》	T14N0481, p627a17
爲	四	本學道時設	《普曜經》	T03N0186, p504b16

Again, this is hyperlinked to the Taisho textfiles, clicking on the title will open the file at the specified position. Additionally, since this is an ordinary Word-file, the table can be printed, pasted into other documents, or sorted according to different rows. For this purpose, characters preceding the searchterm have been put in their own cells, to facilitate their usage as sorting keys. It goes without saying, that there are a lot of other possibilities to analyze such a file.

This prototype proved the feasability of this approach for the retrieval as needed in SMART. It also showed, how a convenient system can be built on top of standard

applications used daily by many researchers. Nevertheless, it proved to inflexible for some other tasks necessary for SMART.

Appendix:

Table of character variants in Unicode

The following table list characters occurring in slightly different forms in Unicode, depending on the format of the non-Unicode source. Some of these characters should have been unified according to the Unicode unification rules, but another rule, the source separation rule, prevented this[7]. The table is also available in electronic form at http://www.chibs.edu.tw/~chris/smart/.

#Table format:
#Four tab separated columns. The first (Big5 source) and third (JIS source) have the pinyin readings attached for reference.
#Currently, only the characters with ++ in the last column will be used by a conversion routine.
#Created Sep. 18, 1999, Updated Jan. 4, 2000.
#Creator: Christian Wittern, chris@ccbs.ntu.edu.tw

Big5 ->	Unicode	-> JIS	Unicode	Remark
內 nei4	U-5167	nei4	U-5185	++
戶 hu4	U-6236	hu4	U-6238	++
朵 duo3	U-6735	duo3	U-6736	++
吞 tun1	U-541E	tun1	U-5451	++
吳 wu2	U-5433	wu2	U-5449	++
步 bu4	U-6B65	bu4	U-6B69	++
每 mei3	U-6BCF	mei3	U-6BCE	++
姊 zi3	U-59CA	zi3	U-59C9	++
拋 pao1	U-62CB	pao1	U-629B	++
狀 zhuang4	U-72C0	zhuang4	U-72B6	++
侉 kua1 kua3	U-4F89	誇 kua1	U-8A87	++
俠 xia2	U-4FE0	xia2	U-4FA0	++
俞 yu2 yu4	U-4FDE	yu2 shu4	U-516A	++
剎 cha4	U-524E	cha4 sha1	U-5239	++
彥 yan4	U-5F65	yan4	U-5F66	++
毗 pi2	U-6BD7	毘 pi2	U-6BD8	++
俱 yu3	U-4FC1	yu3	U-4FE3	++

[7] Many of these characters occurred also in CNS X-11643 (1986) level 14 (which mostly became level 3 in CNS X-11643 (1992)), but these characters are very rarely available in taiwanese computer systems.

Big5 ->	Unicode	-> JIS	Unicode	Remark
柰 nai4	U-67F0	**奈** nai4 nai3	U-5948	++
种 chong2 zhong3 zhong4	U-79CD	**種** zhong3 chong2 zhong4	U-7A2E	++
秏 hao4	U-79CF	**耗** hao4	U-8017	++
值 zhi2	U-503C	zhi2	U-5024	++
俱 ju4 ju1	U-4FF1	ju4 ju1	U-5036	++
剝 bo1	U-525D	bo1 bao1	U-5265	++
唧 ji2 ji1	U-5527	ji1 ji2	U-559E	++
娛 yu2	U-5A1B	yu2	U-5A2F	++
悅 yue4	U-6085	yue4	U-60A6	++
涉 she4	U-6D89	she4	U-6E09	++
玆 zi1	U-7386	**茲** zi1	U-8332	++
荔 li4	U-8354	li4	U-8318	++
偷 tou1	U-5077	tou1	U-5078	++
啞 ya1 ya3 e4	U-555E	ya1 ya3 e4	U-5516	++
巢 chao2	U-5DE2	chao2	U-5DE3	++
啟 qi3	U-555F	qi3	U-5553	++
晚 wan3	U-665A	wan3	U-6669	++
淚 lei4	U-6DDA	lei4	U-6D99	++
產 chan3	U-7522	chan3	U-7523	++
眾 zhong4	U-773E	zhong4	U-8846	++
缽 bo1	U-7F3D	bo1	U-9262	++
脫 tuo1	U-812B	tuo1	U-8131	++
堇 jin3	U-5807	**薑** jin3	U-83EB	++
喻 yu4	U-55BB	yu4	U-55A9	++
廄 jiu4	U-5EC4	jiu4	U-5ED0	++
揭 jie1	U-63ED	jie1	U-63B2	++
渴 ke3	U-6E34	ke3	U-6E07	++
溉 gai4	U-6E89	gai4	U-6F11	++
焰 yan4	U-7130	yan4	U-7114	++
稅 shui4	U-7A05	shui4	U-7A0E	++
絕 jue2	U-7D55	jue2	U-7D76	++
萊 lai2	U-840A	lai2	U-83B1	++
虛 xu1	U-865B	xu1	U-865A	++
鄉 xiang1 xiang4	U-9109	xiang1	U-90F7	++
韌 ren4	U-97CC	ren4	U-9771	++
黃 huang2	U-9EC3	huang2	U-9EC4	++
黑 hei1 hei3	U-9ED1	hei1	U-9ED2	++
畜 yu2	U-756C	yu2	U-756D	++

Big5 ->	Unicode	-> JIS	Unicode	Remark
缾 ping2	U-7F3E	瓶 ping2	U-74F6	++
羨 xian4 yan2 yi2	U-7FA1	羨 xian4	U-7FA8	++
弒 shi4	U-5F12	shi4	U-5F11	++
搔 sao1	U-6414	sao1	U-63BB	++
榆 yu2	U-6986	yu2	U-6961	++
歲 sui4	U-6B72	sui4	U-6B73	++
温 wen1	U-6EAB	wen1	U-6E29	++
粵 yue4	U-7CB5	yue4	U-7CA4	++
腳 jiao3 jue2	U-8173	jiao3 jue2	U-811A	++
躲 duo3	U-8EB2	duo3	U-8EB1	++
馱 tuo2 duo4	U-99B1	tuo2 duo4	U-99C4	++
眞 zhi4	U-5BD8	置 zhi4	U-7F6E	++
屢 lü3	U-5C62	lü3	U-5C61	++
摑 guo2	U-6451	guo2	U-63B4	++
暨 ji4	U-66A8	ji4	U-66C1	++
綠 lü4	U-7DA0	lü4	U-7DD1	++
蒞 li4	U-849E	li4	U-8385	++
説 shuo1 shui4 yue4	U-8AAA	shuo1 shui4 yue4	U-8AAC	++
噓 xu1 shi1	U-5653	xu1	U-5618	++
增 zeng1	U-589E	zeng1	U-5897	++
寬 kuan1	U-5BEC	kuan1	U-5BDB	++
德 de2	U-5FB7	de2		++
潑 po1	U-6F51	po1	U-6E8C	++
瘦 shou4	U-7626	shou4	U-75E9	++
緣 yuan2	U-7DE3	yuan2	U-7E01	++
蔣 jiang3	U-8523	jiang3 jiang1	U-848B	++
銳 rui4	U-92B3	rui4	U-92ED	++
閱 yue4	U-95B1	yue4	U-95B2	++
夐 xuan4 xiong4	U-657B	xiong4	U-5910	++
曆 li4	U-66C6	li4	U-66A6	++
橫 heng2 heng4	U-6A6B	heng2 heng4 guang1 huang2 huang4	U-6A2A	++
歷 li4	U-6B77	li4	U-6B74	++
瘺 lou4	U-763A	lou4 lü2	U-763B	++
篡 cuan4	U-7BE1	cuan4	U-7C12	++
賴 lai4	U-8CF4	lai4	U-983C	++
錄 lu4	U-9304	lu4	U-9332	++
頰 jia2	U-9830	jia2	U-982C	++
頹 tui2	U-9839	tui2	U-983D	++

Big5 ->	Unicode	-> JIS	Unicode	Remark
𡁜 yun2	U-8553	芸 yun2	U-82B8	++
幫 bang1	U-5E6B	bang1	U-5E47	++
擊 ji2 ji1	U-64CA	ji2	U-6483	++
檔 dang3	U-6A94	dang3 dang4	U-6863	++
繈 qiang3	U-7E48	qiang3 jiang3	U-7E66	++
籬 li2	U-7BF1	籬 li2	U-7C6C	++
瀆 du2	U-7006	du2	U-6D9C	++
簞 dan1	U-7C1E	dan1	U-7BAA	++
薰 xun1	U-85B0	xun1	U-85AB	++
蟬 chan2	U-87EC	chan2	U-8749	++
軀 qu1	U-8EC0	qu1	U-8EAF	++
醬 jiang4	U-91AC	jiang4	U-91A4	++
雞 ji1	U-96DE	ji1	U-9DC4	++
瀨 lai4	U-7028	lai4	U-702C	++
禱 dao3	U-79B1	dao3	U-7977	++
繫 ji4 xi4	U-7E6B	xi4 ji4	U-7E4B	++
醱 po1 fa1	U-91B1	po1 fa1	U-9197	++
麴 qu2 qu1	U-9EB4	qu1 qu2	U-9EB9	++
麵 mian4	U-9EB5	mian4	U-9EBA	++
蠟 la4 zha4	U-881F	la4 zha4	U-874B	++
纊 kuang4	U-7E8A	kuang4	U-7D4B	++
纍 lei2	U-7E8D	累 lei4 lei2 lei3	U-7D2F	++
囊 nang2 nang1	U-56CA	nang2 nang1	U-56A2	++
巔 dian1	U-5DD4	dian1	U-5DD3	++
贋 yan4	U-8D17	yan4	U-8D0B	++
鱉 bie1	U-9C49	bie1	U-9F08	++
鷗 ou1	U-9DD7	ou1	U-9D0E	++
攢 zan3 cuan2	U-6522	zan3 cuan2	U-6505	++
驒 tuo2	U-9A52	tuo2 tan2	U-9A28	++
鹼 jian3	U-9E7C	jian3	U-9E78	++
纘 zuan3	U-7E98	zuan3	U-7E89	++
卄 nian4	U-5344	廿 nian4	U-5EFF	+
丰 feng1	U-4E30	豐 feng1	U-8C50	+
扑 pu1	U-6251	撲 pu1	U-64B2	+
圣 sheng4	U-5723	聖 sheng4	U-8056	+
尒 er3	U-5C12	爾 er3	U-723E	+
汙 wu1	U-6C59	wu1	U-6C5A	+
优 you2 you1	U-4F18	優 you1	U-512A	+
异 yi4	U-5F02	異 yi4	U-7570	+

Big5 ->	Unicode	-> JIS	Unicode	Remark
佔 zhan4	U-4F54	占 zhan4 zhan1	U-5360	+
佈 bu4	U-4F48	布 bu4	U-5E03	+
妒 du4	U-5992	du4	U-59AC	+
沅 yuan2	U-6C85	源 yuan2	U-6E90	+
灶 zao4	U-7076	zao4	U-7AC8	+
牠 ta1	U-7260	它 ta1	U-5B83	+
阬 keng1	U-962C	坑 keng1	U-5751	+
怀 huai2	U-6000	懷 huai2	U-61F7	+
扰 you4 rao3	U-6270	擾 rao3	U-64FE	+
歧 qi2	U-6B67	岐 qi2	U-5C90	+
侄 zhi2	U-4F84	姪 zhi2	U-59EA	+
刱 chuang1 chuang4 qiang1	U-5231	chuang1 chuang4 qiang1	U-524F	+
岭 ling2 ling3	U-5CAD	嶺 ling3	U-5DBA	+
构 gou4 jue2	U-6784	構 gou4 jue2	U-69CB	+
炖 dun4	U-7096	燉 dun4	U-71C9	+
查 cha2 zha1	U-67E5	cha2 zha1	U-67FB	+
柒 qi1	U-67D2	漆 qi1 qu4	U-6F06	+
炤 zhao4	U-70A4	照 zhao4	U-7167	+
觔 jin1	U-89D4	斤 jin1 jin5	U-65A4	+
柜 ju3 gui4	U-67DC	櫃 gui4 ju3	U-6AC3	+
洴 ping2 beng4	U-6D34	泙 ping2	U-6CD9	+
洼 wa1	U-6D3C	窪 wa1	U-7AAA	+
玅 miao4	U-7385	妙 miao4	U-5999	+
舡 chuan2 xiang1	U-8221	船 chuan2	U-8239	+
陔 gai1	U-9654	垓 gai1	U-5793	+
唷 yo1	U-5537	唹 yu2	U-5539	+
胭 yan1	U-80ED	臙 yan1	U-81D9	+
凈 jing4	U-51C8	淨 jing4	U-6DE8	+
涂 tu2	U-6D82	塗 tu2	U-5857	+
焚 jiao1	U-832D	椒 jiao1	U-6912	+
适 kuo4 shi4	U-9002	適 shi4 kuo4	U-9069	+
恿 yong3	U-607F	yong3	U-6142	+
莆 pu2 fu3	U-8386	蒲 pu2	U-84B2	+
桮 bei1	U-686E	杯 bei1	U-676F	+
虖 hu1	U-8656	呼 hu1	U-547C	+
閆 yan2 yan3 yan4	U-9586	閻 yan2 yan3 yan4	U-95BB	+
傢 jia1	U-50A2	家 jia1 gu1	U-5BB6	+
勛 xun1	U-52DB	勳 xun1	U-52F3	+

Big5 ->	Unicode	-> JIS	Unicode	Remark
睏 kun4	U-774F	困 kun4	U-56F0	+
确 que4	U-786E	確 que4	U-78BA	+
罥 juan4	U-7F65	juan4	U-7F82	+
腊 xi2 la4	U-814A	臘 la4 xi1	U-81D8	+
溼 shi1	U-6EBC	濕 shi1	U-6FD5	+
詶 chou2	U-8A76	酬 chou2	U-916C	+
望 wang4	U-6722	望 wang4	U-671B	+
碥 bian3	U-78A5	扁 bian3 pian1	U-6241	+
耤 ji2	U-8024	藉 jie4 ji2	U-85C9	+
徵 zheng1 zhi3	U-5FB5	征 zheng1	U-5F81	+
颳 gua1	U-98B3	刮 gua1	U-522E	+
滈 hao4	U-6F94	浩 hao4	U-6D69	+
澨 se4	U-6FC7	瀉 se4	U-6F80	+
螗 tang2	U-8797	螳 tang2	U-87B3	+
閈 ban3	U-95C6	板 ban3	U-677F	+
篲 hui4	U-7BF2	彗 hui4	U-5F57	+
龐 pang2	U-9F90	厖 pang2	U-5396	+
鬍 hu2	U-9B0D	胡 hu2	U-80E1	+
艣 lu3	U-8263	櫓 lu3	U-6AD3	+
癥 zheng1 zheng4	U-7665	症 zheng4 zheng1	U-75C7	+
鏽 xiu4	U-93FD	銹 xiu4	U-92B9	+
灕 li2	U-7055	漓 li2	U-6F13	+
扇 xi4	U-5C6D	xi4	U-5C53	+
灤 luan2	U-7064	欒 luan2	U-6B12	+
豔 yan4	U-8C54	yan4	U-8276	+
籲 xu1 yu1 yu4	U-7C72	吁 xu1 yu4	U-5401	+