日中韓辭典研究所
The CJK Dictionary Institute

Linguistic Issues in Chinese to Chinese Conversion

Jack Halpern

Originally published in *Multilingual Computing*

1. Introduction

Standard Chinese is written in two forms: **Simplified Chinese** (SC), used in the People's Republic of China (PRC) and Singapore; and **Traditional Chinese** (TC), used in Taiwan, Hong Kong, Macao, and among most overseas Chinese. A common fallacy is that there is a straightforward correspondence between the two systems, and that conversion between them merely requires mapping from one character set to another, such as from GB 2312-80 to Big Five.

Although the most important difference between Simplified Chinese and Traditional Chinese lies in character form, there are also differences in character sets, encoding methods, the choice of vocabulary, and sometimes style. The language reforms in the PRC have had a major impact on character form. From the point of view of processing Chinese data, the most relevant issues are:

1. Many character forms underwent major simplifications, to the point where they are no longer recognizable from their traditional forms, e.g. TC 徵 → SC 征.
2. In numerous cases, one simplified form corresponds to two or more traditional forms (less frequently the reverse is also true), e.g. SC 征 maps to TC 徵 and 征. Normally only one of these is the correct one, depending on the context.
3. Sometimes, one simplified form maps to multiple traditional forms, *any* of which may be correct, depending on the context (e.g. SC 编制 maps to both TC 編制 'organize' and 編製 'make by knitting').
4. The GB 2312-80 standard used for SC is incompatible with the Big Five standard used for TC, resulting in numerous missing characters on both sides.

Item (2) above is the central issue in SC-to-TC conversion. The "classical" example given in such discussions are the traditional characters 發 and 髮, etymologically two distinct characters, which were merged into the single simplified form 发. The table below shows these and other examples of SC forms that map to multiple TC forms.

Table 1: SC-to-TC One-to-Many Mappings			
SC Source	**TC Target**	**Meaning**	**TC Example**
发 **fa**^{1}	發	emit	出發 start off
发 **fa**^{4}	髮	hair	頭髮 hair
干 **gan**^{1}	乾	dry	乾燥 dry
干 **gan**^{4}	幹	trunk	精幹 able, strong
干 **gan**^{1}	干	intervene	干涉 interfere with
干 **gan**^{4}	榦	tree trunk	楨榦 central figure

As can be seen, successfully converting such SC forms to their corresponding TC forms depends on the context, usually the word, in which they occur. Often, the conversion cannot be done by merely mapping one codepoint to another, but must be based on larger linguistic units, such as words. There are hundreds of other simplified forms that correspond to two or more traditional ones, leading to ambiguous, one-to-many mappings. In this article, such mappings may be referred to as **polygraphic**, since one simplified character, or *graph*, may correspond to more than one traditional (graphic) character, or vice versa.

2. The Three Conversion Levels

The process of automatically converting SC to TC (and, to a lesser extent, TC to SC) is full of complexities and pitfalls. The conversion can be implemented on three levels, in increasing order of sophistication, from a simplistic code conversion that generates numerous errors, to a sophisticated approach that takes the semantic/lexemic differences into account. A fourth level, which takes the syntactic/contextual differences into account, is omitted from this article.

Table 2: The Four Conversion Levels		
Level 1	**Code**	Character-to-character, *code*-based substitution
Level 2	**Orthographic**	Word-to-word, *character*-based conversion
Level 3	**Lexemic**	Word-to-word, *lexeme*-based conversion

2.1 Level 1: Code Conversion

2.1.1 Basic Concepts

The easiest, but most unreliable, way to convert SC to TC, or vice versa, is to do so on a codepoint-to-codepoint basis; that is, to do a simple substitution by replacing a source codepoint of one character set (such as GB 2312-80 0xB9FA for SC 国) with a target codepoint of another character set (such as Big Five 0xB0EA for TC 國) by looking the source up in a hard-coded, one-to-one mapping table.

This kind of conversion can be described as character-to-character, *code*-based substitution, and is referred to as **code conversion,** because the units participating in the conversion process are limited to single codepoints. That is, the text stream is not parsed into higher level linguistic units, like words. Below is an example of a one-to-one code mapping table.

Table 3: Code Mapping Table

SC Source	GB0 (EUC)	TC Target	BIG5	Omitted Candidates
发	B7A2	發	B56F	髮
干	B8C9	幹	A47A	乾 干 榦
里	C0EF	裡	B8CC	里 裏
征	D5F7	徵	BC78	征
门	C3C5	門	AAF9	

Since such tables map each source character to only one target character, the other possible candidates (shown in the “Omitted Candidates” column) are ignored, which frequently results in incorrect conversion. For example, an SC string such as 头发 ‘hair’ is not treated as a single unit, but is converted character by character. Since SC 头 maps only to TC 頭, the conversion succeeds. On the other hand, since SC 发 ‘hair’ maps to both TC 髮 ‘hair’ and TC 發 ‘emit’, the conversion may fail.

2.1.2 The Conversion Process

Code conversion can be implemented in three different ways, in increasing order of sophistication.

1. Simplistic conversion: This refers to system based on one-to-one mapping tables in which the target codepoint is one of several alternatives selected without sufficiently considering its frequency of occurrence.

2. Frequency-based conversion: This refers to a system based on one-to-one mapping tables in which the target codepoint is the *first* of several alternatives, selected from a list ordered by frequency of occurrence. Although this approach frequently leads to correct results, it is likely to fail in the many cases where the second (or third) alternative of multiple target mappings is itself of high frequency, as in the case of 发, which maps to both TC 發 and 髮.

3. Candidate-based conversion: This refers to a system based on one-to-many mapping tables, with the alternative candidates listed in the output so that the user must select the correct candidate.

Code conversion has three disadvantages: (1) if implemented as simplistic conversion, it will normally produce unacceptable results; (2) even if implemented intelligently (approaches (2) and (3) above), it may require considerable human intervention in the form of candidate selection and/or post-editing; and (3) it totally ignores differences in vocabulary (discussed below).

2.2 Level 2: Orthographic Conversion

The next level of sophistication in SC↔TC conversion can be described as word-to-word, *character*-based conversion. We call this **orthographic conversion,** because the units participating in the conversion process consist of orthographic units: that is, characters or meaningful combinations of characters that are treated as single entries in dictionaries and mapping tables. We refer to these as **word-units.** Word-units represent meaningful linguistic units such as single-character words (free forms), word elements such as affixes (bound morphemes), multi-character compound words (free and bound), and even larger units such as idiomatic phrases.

Orthographic conversion is carried out on a word-unit basis in four steps:

1. Segmenting the source sentence or phrase into word-units.
2. Looking up the word-units in orthographic (word-unit) mapping tables.
3. Generating the target word-unit.
4. Outputting the target word-unit in the desired encoding.

For example, the SC phrase 梳头发 (**shu^1 tou^{2}fa^0**) ‘comb one’s hair’, is first segmented into the word-units 梳 ‘comb’ (single-character free morpheme) and 头发 ‘hair’ (two-character compound), each is looked up in the mapping table, and they are converted to the target string 梳頭髮. The important point is that 头发 is *not* decomposed, but is treated as a single word-unit. (Actually, this example is complicated by the fact that 梳頭 'comb one's hair' is also a legitimate word-unit.) Below is an example of an orthographic (word-unit) mapping table.

Table 4: Orthographic Mapping Table

SC Word-Unit	TC Word-Unit	Pinyin	Meaning
头发	頭髮	**tou^{2}fa^0**	hair
出发	出發	**chu^{1}fa^1**	start off
干燥	乾燥	**gan^{1}zao^4**	dry
暗里	暗裡	**an^{4}li^3**	secretly
千里	千里	**qian^{1}li^3**	long distance
秋千	鞦韆	**qiu^{1}qian^1**	a swing

It is important to note that in both code conversion and orthographic conversion, the results must be in **orthographic correspondence** with the source. That is, the source and target are merely orthographic variants of the same underlying *lexeme* (see section 2.3 below). This means that each source character must be either identical to, or in exact one-to-one correspondence with, the target character.

For example, in converting SC 计算机 (**ji^{4}suan^{4}ji^1**) to TC 計算機 'computer', 计 corresponds to 計, 算 corresponds to 算 (identical glyph), and 机 corresponds to 機 on a one-to-one basis. No attempt is made to "translate" SC 计算机 to TC 電腦 (**dian^{4}nao^3**), as is done in lexemic (Level 3) conversion.

2.3 Level 3: Lexemic Conversion

Orthographic conversion works well as long the source and target words are in orthographic correspondence, as in the case of SC 头发 and TC 頭髮. Unfortunately, Taiwan, Hong Kong, and the PRC have sometimes taken different paths in coining technical terminology, proper nouns and even many ordinary words. As a result, there are numerous cases where SC and TC have entirely different words for the same concept. Probably the best known of these is *computer*, which is normally 计算机 (**ji^{4}suan^{4}ji^1**) in SC but always 電腦 (**dian^{4}nao^3**) in TC.

The next level of sophistication in SC↔TC conversion is to take these differences into account by "translating" from one to the other, which can be described as word-to-word, *lexeme*-based conversion. We call this **lexemic conversion,** because the units participating in the conversion process consist of semantic units, or *lexemes*. A

lexeme is a basic unit of vocabulary, such as a single-character word, affix, or compound word. Here it also denotes larger units, such as idiomatic phrases. For practical purposes, it is similar to the word-units used in orthographic conversion, but the term *lexeme* is used here to emphasize the semantic nature of the conversion process.

Let us take the SC string 信息处理 (**xin^{1}xi^4 chu^{3}li^3**) ‘information processing’, as an example. It is first segmented into the lexemes 信息 and 处理, each is looked up in a lexemic mapping table, and they are then converted to the target string 資訊處理 (**zi^{1}xun^4 chu^{3}li^3**).

It is important to note that 信息 and 資訊 are *not* in orthographic correspondence; that is, they are distinct lexemes in their own right, not just orthographic variants of the same lexeme. This is not unlike the difference between American English ‘gasoline’ and British English ‘petrol’.

The difference between 处理 and 處理, on the other hand, is analogous to the difference between American English ‘color’ and the British English ‘colour’, which are orthographic variants of the same lexeme. This analogy to English must not be taken too literally, since the English and Chinese writing systems are fundamentally different.

Lexemic conversion is similar to orthographic conversion, but differs from it in two important ways: (1) The mapping tables must map one lexeme to another on a semantic level, if appropriate. For example, SC 计算机 must map to its TC lexemic equivalent 電腦, not to its orthographic equivalent 計算機, and (2) The segmentation algorithm must be sophisticated enough to identify proper nouns, since the choice of target character could depend on whether the lexeme is a proper noun or not. Below is an example of a lexemic mapping table.

Table 5: Lexemic Mapping Table

English	**SC Lexeme**	**SC Pinyin**	**TC Lexeme**	**TC Pinyin**
bit	位	**wei^4**	位元	**wei^{4}yuan^2**
byte	字节	**zi^{4}jie^2**	位元組	**wei^{4}yuan^{2}zu^3**
CD-ROM	光盘	**guang^{1}pan^2**	光碟	**guang^{1}die^2**
computer	计算机	**ji^{4}suan^{4}ji^1**	電腦	**dian^{4}nao^3**
database	数据库	**shu^{4}ju^{4}ku^4**	資料庫	**zi^{1}liao^{4}ku^4**
file	文件	**wen^{2}jian^4**	檔案	**dang^4'an^4**

information	信息	**xin^{1}xi^4**	資訊	**zi^{1}xun^4**
Bin Ladin	本拉登	**ben^{3}la^{1}deng^1**	賓拉登	**bin^{1}la^{1}deng^1**
software	软件	**ruan^{3}jian^4**	軟體	**ruan^{3}ti^3**
Taxi	出租车	**chu^{1}zu^{1}che^1**	計程車	**ji^{4}cheng^{2}che^1**

As can be seen, the above table maps the semantic content of the lexemes of one variety of Chinese to the other, and in that respect is identical in structure to a bilingual glossary.

Code and orthographic converters are obviously incapable of dealing with lexemic differences, such as between SC 计算机 and TC 電腦, since these are distinct lexemes for the same concept. There are also many non-Chinese proper nouns that are not transliterated with the same characters. For example, SC 佐治亚 (**zuo^{3}zhi^{4}ya^4**), a phonetic transliteration of ‘Georgia’, should map to TC 喬治亞 (**qiao^{2}zhi^{4}ya^4**), *not* to its orthographically equivalent 佐治亞.

3. Advanced Conversion Technology

3.1 Project Overview

In 1996, **The CJK Dictionary Institute** (CJKI) (based near Tokyo), which specializes in CJK computational lexicography, launched a project whose ultimate goal is to develop a Chinese-to-Chinese conversion system that gives near-perfect results. This has been a major undertaking that required considerable investment of funds and human resources.

To this end, we have engaged in the following research and development activities: (1) in-depth investigation of all the technical and linguistic issues related to Chinese-to-Chinese conversion, (2) construction of comprehensive SC↔TC mapping tables, and (3) research on Chinese word segmentation technology.

To achieve a high level of conversion accuracy, our large-scale Chinese lexical databases include approximately three million general vocabulary lexemes, technical terms, and proper nouns. They also include various other attributes, such as pinyin readings, grammatical information, part of speech, and semantic classification codes.

3.2 How Severe is the Problem?

Calculations based on our comprehensive Chinese lexical database, which currently contains approximately three million items, show that more than 20,000 of the approximately 97,000 most common SC word-units contain at least one polygraphic character, which leads to one-to-many SC-to-TC mappings. This represents an astounding 21%. A similar calculation for TC-to-SC mappings resulted in 3025, or about 3.5%, out of the approximately 87,000 most common TC word-units. These figures demonstrate that merely converting one codepoint to another, especially in the SC-to-TC direction, will lead to unacceptable results.

Since many high-frequency polygraphic characters are components of hundreds, or even thousands, of compound words, incorrect conversion will be a common occurrence unless the one-to-many mappings are disambiguated by (1) segmenting the byte stream into semantically meaningful units (word-units or lexemes) and, (2) analyzing the context to determine the correct choice out of the multiple candidates.

3.3 System Components

Below is a brief description of the principal components of the conversion system:

1. **Code mapping tables:** Our SC↔TC code mapping tables are comprehensive and complete. They are not restricted to the GB 2312-80 and Big Five character sets, but cover all SC and TC codepoints. In the case of one-to-many SC-to-TC mappings, the candidates are arranged in order of frequency based on statistics derived from a corpus of 170 million characters, as well as on several years of research by our team of TC specialists.
2. **Orthographic mapping tables:** Constructing accurate orthographic mapping tables for tens of thousands of polygraphic compounds requires extensive manual labor. Our team of TC specialists compiled such tables by examining and double-checking each word individually.
3. **Lexemic mapping tables:** Constructing accurate lexemic mapping tables is even more laborious, since there is no orthographic correspondence between the SC and TC characters, and since dictionaries showing SC/TC differences do not exist. Each word must be examined individually, while taking into account the extra complications resulting from semantic ambiguities and proper nouns.
4. **Proper noun mapping tables:** Special treatment has been given to proper nouns, especially personal and place names. Our mapping tables for Chinese and non-Chinese names currently contain over 800,000 entries.
5. **Conversion Engine:** The conversion system requires a conversion engine whose major components consist of: (1) a sophisticated **Chinese word segmenter,** which segments the text stream into word-units and identifies their

grammatical functions, and (2) the **conversion module,** which looks up the word-units in the mapping tables and generates the output in the target encoding

The mapping tables contain about 1.2 million entries, a sufficiently large coverage to support robust industrial-strength applications.

4.3 Conclusions

Chinese-to-Chinese conversion has become increasingly important to the localization, translation, and publishing industries, as well as to software developers aspiring to penetrate the East Asian market. But, as we have seen, the issues are complex and require a major effort to build mapping tables and to develop segmentation technology.

The CJK Dictionary Institute finds itself in a unique position to provide software developers with high quality Chinese lexical resources and reliable conversion technology, thereby eliminating expensive manual labor and significantly reducing costs. We are convinced that our ongoing research and development efforts in this area are inexorably leading us toward achieving the elusive goal of building the perfect converter.

Jack Halpern, *the CEO of Japan-based* ***The CJK Dictionary Institute*** *(cjk.org), which specializes in the compilation of large-scale CJK and Arabic dictionaries. Author of several comprehensive Japanese and Chinese dictionaries, Jack can be reached at jack@cjki.org. For a detailed treatment of Chinese to Chinese conversion, see cjk.org/cjk/c2c/c2centry.htm.*